Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

# Multimodal Learning of Actions with Deep Neural Network Self-Organization

**Dissertation**

submitted to the Universität Hamburg, Faculty of Mathematics, Informatics and Natural Sciences, Department of Informatics, in partial fulfilment of the requirements for the degree of *Doctor rerum naturalium* (Dr. rer. nat.)

**German I. Parisi**
Hamburg, 2016

Submitted:

November 10, 2016

Day of oral defence:

March 10, 2017

Dissertation committee:

Prof. Dr. Stefan Wermter (advisor)
Department of Informatics, Universität Hamburg, Germany

Dr. Victor Uc-Cetina (reviewer)
Department of Informatics, Universität Hamburg, Germany

Prof. Dr. Jianwei Zhang (chair)
Department of Informatics, Universität Hamburg, Germany

*To my parents E.J. and Laura, and my brother Matías.*

*"Experience is the teacher of all things."*
— JULIUS CAESAR

# Abstract

Perceiving the actions of other people is one of the most important social skills of human beings. We are able to reliably discern a variety of socially relevant information from people's body motion such as intentions, identity, gender, and affective states. This ability is supported by highly developed visual skills and the integration of additional modalities that in concert contribute to providing a robust perceptual experience. Multimodal integration is a fundamental feature of the brain that together with widely studied biological mechanisms for action perception has served as inspiration for the development of artificial systems. However, computational mechanisms for processing and integrating knowledge reliably from multiple perceptual modalities are still to be fully investigated.

The goal of this thesis is to study and develop artificial learning architectures for action perception. In light of a wide understanding of the brain areas and underlying neural mechanisms for processing biological motion patterns, we propose a series of neural network models for learning multimodal action representations. Consistent with neurophysiological studies evidencing a hierarchy of cortical layers driven by the distribution of the input, we demonstrate how computational models of input-driven self-organization can account for the learning of action features with increasing complexity of representation. For this purpose, we introduce a novel model of recurrent self-organization for learning action features with increasingly large spatiotemporal receptive fields. Visual representations obtained through unsupervised learning are incrementally associated to symbolic action labels for the purpose of action classification.

From a multimodal perspective, we propose a model in which multimodal action representations can develop from neural network organization in terms of associative connectivity patterns between unimodal representations. We report a set of experiments showing that deep self-organizing hierarchies allow to learn statistically significant features of actions, with multimodal representations emerging from co-occurring audiovisual stimuli. We evaluated our neural network architectures on the tasks of human action recognition, body motion assessment, and the detection of abnormal behavior. Finally, we conducted two robot experiments that provide quantitative evidence for the advantages of multimodal integration for triggering sensory-driven motor behavior. The first scenario consists of an assistive task for the detection of falls, whereas in the second experiment we propose audiovisual integration in an interactive reinforcement learning scenario. Together, our results demonstrate that deep neural self-organization can account for robust action perception, yielding state-of-the-art performance also in the presence of sensory uncertainty and conflict.

The research presented in this thesis comprises interdisciplinary aspects of action perception and multimodal integration for the development of efficient neurocognitive architectures. While the brain mechanisms for multimodal perception are still to be fully understood, the proposed neural network architectures may be seen as a basis for modeling higher-level cognitive functions.

# Zusammenfassung

Die Wahrnehmung von Aktionen anderer Personen ist eine der wichtigsten sozialen Kompetenzen von Menschen. Wir sind in der Lage, eine Vielzahl von relevanten sozialen Informationen aus den Körperbewegungen von Personen zu extrahieren; dazu gehören Absichten, Identität, Geschlecht und Gefühlszustände. Diese Fähigkeit stützt sich auf ein hochentwickeltes visuelles System und die Integration von zusätzlichen Modalitäten, die gemeinsam dazu beitragen, eine robuste Wahrnehmungserfahrung zu schaffen. Multimodale Wahrnehmung ist eine fundamentale Eigenschaft des Gehirns, welche zusammen mit den biologisch gut erforschten Mechanismen zur Aktionswahrnehmung als Inspiration für die Entwicklung künstlicher Systeme gedient hat. Dennoch ist die Forschungsfrage, wie man Wissen maschinell aus einer Vielzahl von Modalitäten verlässlich verarbeiten und verbinden kann, noch offen.

Die vorliegende Arbeit beschäftigt sich mit der Erforschung und Entwicklung von künstlichen Lernarchitekturen zur Aktionswahrnehmung. Vor dem Hintergrund des weitverbreiteten Verständnisses der Gehirnregionen und zugrundeliegenden neuronalen Mechanismen zur Verarbeitung von Bewegung in biologischen Systemen, präsentieren wir eine Reihe von neuronalen Netzwerkmodellen zum Erlernen von Repräsentationen von multimodalen Aktionen. In Einklang mit neurophysiologischen Studien, die eine stimulusgetriebe Hierarchie von kortikalen Ebenen belegen, zeigen wir, wie Computermodelle von stimulusgetriebener Selbstorganisation für das Erlernen von Aktionsmerkmalen Rechnung tragen können. Zu diesem Zweck stellen wir ein neues Modell rekurrenter Selbstorganisation zum Erlernen von Aktionsmerkmalen vor, welches wachsende raum-zeitliche rezeptive Felder nutzt. Visuelle Repräsentationen, welche mit Hilfe von unüberwachtem Lernen gewonnen werden, werden zum Zweck der Aktionsklassifikation inkrementell mit symbolischen Aktionslabeln assoziiert.

Von einer multimodalen Perspektive stellen wir ein Modell vor, in dem sich Aktionsrepräsentation aus neuronaler Netzwerkorganisation ergibt, im Sinne von Mustern in der Konnektivität von Assoziationen unimodaler Repräsentationen. Wir führen eine Reihe von Experimenten durch, die zeigen, wie tiefe, selbstorganisierende Hierarchien das Erlernen von statistisch signifikanten Aktionsmerkmalen erlauben, wobei multimodale Repräsentation aus gemeinsam auftretenden audiovisuellen Stimuli hervorgeht. Wir evaluieren unsere neuronalen Netzwerkarchitekturen mit Aufgaben zur Erkennung menschlicher Bewegung, zur Körperbewegungsbeurteilung und zur Erkennung von abnormalem Verhalten. Abschliessend führen wir zwei Experimente mit Robotern durch, welche quantitativ die Vorteile von multimodaler Integration zum Auslösen von sensorgetriebenem motorischen Verhalten belegen. Das erste Szenario besteht aus einer assistiven Aufgabe zur Sturtzerkennung, während im zweiten Experiment ein Vorschlag zur audiovisuellen Integration in einem interaktiven Szenario erbracht wird. Zusammen zeigen unsere Ergebnisse, dass tiefe neuronale Selbstorganisation eine robuste Aktionswahrnehmung ermöglicht und dem Stand der Technik entsprechende Ergebnisse liefern kann, selbst bei unsicheren oder widersprüchlichen Sensormessungen.

Die Forschung in dieser Arbeit beinhaltet interdisziplinäre Aspekte der Aktionswahrnehmung und der multimodalen Integration mit dem Ziel der Entwicklung von effizienten neurokognitiven Architekturen.Während die Mechanismen, welche das Gehirn zur multimodalen Wahrnehmung nutzt, noch näher erforscht werden müssen, können die vorgestellten neuronalen Netzwerkarchitekturen als Basis zur Modellierung von höheren kognitiven Funktionen gesehen werden.

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

The daily perceptual experience of human beings is driven by an array of sensors that in concert contribute to the efficient and robust interaction with the environment (Stein and Meredith, 1993; Ernst and Bülthoff, 2004; Stein et al., 2009). We are able to reliably discern a variety of relevant social cues from people's body motion such as intentions, identity, gender, and affective states (Blake and Shiffrar, 2007; Giese and Rizzolatti, 2015), which is supported by the development of a highly skilled visual perception and the integration of additional modalities. The ability to integrate multisensory information is a fundamental and widely studied feature of the brain, yielding the effective processing of body motion patterns also from strongly degraded stimuli (Neri et al., 1998; Thornton et al., 1998; Poom and Olsson, 2002). Therefore, the findings of the underlying biological mechanisms for action perception have played an inspiring role in the development of artificial systems aimed to address the robust recognition of actions, for instance, by integrating auditory and visual patterns. Computational models for multimodal integration are a paramount ingredient of autonomous robots to forming robust and meaningful representations of perceived events (Ursino et al., 2014).

Multimodal representations have been shown to improve performance in the research areas of human action recognition, human-robot interaction, and sensory-driven robot motor behavior (Kachouie et al., 2014; Noda et al., 2014; Bauer et al., 2015). However, multisensory inputs must be represented and integrated in an appropriate way so that they result in a reliable perceptual experience aimed to trigger adequate behavioral responses. Since real-world events unfold at multiple spatial and temporal scales, artificial learning architectures aiming at tackling complex perceptual tasks should account for the multimodal processing of spatiotemporal stimuli with multiple levels of complexity and abstraction (Fonlupt, 2003; Hasson et al., 2008; Lerner et al., 2011). This kind of hierarchical aggregation is an essential organizational principle of brain cortical networks that together with the interplay of multiple modalities drives a series of perceptual and cognitive processes (Taylor et al., 2015). Consequently, the question of how to acquire, process, and integrate multimodal knowledge in artificial neurocognitive systems represents a fundamental issue still to be fully investigated.

## Research Objective

The main goal of this thesis is the study and development of artificial learning architectures for action perception motivated by a set of neurophysiological findings and behavioral studies. We take inspiration from the underlying neural mechanisms of the brain areas dedicated to processing biological motion from a set of available perceptual cues. These mechanisms include the hierarchical nature of cortical areas for processing spatiotemporal patterns with an increasing complexity and abstraction of representation (Hasson et al., 2008; Taylor et al., 2015) and the development of cortical connectivity patterns through neural network self-organization (Willshaw and von der Malsburg, 1976; Nelson, 2000). In the light of a more substantial understanding of the development and properties of cortical maps in the mammalian brain, well-studied computational mechanisms of input-driven self-organization can be extended to model learning architectures that account for complex multimodal tasks, e.g., from rudimentary action perception to higher-level cognitive functions.

The key objective of this thesis is in the development of multimodal action representations from neural network self-organization. More specifically, how can statistically significant action cues from co-occurring auditory and visual inputs be combined in an unsupervised manner by learning connectivity patterns between unimodal representations. Although the development of associations between co-occuring stimuli for multimodal binding has been supported extensively by neurophysiological studies (Fiebelkorn et al., 2009) with strong links between the brain areas governing visual and language processing (Foxe et al., 2000; Pulvermüller, 2005), computational models for the efficient multimodal binding of spatiotemporal features have remained an open issue (Ursino et al., 2014).

As a complementary goal, we aim to validate the proposed neural network models for multimodal action perception in robot experiments with real-world tasks. In contrast to the evaluation of computational models with data collected in highly controlled conditions, these experiments are aimed at assessing how the proposed neural architectures deal with rich streams of information also in the case of sensory uncertainty and conflict. In particular, we wish to provide quantitative evidence on the advantages conveyed by the use of multiple modalities for human-robot interaction tasks comprising sensory-driven motor behavior.

## Contribution to Knowledge

The contribution to knowledge of this thesis is a detailed study of neural network self-organization and the development of deep self-organizing architectures for learning multimodal action representations. These architectures are in line with a set of biological findings evidencing a hierarchy of neural detectors for processing spatiotemporal body motion cues with increasing complexity of representation. We demonstrate how self-organizing architectures can be extended to account for a set of visual tasks such as human action recognition, body motion assessment, and the detection of abnormal behavior. In particular, we propose a

deep self-organizing architecture for learning visual action representations in an unsupervised manner. This architecture comprises multiple layers of recurrent neural networks to implement the hierarchical processing of visual cues with increasingly larger spatiotemporal receptive fields from depth map videos. Furthermore, we propose an approach for learning multimodal action representations from neural self-organization in terms of asymmetric connectivity patterns between unimodal representations, allowing the bidirectional retrieval of audiovisual patterns. Our experimental results with computer simulations and interactive robots show the importance of multimodal processing for improving human-robot interaction and sensory-driven motor behavior, especially in the case of sensory uncertainty and conflict in real-world tasks.

## Thesis Organization

For a better understanding of the challenges considered in this thesis, we provide an introduction to multimodal action recognition in Chapter 2, where we review well-established findings regarding action perception in the brain along with a background on computational architectures for state-of-the-art human action recognition, body motion assessment, and abnormal behavior detection in assistive robot scenarios. In Chapter 3, we present the pillars of experience-driven cortical organization and computational models of neural network self-organization. As a modelling foundation to address our research question, we focus on a number of topology-preserving networks for the development of topological maps driven by the distribution of the input.

In Chapter 4, we propose a set of neurobiologically-motivated neural network architectures for action recognition from depth map videos in real time. Our approach consists of hierarchically-arranged self-organizing networks processing action cues in terms of body posture and motion features. Furthermore, we introduce our dataset of full-body actions that we use to evaluate the architectures proposed in this and following chapters. In Chapter 5, we investigate the use of hierarchical self-organizing learning for the development of congruent multimodal action representations. In particular, we propose a model where multimodal representations emerge from the co-occurrence of auditory and visual stimuli via the learning of associative connections between unimodal representations, yielding the bidirectional retrieval of audiovisual patterns.

In Chapter 6, we propose a novel temporal extension of a self-organizing network equipped with recurrent connectivity for dealing with time-varying patterns. We use this recurrent network in a hierarchical architecture for the unsupervised learning of action representations with increasingly larger spatiotemporal receptive fields. In order to compare our proposed architecture with respect to current trends in deep learning, we show how our model accounts for the learning of robust action-label mappings also in the case of occasionally absent or even contradictory action class labels during training sessions. Additionally, we show how the same recurrent neural network mechanism can deal with both action recognition and body motion assessment in real time.

In Chapter 7, we apply aspects of multimodal integration for enhancing human-robot interaction and triggering robust sensory-driven robot behavior in dynamic environments. We conduct experiments in two scenarios: a robot-human assistance task for fall detection and a multimodal interactive reinforcement learning task with a robot cleaning a table and receiving instructions from both vocal and gesture commands. Experiments show that the integration of multiple modalities leads to a significant improvement of performance with respect to unimodal approaches.

Concluding in Chapter 8, the proposed neural network architectures and reported results are discussed from the perspective of our research questions, analyzing analogies and limitations with respect to biological findings and providing a number of future research directions.

# Chapter 2

# Multimodal Action Recognition

The robust recognition of others' actions represents a crucial component underlying social cognition. Humans can reliably discriminate a variety of socially relevant cues from body motion such as intentions, identity, gender, and affective states (Blake and Shiffrar, 2007; Giese, 2015). Neurophysiological studies have identified a specialized area for the visual coding of complex motion in the mammalian brain (Perrett et al., 1982), comprising neurons selective to biological motion in terms of time-varying patterns of form and motion features in a wide number of brain structures (Giese and Rizzolatti, 2015). Furthermore, the ability of the brain to integrate multisensory information plays a crucial role aimed to provide a robust perceptual experience for an efficient interaction with the environment (Stein and Meredith, 1993; Ernst and Bülthoff, 2004; Stein et al., 2009). Consequently, the investigation of the biological mechanisms of action perception is fundamental to the development of artificial systems that should account for the robust processing of body motion cues from cluttered environments and rich streams of information.

In Section 2.1, we provide an introduction to multimodal action perception in humans and the underlying neural mechanisms in the brain, whereas in Section 2.2 we describe a variety of computational models aimed to tackle complex visual tasks such as human action recognition, body motion assessment, and the detection of abnormal behavior, along with a set of technical challenges involve in embedding these systems into robotic platforms.

## 2.1 Action Recognition in the Brain

### 2.1.1 How We Learn to See Others

The skill to recognize biological motion in humans arises in early life. The ability of neonates to imitate manual gestures suggests that the recognition of complex motion may depend on innate neural mechanisms (Meltzoff et al., 1977). Studies on preferential looking with four-month-old infants evidence a preference for staring at human motion sequences for a longer duration than at sequences with random motion (Bertenthal and Pinto, 1993). Additional behavioral studies have shown

that young children aged three to five years steadily enhance their skills to identify human and non-human biological motion portrayed as animations of point-light tokens and reach adult performance by age five (Pavlova et al., 2001).

The preservation of the ability to reliably discriminate different forms of body motion from normal and impoverished stimuli has been reported for observers older than sixty years old (Norman et al., 2004), in contrast to reported age-related deficits in the visual system such as deterioration of speed discrimination and detection of low-contrast moving contours. Experiments on action discrimination tasks have evidenced a remarkable efficiency of adult observers to temporally integrate body motion from highly improvised visual stimuli, e.g., partially occluded bodies, body motion embedded within noise or animated figures represented by a small number of moving dots (Johansson, 1973; Neri et al., 1998; Thornton et al., 1998; Poom and Olsson, 2002). On the other hand, significantly decreased performance of action perception has been reported for temporal disruptions of the stimuli (temporally scrambled frames of videos) and strong spatial rotation (upside-down clips) of both biological and artificial motion morphs (Bertenthal and Pinto, 1993; Jastorff et al., 2006). Interestingly, Jastorff et al. (2006) have shown that after a number of trials, observers improve their ability to recognize sequences of upside-down body motion, whereas such an improvement over multiple trials has not been reported for temporally disrupted versions of videos, thus suggesting that action recognition is highly selective in terms of the temporal order of presented stimuli. Moreover, these studies have shown that learning plays an important role in complex motion discrimination, with recognition speed and accuracy of humans being improved after a number of training sessions, not only for biologically relevant motion but also for artificial motion patterns underlying a skeleton structure (Jastorff et al., 2006; Hiris, 2007).

In addition to highly skilled visual mechanisms for motion analysis, a vast variety of studies has shown that visual perception is strongly interwoven with additional perceptual modalities and higher-level cognitive processes (Foxe et al., 2000; Raij et al., 2000; Pulvermüller, 2005). Words for actions and events appear to be among children's earliest vocabulary (Bloom, 1993). A central question in the field of developmental learning is how children first attach verbs to their referents. During their development, children have a wide range of perceptual, social, and linguistic cues at their disposal that they can use to attach a novel label to a novel referent (Hirsch-Pasek et al., 2000). The referential ambiguity of verbs may then be solved by children assuming that words map onto the most perceptually salient action in their environment. Recent experiments have shown that human infants are able to learn action–word mappings using cross-situational statistics, thus also in the presence of occasionally unavailable ground-truth action words (Smith and Yu, 2008). Furthermore, action words can be progressively learned and improved from linguistic and social cues so that novel words can be attached to existing visual representations. This hypothesis is supported by neurophysiological studies evidencing strong links between the cortical areas governing visual and language processing, and suggesting high levels of functional interaction of these areas for the formation of multimodal representations of audiovisual stimuli (Foxe et al.,

2000; Raij et al., 2000; Belin et al., 2000, 2002; Pulvermüller, 2005).

Together, these studies suggest a highly robust and adaptive system for the efficient analysis of biological motion and synthetically generated patterns of biomechanically plausible motion. For over five decades, the neural mechanisms of the mammalian brain for action perception have been subject to multidisciplinary studies, with insights about biological motion processing having the dual goal of improving our understanding of the brain and contributing to the development of artificial models of perception.

### 2.1.2 Neural Mechanisms for Action Perception

Studies have identified a specialized area for the visual coding of complex, articulated motion in the mammalian brain (Perrett et al., 1982). Early processing of visual input starts in the primary visual cortex (V1) and extends to higher-level and diverse areas of the brain. In particular, neurons selective to biological motion in terms of time-varying patterns of form and motion features have been found in a wide number of brain structures such as the superior temporal sulcus (STS), the parietal, the premotor and the motor cortex (Giese and Rizzolatti, 2015). A schematic illustration of the brain containing a series of areas involved in visual processing is shown in Fig. 2.1.

#### Two-Pathway Processing of Visual Cues

Neurophysiological studies have shown that the mammalian visual system processes biological motion in two neural pathways (Ungerleider and Mishkin, 1982; Felleman and Van Essen, 1991). The ventral pathway recognizes sequences of snapshots of body form, while the dorsal pathway recognizes movements in terms of optic-flow patterns. Both pathways comprise hierarchies that extrapolate visual features with increasing complexity of representation. Visual processing in cortical areas is hierarchical, with increasingly larger spatiotemporal receptive windows where simple features manifest in low-level layers closest to sensory inputs, while increasingly complex representations develop in deeper layers (Taylor et al., 2015; Hasson et al., 2008; Lerner et al., 2011). Specifically for the visual cortex, Hasson et al. (2008) have shown that while early visual areas such as the primary visual cortex (V1) and the motion-sensitive area (MT+) yield higher responses to instantaneous sensory input, high-level areas such as the superior temporal sulcus (STS) are more affected by information accumulated over longer timescales. Neurons in higher levels of the hierarchy are also characterized by gradual invariance to the position and the scale of the stimulus (Orban et al., 1982). This kind of hierarchical aggregation is a fundamental organizational principle of cortical networks for dealing with perceptual and cognitive processes that unfold over time (Fonlupt, 2003).

Although there has been a long-standing debate on which visual cue is predominant to action understanding, i.e. either snapshots of body form (Lange and Lappe, 2006) or optic flow patterns (Troje, 2002), it has been found that neurons

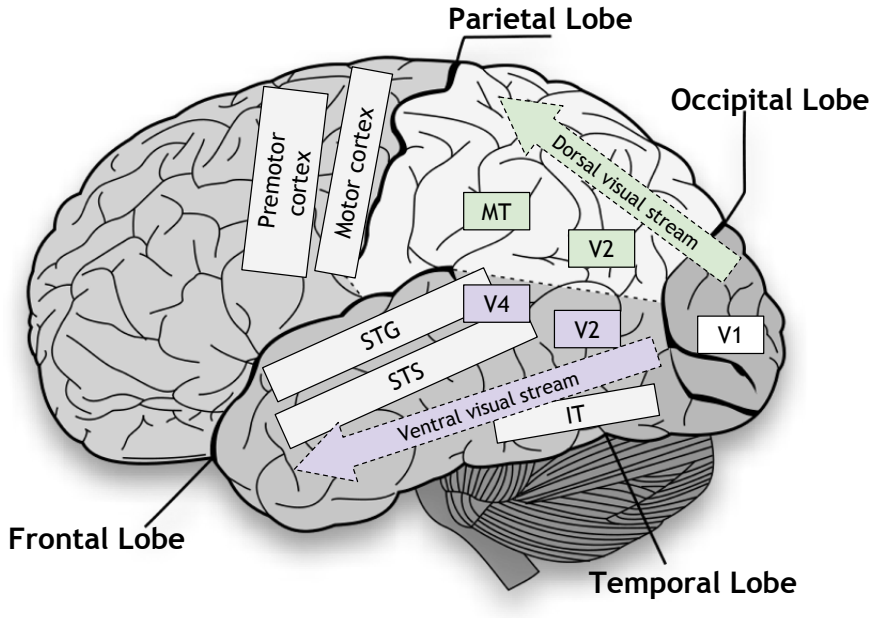



Figure 2.1: Schematic illustration of the brain for visual processing. IT, inferior temporal cortex; MT, middle temporal cortex; STG, superior temporal gyrus; STS, superior temporal sulcus; V1, primary visual cortex; V2, secondary visual cortex (prestriate cortex); V4, visual area in the extrastriate visual cortex.

in the macaque STS that are sensitive to both motion and posture for representing similarities among actions, thus suggesting contributions from converging cues received from the ventral and dorsal pathways (Oram and Perrett, 1996). On the basis of additional studies showing that neurons in the human STS activate by body articulation (Beauchamp et al., 2003), there is a consensus that posture and motion together play a key role in biological motion perception (Garcia and Grossman, 2008; Thirkettle et al., 2009). It is to be noted that the conceptual separation into two distinct pathways represents a simplification, while it is known that the two processing streams comprise interactions at several levels (Felleman and Van Essen, 1991). The underlying neural mechanisms and functional underpinning of this interaction are still to be fully investigated.

A well-established computation model used to provide a qualitative analysis of existing data on biological movement recognition was proposed by Giese and Poggio (2003). It consists of a feedforward, two-pathway architecture for learning prototypical action patterns based on neurophysiological evidence. The architecture includes primarily visual areas involved in the recognition of body movement. An overview of the architecture is illustrated in Fig. 2.2, showing the different types of neuron detectors and corresponding areas in the mammalian brain involved in the processing. Consistent with biological findings, both streams comprise a hierarchy of neural detectors that process form-motion features with increasing complexity, i.e. the size of the receptive fields and the position and scale invariance of the detectors increase along the hierarchy. The model assumes that the hierarchy is

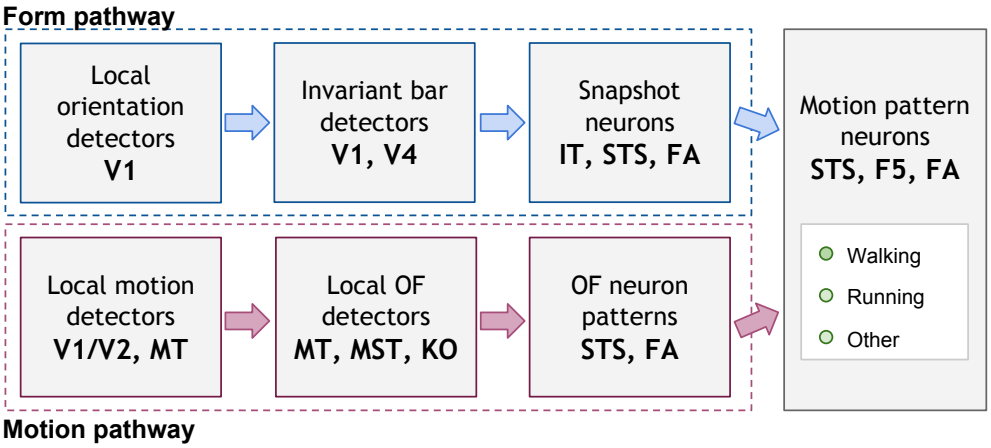



Figure 2.2: Hierarchical, two-pathway neural model for the processing of form and motion. F5, ventral premotor cortex; IT, inferior temporal cortex; KO, kinetic occipital cortex; MT, middle temporal cortex; MST, medial superior temporal cortex, OF, optic flow; STS, superior temporal sulcus; V1, primary visual cortex; V2, secondary visual cortex (prestriate cortex); V4, visual area in the extrastriate visual cortex. Adapted from (Giese and Poggio, 2003).

predominantly feedforward. While this assumption does not rule out the need for top-down signals, the assumption is based on the fact that recognition of biological motion in the STS exhibits short latencies, thus making the key role of top-down modulation unlikely for early action perception. For instance, Johansson (1976) showed that stimulus presentation times below 300 ms are sufficient for the recognition of biological motion, while Oram and Perrett (1996) observed that motion-selective neurons in the STS exhibit latencies of less than 200 ms. However, anatomical and neurophysiological studies have shown that the visual cortex is characterized by significant feedback connectivity between different cortical areas (Felleman and Van Essen, 1991; Salin and Bullier, 1995). In particular, action perception demonstrates strong top-down modulatory influences from attentional mechanisms (Thornton et al., 2002) and higher-level cognitive representations such as biomechanically plausible motion (Shiffrar and Freyd, 1990). Furthermore, although the model accounts for the biologically plausible processing of form-motion cues, it does not explain how information from the two streams is subsequently integrated as a joint percept.

## Multimodal Action Perception

It has been argued that the STS in the mammalian brain may be the basis of an action-encoding network with neurons driven by the perception of dynamic human bodies and that for this purpose it receives converging inputs from earlier visual areas from both the ventral and dorsal pathways (Beauchamp, 2005; Garcia and Grossman, 2008; Vangeneugden et al., 2009; Thirkettle et al., 2009). Neuroimaging studies have shown that the posterior STS (pSTS) shows a greater response

for audiovisual stimuli than to unimodal visual or auditory stimuli (Calvert, 2001; Beauchamp et al., 2004; Wright et al., 2003; Senkowski et al., 2011). Wright et al. (2003) conducted an event-related fMRI study showing a strong activation of the STS region in subjects evoked by both unimodal and multimodal audiovisual stimuli from an animated character, but that greatest levels of activity were elicited by audiovisual speech. In a study of actions involving the use of objects, (Beauchamp et al., 2004) observed that the pSTS and middle temporal gyrus (MTG) showed an enhanced response when auditory and visual object features were presented together with respect to the response to a single modality. Thus, the STS area is thought to be an associative learning device for linking different unimodal representations and accounting for the mapping of naturally occurring, highly correlated features such as body pose and motion, the characteristic sound of an action (Beauchamp et al., 2004; Barraclough et al., 2005) and linguistic stimuli (Belin et al., 2002; Wright et al., 2003; Stevenson and James, 2009).

These findings together suggest that multimodal representations of actions in the brain play an important role for a robust perception of complex action patterns, with the STS representing a multisensory area in the brain network for signaling the social significance of biological motion (Allison et al., 2000; Adolphs, 2003; Beauchamp, 2005; Beauchamp et al., 2008).

### Formation of Cortical Maps

It is now known that rudimentary patterns of cortical connectivity for visual processing are established early in development (see Section 3.1). However, normal visual input is required for the correct development of the visual cortex through input-driven self-organization (Hubel and Wiesel, 1962, 1967, 1970; Hubel et al., 1977). The ability of the cortex to self-organize with respect to the distribution of the inputs becomes a less prominent feature as the system stabilizes through a well-specified set of developmental stages (Nelson, 2000). However, this ability is not absent in the adult system that exhibits mechanisms of transient reorganization at a smaller scale (Stiles, 2000).

The ability of the brain to adapt to dynamic input distributions provides vital insight into how connectivity and function of the cortex are shaped and recovered from injuries. We will discuss the pillars of cortical experience-driven learning mechanisms and computational models of self-organization in Chapter 3.

## 2.2 Computational Approaches

### 2.2.1 Trends in Action Recognition

The task of human action recognition has been of strong interest for different fields of research. Artificial systems aimed to tackle complex visual tasks such as the classification of actions from videos have been extensively studied in the literature, with a large variety of models and methodologies tested on different

action benchmark datasets (Poppe, 2010). In particular, learning-based approaches have been successfully used to generalize a set of training action samples and then predict the labels of unseen samples by computing their similarity with respect to the learned action templates. Deep learning architectures motivated by biological evidence have been shown to recognize actions with high accuracy from video sequences with the use of spatiotemporal hierarchies that functionally resemble the organization of earlier areas of the visual cortex (see Section 2.1.2). Many of these models show high computational costs linked to the extraction of action features such as body posture and motion characteristics from rich streams of information (Guo et al., 2016).

In the last half decade, the emergence of low-cost depth sensing devices such as the Microsoft Kinect and ASUS Xtion Live has led to a large number of vision-based applications using depth information instead of, or in combination with, color information. This sensor technology provides depth measurements used to obtain reliable estimations of 3D human motion in cluttered environments, including a set of body joints in real-world coordinates and their orientations. Depth sensors represent a significant contribution to the field of action recognition since they address a set of limitations related to traditional 2D sensors (e.g. RGB cameras), thereby increasing robustness under varying illumination conditions and reducing computational effort for motion segmentation and body pose estimation (see Han et al. (2013) for a survey). Depth sensors have the additional advantage of avoiding privacy issues regarding the identity of the monitored person since color information is not required at any stage. However, although this approach allows to efficiently compute 3D motion features in real time, robust mechanisms for learning relevant spatiotemporal action features represent still an open question.

Contrary to fixed sensors, mobile robots may be designed to process the sensed information and undertake actions that benefit people with disabilities and seniors in a residential context (Fig. 2.3). In this context, the reliable recognition of actions and potentially dangerous behaviors such as fall events play a crucial role. There has been an increasing number of ongoing research projects aimed to develop assistive robots in smart environments for self-care and independence at home. Moreover, advanced robotic technologies may encompass socially-aware assistive solutions for interactive robot companions, able to support basic daily tasks of independent living and enhance the user experience through a more flexible human-robot interaction (e.g., gesture recognition, dialogues, and vocal commands). Recent studies support the idea that the use of socially assistive robots leads to positive effects on the senior's well-being in domestic environments (see Kachouie et al. 2014 for a review). On the other hand, the use of robotic technologies brings a vast set of challenges and technical concerns.

To cope with the dynamic nature of real-world scenarios, learning artificial systems may also be adaptive to unseen situations. In addition to detecting short-term behavior such as domestic daily actions and abnormal behavior with respect to specific action patterns, it may be of particular interest to learn the user's behavior over longer periods of time (Vettier and Garbay, 2014). In this setting, it would be desirable to collect sensory data to, e.g., perform medium- and long-

Figure 2.3: Person monitoring in a home-like environment. The humanoid robot tracks the person while performing daily activities (Parisi et al., 2016c).

term gait assessment of the person, which can be an important indicator for a variety of health problems, e.g. physical diseases and neurological disorders such as Parkinson's disease (Aerts et al., 2012). To enhance the user's experience, assistive robots may be given the capability to adapt over time to better interact with the monitored user. This would include, for instance, a more natural human-robot communication including the recognition of hand gestures and full-body actions, speech recognition, and a set of reactive behaviors based on the user's habits. In this context, interdisciplinary research aimed to address the vast set of technical and social issues regarding robots for assisted living is fundamental to provide feasible and reliable solutions in the near future.

Computational models for action recognition through multiple sensor modalities are a paramount ingredient of autonomous robots to forming robust and meaningful representations of perceived events (Ursino et al., 2014). There are numerous advantages from the multimodal processing of sensory inputs conveyed by rich and uncertain information streams. For instance, the integration of stimuli from different sources may be used to attenuate noise and remove ambiguities from converging or complementary inputs. Multimodal representations have been shown to improve robustness in the context of action recognition, human-robot interaction, and sensory-driven motor behavior (Kachouie et al., 2014; Noda et al., 2014; Bauer et al., 2015). However, multisensory inputs must be integrated in an appropriate way so that they result in a reliable cognitive experience aimed to trigger adequate behavioral responses. Consequently, the question of how to effectively acquire, process, and bind multimodal knowledge from rich information streams represents a fundamental issue still to be fully investigated.

### 2.2.2 Learning to Recognize Actions

Machine learning and neural network techniques processing multi-cue features from natural images have shown motivating results for classifying a set of training actions. Typically, baselines of performance in terms of classification accuracy are provided by evaluating the approach with publicly available action datasets. Examples of common public datasets are the KTH human motion dataset (Schuldt et al., 2004), the Weizmann human action dataset (Gorelick et al., 2005), the UCF sports action dataset (Rodriguez et al., 2008), and the CAD-60 with depth map video sequences (Sung et al., 2012).

Xu et al. (2012) presented a system for action recognition using dynamic poses by coupling local motion information with pose in terms of skeletal joint points. They generated a codebook of dynamic poses from two RGB action benchmarks (KTH and UCF-Sports), and then classified these features with an Intersection Kernel Support Vector Machine. Jiang et al. (2012) explored a prototype-based approach using pose-motion features in combination with tree-based prototype matching via hierarchical clustering and look-up table indexing for classification. They evaluated the algorithm on the Weizmann, KTH, UCF Sports, and CMU action benchmarks. To be noted is that although these two approaches use pose-motion cues to enhance classification accuracy with respect to traditional single-cue approaches, they do not take into account an integration function that learns order-selective prototypes of joint pose-motion representations of action segments from training sequences. Furthermore, these classification algorithms can be susceptible to noise which may occur during live recognition.

Learning systems using depth information from low-cost sensors have been increasingly popular in the research community encouraged by the combination of computational efficiency and robustness to light changes in indoor environments. In recent years, a large number of applications using 3D motion information has been proposed for human activity recognition such as classification of full-body actions (Faria et al., 2014; Shan and Akella, 2014), fall detection (Rougier et al., 2011; Parisi and Wermter, 2013), and recognition of hand gestures (Suarez and Murphy, 2012). A vast number of depth-based methods has used a 3D human skeleton model to extract relevant action features for the subsequent use of a classification algorithm. For instance, Sung et al. (2012) combined the skeleton model with Histogram of Oriented Gradient (HOG) features and then used a hierarchical maximum entropy Markov model to classify 12 different actions. The learning model used a Gaussian mixture model to cluster and segment the original training data into activities.

Using the same action benchmark for the evaluation, Shan and Akella (2014) used action templates computed from 3D body poses to train multiple classifiers: Hidden Markov Model, Random Forests, k-Nearest Neighbor, and Support Vector Machine (SVM). Faria et al. (2014) used a dynamic Bayesian Mixture Model designed to combine multiple classifier likelihoods and compute probabilistic body motion. Zhu et al. (2014) evaluated a set of spatiotemporal interest point features from raw depth map images to classify actions with an SVM. Experiments were

conducted also using interest points in combination with skeleton joint positions and color information, obtaining better results. However, the authors also showed that noisy depth data and cluttered background have a significant impact on the detection of points of interest, and that actions without much motion are not well recognized.

Computational models inspired by the hierarchical organization of the visual cortex (see Section 2.1.2) have become increasingly popular for learning complex visual patterns such as action sequences from video (Giese and Poggio, 2003; Layher et al., 2013). In particular, neural network approaches with deep learning architectures have produced state-of-the-art results on a set of benchmark datasets containing daily actions (e.g. Baccouche et al. 2011; Jain et al. 2015; Jung et al. 2015). Typically, visual models using deep learning comprise a set of convolution and pooling layers trained in a hierarchical fashion for obtaining action feature representations with an increasing degree of abstraction (see Guo et al. (2016) for a recent survey). This processing scheme is in agreement with neurophysiological studies supporting the presence of functional hierarchies with increasingly larger spatial and temporal receptive fields along cortical pathways.

The above-described methods are trained by a batch learning scheme, and thus assuming that all the training samples and sample labels are available during the training phase. However, an additional strong assumption is that training samples, typically represented as a sequence of feature vectors extracted from video frames, are well segmented so that ground-truth labels can be univocally assigned. Therefore, it is usually the case that raw visual data collected by sensors must undergo an intensive pre-processing pipeline before training a model. These pre-processing stages are mainly performed manually, thereby hindering the automatic, continuous learning of actions from live video.

From a multimodal perspective, a number of computational models have been proposed aiming to effectively integrate multisensory information, in particular audiovisual input. These approaches typically use unsupervised learning for obtaining visual representations of the environment and then link these features to auditory cues. For instance, Vavrečka and Farkaš (2014) presented a connectionist architecture that learns to bind visual properties of objects (spatial location, shape and color) to proper lexical features. These unimodal representations are bound together based on the co-occurrence of audiovisual inputs using a self-organizing neural network (see Section 3.2). Similarly, Morse et al. (2015) investigated how infants may map a name to an object and how body posture may affect these mappings. The computational model is driven by visual input and learns word–to–object mappings through body posture changes and online speech recognition. Unimodal representations are obtained with neural network self-organization and multimodal representations develop through the activation of unimodal modules via associative connections.

The development of associations between co-occurring stimuli for multimodal binding has been strongly supported by neurophysiological evidence (Fiebelkorn et al., 2009). However, these approaches do not naturally scale up to learn more complex spatiotemporal patterns such as action–word mappings. In fact, action

words do not label actions in the same way that nouns label objects (Gentner, 1982). While nouns typically refer to objects that can be perceived as distinct units, action words refer instead to spatiotemporal relations within events that may be performed in many different ways with high spatial and temporal variance. Thus, further work is required to address the learning of multimodal representation of spatiotemporal inputs for obtaining robust action–word mappings.

### 2.2.3 Body Motion Assessment

The analysis and assessment of human body motion have recently attracted significant interest in the healthcare community with many application areas such as physical rehabilitation, diagnosis of pathologies, and assessment of sports performance. In this context, the correctness of postural transitions is paramount during the execution of well-defined physical routines, since inaccurate movements may significantly reduce the overall efficiency of the movement and increase the risk of injury (Kachouie et al., 2014). For instance, in the case of weight-lifting training, correct postures improve the mechanical efficiency of the body and allow the athlete to achieve higher effectiveness during training sessions. Similarly, in the healthcare domain, the correct execution of physical rehabilitation routines is crucial for patients to improve their health condition (Velloso et al., 2013a).

Human proprioception may not be sufficient to spot movement mistakes. Thus, expert trainers observing the movement can give the trainee proficient feedback for timely improving the quality of the performance and avoiding persistent inaccuracies. However, it is not the case that a personal trainer is always available to assess the quality of movements during their execution. Therefore, there is a strong motivation to develop automatic systems able to detect mistakes during the performance of well-defined routines for providing feedback in real time.

While the aim of action recognition is to categorize a set of distinct classes by extrapolating inter-class spatiotemporal differences, action assessment requires instead a model to capture intra-class dissimilarities that allow to express a measurement on how much an action follows its learned template. In this setting, efficient approaches to learn spatiotemporal templates for computing intra-class dissimilarities have remained an open issue. Common computational bottlenecks are the robust extraction of body features from video streams and the definition of suitable metrics aimed to compare two actions in terms of their spatiotemporal structure. The former issue has been partly addressed with the use of depth sensors that allow the efficient tracking of human motion and the estimation of a 3D skeleton model. On the other hand, effective methods for the computation of a similarity measure between two actions still represent a major challenge.

Automatic systems for the visual assessment of body motion have been previously investigated for applications mainly focused on physical rehabilitation and sports training. For instance, Chang et al. (2011) proposed a physical rehabilitation system for young patients with motor disabilities using a Kinect sensor. The idea was to assist the users while performing a set of simple movements necessary to improve their motor proficiency during the rehabilitation period. Users

were instructed by a therapist on how to perform the movements. During the autonomous execution, visual hints were shown to users to motivate the performance of the routines. Although experimental results have shown improved motivation for users using visual hints, only movements involving the arms at constant speed were considered. Furthermore, the estimation of real-time feedback in order to enable the user to spot and correct mistakes was not considered.

Similarly, Su (2013) proposed the estimation of feedback for Kinect-based rehabilitation exercises by comparing performed motion with a pre-recorded execution by the same person. The comparison was carried out on sequences using dynamic time warping (DTW) and fuzzy logic with the Euclidean distance as a similarity measure. The evaluation of the exercises was based on the degree of similarity between the current sequence and a correct sequence. The system provided qualitative feedback on the similarity of body joints and execution speed, but it did not suggest the user how to correct the movement.

Paiement et al. (2014) proposed a method for assessing the quality of gait from sequences of people on stairs. As a measure of quality, Kinect-based body poses were compared to learned normal occurrences of a movement from a statistical model. The likelihood of a model for describing the current movement was computed frame-by-frame over a sequence of postures and motion speed. The system triggered an alarm if the current movement differed from the correct movement template. For this purpose, a proper threshold must be empirically chosen to decide the degree of tolerance with respect to the template. Although this method represents a useful application for detecting abnormal behavioral patterns, it does not provide any hints on how to correct motion mistakes.

Velloso et al. (2013b) investigated qualitative action recognition with a Kinect sensor for specifying the correct execution of movements, detecting mistakes, and providing feedback to the user. A baseline was created by asking the users to perform a routine ten times, from which individual repetitions were manually segmented. Hidden Markov Models were trained with tuples containing the joint angles and the timestamp for individual exercises. Similar to Chang et al. (2011) and Su (2013), the system was tested only on arm movements, in this case for dumbbell lifting. A strong limitation of this approach is that the correct duration and motion intensity of movements were computed by using the timestamp from body joint estimation. Therefore, although the system provides feedback to correct body posture in terms of joint angles, it does not provide any robust feedback on temporal discrepancies.

For the assessment of human motion in sports, Pirsiavash et al. (2014) predicted scores of performed movements from annotated footage. The system compared the gradient for each body joint with a regression model from spatiotemporal pose features to scores obtained from expert judges. Feedback is provided in terms of which joints should be changed to obtain the maximum score. Different from the previously discussed approaches, this method extracts body features from RGB sequences. Thus, the estimation of body joints is not as robust as the 3D skeleton model using a depth sensor. Experimental results showed that the system predicted scores better than non-expert humans but significantly worse than expert judges.

While the correct execution of well-defined movements plays a crucial role in physical rehabilitation and sports, artificial learning systems for assessing the quality of actions and providing feedback for correcting inaccurate movements have remained an open issue in the literature.

### 2.2.4 Abnormal Event Detection

Falls represent a major concern in the public health care domain, especially among the elderly population. According to the World Health Organization, fall-related injuries are common among older persons and represent the leading cause of pain, disability, loss of independence and premature death[1]. Although fall events do not necessarily cause a fatal injury, fallen people may be unable to get up without assistance, thereby resulting in *long lie* time complications such as hypothermia, dehydration, bronchopneumonia, and pressure sores (Tinetti et al., 1993). Moreover, fear of falling has been associated with a decreased quality of life, avoidance of activities, and mood disorders such as depression (Scheffer et al., 2008).

As a response to increasing life expectancy, research has been conducted to provide technological solutions for supporting living at home and smart environments for assisted living. The motivation of assistive fall systems is the ability to promptly report a fall event and by this enhancing the person's safety perception and avoiding the loss of confidence due to functional disabilities. Recent systems for elderly care aim mostly to detect hazardous events such as falls and allow the monitoring of physiological measurements (e.g. heart rate, breath rate) using wearable sensors to detect and report emergency situations in real time (Kaluza et al., 2013; Vettier and Garbay, 2014). Vision-based fall detection is currently the predominant approach due to the constant development of computer vision techniques that yield increasingly promising results in both experimental and real-world scenarios. While the number of advantages introduced by low-cost depth sensors is significant in terms of body motion and posture estimation, these approaches are characterized by a number of issues that may prevent them from operating in real-world environments. For instance, their operation range (distance covered by the sensor) is quite limited (between 0.8 m and 5 m), as well as their limited field of view, thereby requiring a mobile or multi-sensor scenarios to monitor an extensive area of interest.

Lee and Mihailidis (2005) proposed a vision-based method with a ceiling camera for monitoring falls at home. The authors considered falls as lying down in a stretched or tucked position. The system accuracy was evaluated with a pilot study using 21 subjects consisting of 126 simulated falls. Personalized thresholds for fall detection were based on the height of the subjects. The system detected fall events with 77% accuracy and had a false alarm rate of 5%. Miaou et al. (2006) proposed a customized fall detection system using an omni-camera for capturing 360-degree scene images. Falls were detected based on the change of the ratio of

[1]World Health Organization: Global report on falls prevention in older age – `http://www.who.int/ageing/publications/Falls_prevention7March.pdf`

people's height and width. Two scenarios were used for the detection: with and without considering user health history, for which the system showed 81% and 70% accuracy respectively.

In a multi-camera scenario, Cucchiara et al. (2007) presented a vision system with multiple cameras for tracking people in different rooms and detecting falls based on a hidden Markov model (HMM). People tracking was based on geometrical and color constraints and then sent to the HMM-based posture classifier. Four main postures were considered: walking, sitting, crawling, and lying down. When a fall was detected, the system triggered an alarm via SMS to a clinician's PDA with a link to live low-bandwidth video streaming. Experiments showed that occlusions had a strong negative impact on the system's performance. Hazelhoff et al. (2008) detected falls using two fixed perpendicular cameras. The foreground region was extracted from both cameras and the principal components (PCA) for each object were computed to determine the direction of the main axis of the body and the ratio of the variances. Using these features, a Gaussian multi-frame classifier was used to recognize fall. In order to increase robustness and mitigate false positives, the position of the head was taken into account. The system was evaluated also for partially occluded people. Experiments showed real-time performance with an 85% overall detection rate.

Rougier et al. (2011) presented a method for fall detection by analysing human shape deformation in depth map image sequences. Falls were detected from normal activities using a Gaussian mixture model with 98% accuracy. The overall system performance increased when taking into account the lack of significant body motion after the detected fall event. Liu et al. (2010) detected falls considering privacy issues, thereby processing only human silhouettes without featural properties such as the face. A k-nearest neighbor (kNN) algorithm was used to classify the postures using the ratio and difference of human body silhouette bounding box height and width. Recognized postures were divided into three categories: standing, temporary transition, and lying down. Experiments with 15 subjects showed a detection accuracy of 84.44% on fall detection and lying down events. Diraco et al. (2010) addressed the detection of falls and the recognition of several postures with 3D information. The system used a fixed time-of-flight camera that provided robust measurements under different illumination settings. Moving regions with respect to the floor plane were detected applying a Bayesian segmentation to the 3D point cloud. Posture recognition was carried out using the 3D body centroid distance from the floor plane and the estimated body orientation. The system yielded promising results on synthetic data with threshold-based clustering for different centroid's height thresholds.

Most of the above-described approaches rely on predefined threshold values to detect abnormal behavior. Furthermore, reported experiments were conducted in highly controlled environments with fixed vision sensors. Whether these approaches would account for the robust detection of abnormal behavior if embedded in mobile robot platforms, is questionable.

### 2.2.5 Assistive Robotics

Mobile robots have been characterized by a constant development for *aging at home* scenarios. In contrast to fixed sensors, mobile assistive robots may be designed to process the sensed information and undertake actions that benefit people with disabilities and seniors in a residential context. In fact, the mobility of robots represents a big benefit for non-invasive monitoring of users, thereby better addressing fixed sensors' limited field of view, blind spots, and occlusions. Despite different functional perspectives concerning elderly care and user needs (e.g. rehabilitation,social robotics), there is a strong affinity regarding the intrinsic challenges and issues needed to operate these systems in real-world scenarios. For instance, the use of mobile robots may be generally combined with ambient sensors embedded in the environment (e.g. cameras, microphones) to enhance the agent's perception and increase robustness under in-the-wild conditions. On the other hand, complementary research efforts has been conducted on the deployment of stand-alone mobile robot platforms, able to sense and navigate the environment by relying exclusively on onboard sensors.

In particular when operating in natural environments, the robust and efficient processing of multimodal information plays a key role to perceive human activity. Research efforts have been made towards robots exploiting multi-sensory integration to improve HRI capabilities. For instance, Lacheze et al. (2009) used auditory information to recognize objects that were partially occluded and thus difficult to detect by vision only. Sanchez-Riera et al. (2009) presented a scenario with a robot companion that performs audio-visual fusion for multimodal speaker detection. The system targeted multiple speakers in a domestic environment processing information from two microphones and two cameras mounted on a humanoid robot. Martinson (2014) introduced a robot with a navigational aid for visually impaired people using a mobile robot platform. The system used depth information to detect other people in the environment and avoid dynamic obstacles. The system communicated to the person the direction of motion to reach the goal destination via a tactile belt around the waist.

For abnormal behavior detection, promising experimental results have been obtained by combining mobile robots and 3D information from depth sensors. This approach overcomes limitations in the operation range of sensors while preserving reduced computational power for real-time characteristics. Mundher and Zhong (2014) proposed a mobile robot with a Kinect sensor for fall detection based on floor-plane estimation. The robot tracks and follows the user in an indoor environment, and can trigger an alarm in case of a detected fall event. The system recognizes two gestures to start and stop a distance-based *user-following* procedure, and three voice commands to enable fall detection and call for help in case of a fall. Volkhardt et al. (2013) presented a mobile robot to detect fallen persons, i.e. a user already lying on the floor. The system segments objects from the ground plane and layers them to address partial occlusions. A classifier trained on positive and negative examples is used to detect object layers as a fallen human. Experiments reveal that the overall accuracy of the system is strongly dependent

on the type of extracted features and the classifier.

Additional challenges conveyed by the use of mobile robots for detecting fall events may comprise the tolerance of noise in a moving sensor scenario (Parisi et al., 2015a), the robust tracking of occluded persons (Martinson, 2014), and effective navigation strategies for following and finding people in domestic environments (Volkhardt and Gross, 2013). Multimodal systems embedded in mobile robots that remain operative under situations of uncertain sensory information, e.g. temporary unavailability of one of the modalities, represent an enticing milestone for assistive robots and are still to be extensively investigated. In fact, the main challenge is characterized by the ability of artificial agents to robustly integrate available multi-sensory cues from uncertain information streams, thereby yielding robust perceptual experience aimed to trigger an adequate motor behavior in highly dynamic environmental conditions.

## 2.3 Summary

Humans possess an astonishing ability to promptly process complex visual stimuli such as body motion patterns, exhibiting a high tolerance to sensory distortions and temporal variance (Blake and Shiffrar, 2007). The underlying neural mechanisms for action perception have been extensively studied, comprising cortical areas with a hierarchy of spatiotemporal receptive fields for processing body motion cues with increasing complexity of representation (Taylor et al., 2015; Hasson et al., 2008; Lerner et al., 2011), i.e. higher-level areas process information accumulated over larger temporal windows with increasing invariance to the position and the scale of stimuli. The brain integrates multisensory information to provide a robust perceptual experience (Stein and Meredith, 1993; Ernst and Bülthoff, 2004; Stein et al., 2009), thereby yielding the efficient processing of motion patterns also in situations of highly degraded stimuli and uncertainty (Neri et al., 1998; Thornton et al., 1998; Poom and Olsson, 2002). Therefore, the study of the biological mechanisms for action perception is fundamental for the development of artificial systems aimed to address the robust recognition of actions in real-world scenarios.

To tackle the visual recognition of actions, learning-based approaches typically generalize a set of labeled training action samples and then predict the labels of unseen samples by computing their similarity with respect to the learned action templates. Simplified models of brain areas processing visual cues have been proposed as a stepping stone to numerous artificial systems dealing with the detection and classification of biological motion (Giese and Poggio, 2003; Layher et al., 2014). In particular, deep neural network architectures have been shown to recognize actions with high accuracy from video sequences with the use of spatiotemporal hierarchies that functionally resemble the organization of earlier areas of the visual cortex (Guo et al., 2016). However, despite recent research efforts in machine learning and neural network, the question remains open of how to better process extracted body features for effectively learning the complex dynamics of actions in real-world scenarios. For instance, the reliable classification of actions

may be hindered by noisy and missing body joints caused by systematic sensor errors or temporary occluded body parts (Parisi and Wermter, 2013). Nevertheless, a robust, noise-tolerant system should also operate under such adverse conditions.

In the next chapters, we propose a set of neurobiologically-motivated neural network architectures for action recognition from depth map videos. For this purpose, we take into account different aspects of biological action perception and multimodal integration for enhancing human-robot interaction and triggering robust sensory-driven robot behavior in dynamic environments.

# Chapter 3

# Computational Models of Self-Organization

Experience-driven development plays a crucial role in the brain (Nelson, 2000), with topographic maps being a common feature of the cortex for processing sensory inputs (Willshaw and von der Malsburg, 1976). Different models of neural self-organization have been proposed to resemble the dynamics of basic biological findings on Hebbian-like learning and map plasticity. In this chapter, we provide an overview of experience-driven self-organization and describe a set of well-studied computational models of self-organizing systems. In particular, we review the main properties, functionality and potential drawbacks of self-organizing neural networks with feedforward competitive layers, growing models for adapting to dynamic input distributions, and networks with recurrent connectivity for learning latent spatiotemporal relations of the input.

## 3.1 Experience-driven Self-Organization

### 3.1.1 Introduction

It was first found by Mountcastle (1957) and Hubel and Wiesel (1962) that certain single neural cells in the brain of cats respond selectively to specific sensory stimuli. At that time, a number of experiments were conducted showing that altering the visual environment leads to drastic changes in the organization of the cat's visual cortex (Hubel and Wiesel, 1962, 1967, 1970; Hubel et al., 1977). Most cells in the cortex develop preferences for particular orientations, while they do not respond well to the other orientations (Blakemore and Cooper, 1970; Blakemore and Van Sluyters, 1975; Hirsch and Spinelli, 1970; Sengpiel et al., 1999). These studies suggested that visual inputs are crucial for normal cortical organization, since the cortex tunes itself to the distribution of visual inputs. Additional studies showed that brain maps develop through self-organization of input connections from the thalamus and are shaped by visual experience (Shatz, 1992).

The concept of input-driven self-organization has been challenged from a bi-

ological perspective. In particular, it has been argued that since the process of self-organization requires time, the animal would not be able to properly react to visual input until the process is at an advanced stage. Furthermore, with self-organizing structures critically depending on the available input, statistically relevant patterns may not be the most relevant information for the organism and its survival. Today, it is well known that visual development in nature is highly stable, with the visual cortex of many animals being partially organized already at birth, or at least at the moment of first eye opening (O'Donovan, 1999; Wong, 1999). However, the information available in the genome to achieve a fixed genetic blueprint is not sufficient. Therefore, it has been hypothesized that while intrinsic factors such as genes or molecular gradients drive the initial development for granting a rudimentary level of performance from the start, extrinsic factors such as sensory experience completes this process for achieving higher complexity and performance (Hirsch and Spinelli, 1970; Hirsch, 1985; Shatz, 1990, 1996; Sur and Leamey, 2001).

The ability of the brain to adapt to changes in its environment - referred to as developmental plasticity - provides vital insight into how connectivity and function of the cortex are shaped and recovered from injuries. For instance, studies showed that while rudimentary patterns of connectivity in the visual system are established early in development, normal visual input is required for the correct development of the visual cortex. The work of Hubel and Wiesel (1967) on the emergence of ocular dominance columns also highlighted the importance of timing of experience on the development of normal patterns of cortical organization. The visual experience of newborn kittens was experimentally manipulated to study the effects of varied input on brain organization. As a result, they observed that the disruption of cortical organization was more severe when deprivation of the visual input began prior to 10 weeks of age, while no changes were observed in adult animals. These experiments presented evidence that neural patterns of cortical organization can be modulated by external environmental factors at least for a period early in development. These findings together demonstrated that optimal patterns of cortical organization are not fixed, but rather depend on local patterns of connectivity that may be altered by remote changes in structure or input.

The most well-known theory describing the mechanisms of synaptic plasticity for the adaptation of neurons during the learning process was first proposed by Hebb (1949), postulating that when one neuron drives the activity of another neuron, the connection between these neurons is strengthened. More specifically, the Hebb's rule states that the repeated and persistent stimulation of the postsynaptic cell from the presynaptic cell leads to an increased synaptic efficacy. Subsequent studies indicated that in addition to a Hebbian-like synaptic potentiation, a mechanism of depression between two neurons that are not sufficiently co-active is necessary (Sejnowski, 1977). Throughout the process of development, neural systems stabilize to shape optimal functional patterns of neural connectivity. Plasticity becomes a less prominent feature as the system stabilizes through a well-specified set of developmental stages, however, it is not absent from the adult system, yielding the transient capacity for plastic reorganization at a smaller scale (for a review on

neural plasticity, see Stiles, 2000).

These results together show that plasticity, in terms of the ability to adapt and respond to both intrinsic and extrinsic factors, plays a crucial role in the normal development of neural systems, and in particular in the postnatal period.

### 3.1.2 Artificial Self-Organizing Networks

Computational models implementing experience-driven self-organization have been used to demonstrate that the preferences and their organization can result from statistical learning with the nonlinear approximation of the distribution of visual inputs. These models have contributed to better understand the underlying neural mechanisms for the development of cortical organization.

As presented in Section 2.1, a vast body of literature has shown that the actual organization of the visual cortex and other cortical areas is quite complex, with an incredibly large number of sheets of neurons and topographically mapped connections between them. Therefore, most of the models implementing biologically motivated self-organization have focused on small-scale systems with simplified mathematical models governing the formation of topographic maps. Different artificial self-organizing neural networks have been proposed to resemble the dynamics of basic biological findings on Hebbian-like learning and map plasticity. In the study of neural networks and cognitive functions, the Hebb's associative rule (Hebb, 1949) is seen as the basis of unsupervised learning.

The goal of the self-organizing learning is to cause different parts of the network to respond similarly to certain input samples starting from an initially unorganized state. Typically, during the training phase these networks build a map through a competitive process, also referred to as *vector quantization*, so that a set of neurons represent prototype vectors encoding a submanifold in the input space. In doing so, the network learns significant *topological relations* of the input without supervision.

#### Vector Quantization

Vector quantization (VQ) is a quantization technique that models probability density functions via the distribution of prototype vectors. The optimal quantization of a vector space was first introduced by Dirichlet (1850) with the so-called Dirichlet tessellation in two- and three-dimensional spaces, and subsequently extended to spaces of arbitrary dimensionality by Voronoi (1907).

The standard VC technique encodes a data manifold $V \subseteq \mathbb{R}^d$ using a finite set of protoype (or codebook) vectors $\mathbf{w}_i \in \mathbb{R}^d$. For simplicity, we consider input data samples in terms of $d$-dimensional Euclidean vectors. A vector-valued input $\mathbf{v} \in V$ can be then described by the best-matching reference vector $\mathbf{w}_{i(\mathbf{v})}$ for which the distortion (or quantization error) $d(\mathbf{v}, \mathbf{w}_{i(\mathbf{v})})$, e.g. the squared error $\|\mathbf{v} - \mathbf{w}_{i(\mathbf{v})}\|^2$, is minimized. This procedure divides the manifold $V$ into a number of subregions

$V_i$, referred to as Voronoi regions, such that:

$$V_i = \{\mathbf{v} \in V : \|\mathbf{v} - \mathbf{w}_i\|^2 \leq \|\mathbf{v} - \mathbf{w}_j\|^2 \quad \forall i \neq j\}, \tag{3.1}$$

with all vectors in $V$ having a distance to $\mathbf{w}_i$ not greater than their distance to $\mathbf{w}_j$.

If $p(\mathbf{v})$ is the probability density of $\mathbf{v}$ and $b$ is the index of the best-matching vector of $\mathbf{v}$, then the mean quantization error $E$ is defined as

$$E = \int_V \|\mathbf{v} - \mathbf{w}_b\|^2 p(\mathbf{v}) dV, \tag{3.2}$$

where $dV$ is a volume differential of $V$, and $E$ is an energy function that can be minimized by a gradient-descent procedure. Kohonen (1991) showed that this highly nonlinear function converges to a local minimum.

This VQ procedure can be applied to self-organizing networks for performing a nonlinear approximation of the distribution of the input through statistical learning, so that a continuous input space is mapped onto a discrete feature space of neurons by a process of neural competition (see Section 3.2).

## Topology Preservation

The preservation of the topology of the input in terms of neighborhood relations is a useful and well-studied property of self-organizing networks (Martinetz, 1993; Martinetz et al., 1993; Goodhill and Sejnowski, 1997). In general, a network can perform a perfect topology-preserving mapping only if the dimensionality of the map space reflects the (intrinsic) dimensionality of the input space.

A network $G$ consists of a number $n$ of neurons and receives input samples from a data manifold $V \subset \mathbb{R}^d$, with every neuron $i \in G$ having a synaptic weight vector $\mathbf{w}_i \in \mathbb{R}^d$. The representation of $V$ in $G$ is defined by the mapping $V_G(\Psi_{V\to G}, \Psi_{G\to V})$ defined as

$$V_G = \begin{cases} \Psi_{V\to G} : V \to G; v \in V \implies i^*(v) \in G \\ \Psi_{G\to V} : G \to V; i \in G \implies \mathbf{w}_i \in V \end{cases} \tag{3.3}$$

where $i^*(v)$ is the map neuron with weight vector $w_{i^*(v)}$ closest to $v$. A connection matrix $C$ that stores connections between neurons is defined on $G$ as:

$$C_{ij} = \begin{cases} 1 & \text{if } i \text{ and } j \text{ are connected} \\ 0 & \text{otherwise} \end{cases} \tag{3.4}$$

As formalized by Martinetz (1993), there is a perfect topology-preserving mapping between the input manifold and a network if and only if connected neurons $i, j$ that are adjacent in $G$ have weight vectors $\mathbf{w}_i, \mathbf{w}_j$ adjacent in $V$.

## Taxonomy of Self-Organizing Networks

Although self-organizing models are governed by similar principles with the aim to cause different parts of a network to respond similarly to certain input patterns,

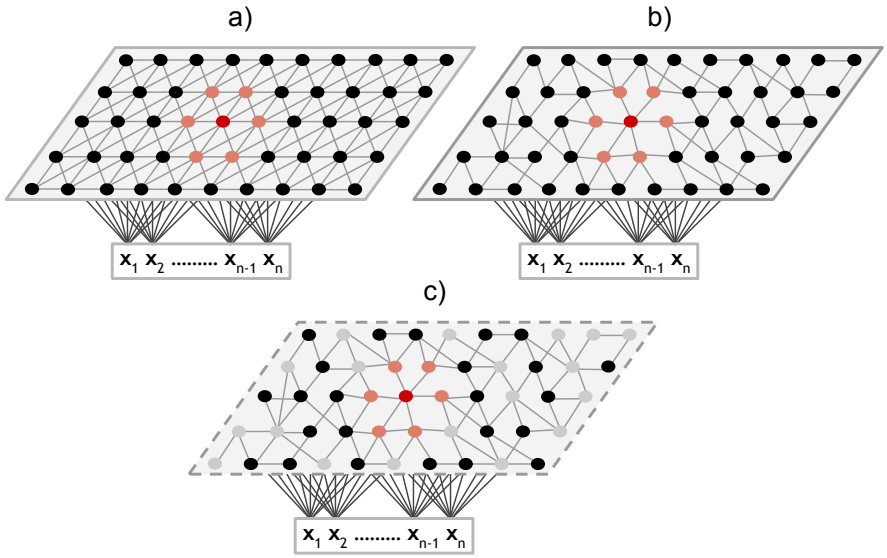



Figure 3.1: Different types of self-organizing networks: (a) competitive layer with fixed topology, (b) competitive layer with adaptive topology, and (c) growing network with a variable number of neurons and adaptive topology.

the models may significantly differ in terms of the shape of the maps, the development of neighborhood relations, and the learning procedure. Different examples of self-organizing networks are illustrated in Fig. 3.1, showing a competitive layer with fixed topology, a competitive layer with adaptive topology, and a growing network with a variable number of neurons and adaptive topology. Generally, the competitive layer is fully connected to the input. The number of neurons and the shape of the map are related to the representational power of the model. For instance, in a network with fixed topology, each neuron has a fixed set of neighbors, and this may constrain the mapping accuracy. Static networks in which the number of neurons must be specified a priori, i.e., before the training phase starts, have shown compelling performance in a huge number of data-analysis tasks. On the other hand, the ability of a network to create new neurons (and remove unused ones) for adapting to novel incoming signals is crucial for learning non-stationary input distributions.

In the next sections, we introduce a set of well-established self-organizing neural networks that have been successfully applied to a large variety of learning tasks such as clustering and novelty detection. Furthermore, for each of these self-organizing networks, we will show extended variants with recurrent connectivity for learning the spatiotemporal structure of the input.

## 3.2 Feedforward Self-Organizing Networks

In this section, we introduce four self-organizing networks: the self-organizing map (SOM), the neural gas (NG), and growing neural gas (GNG), and the growing when

required (GWR). We refer to these networks as feedforward to differentiate them from their recurrent variants, in which the formation of the map depends on the current input plus previously activated map patterns. The number of neurons in the SOM and the NG is fixed beforehand and cannot be changed over time, while growing models have the ability to create (and remove) neurons to better fit the (dynamic) distribution of the input.

### 3.2.1 Self-Organizing Feature Maps

The self-organizing map (SOM, Kohonen, 1990) is an unsupervised learning algorithm that nonlinearly projects a high-dimensional input space onto a low-dimensional discretized (typically two-dimensional) representation. It consists of a layer with competitive neurons connected to adjacent neurons by a neighborhood relation (Fig. 3.1.a). Similar to the VQ technique, the SOM represents the input distribution using a finite set of prototype neurons. The number $n$ of neurons must be decided a priori (i.e., before the training phase starts), and the topology of the network (neighborhood relation) is fixed. The network learns by iteratively reading each vector-valued training sample and organizes the neurons so that they describe the domain space of input.

Each neuron $j$ of the network is associated with a $d$-dimensional synaptic weight vector:

$$\mathbf{w}_j = [\mathbf{w}_{j,1}, \mathbf{w}_{j,2}, ..., \mathbf{w}_{j,d}], \quad j = 1, 2, .., n. \tag{3.5}$$

For an input vector $\mathbf{x} = [\mathbf{x}_1, ..., \mathbf{x}_d]$ presented to the network, the best-matching unit (BMU) $b$ for $\mathbf{x}$ is selected by the smallest Euclidean distance as:

$$b(\mathbf{x}) = \arg\min_j \|\mathbf{x} - \mathbf{w}_j\|, \quad j = 1, 2, .., n. \tag{3.6}$$

The weights of the winner neuron $b$ and those of the neurons within a neighborhood $H_b$ on the neuron grid are modified according to the update rule:

$$\mathbf{w}_j(t+1) = \mathbf{w}_j(t) + \epsilon(t) \cdot h_b(t) \cdot [\mathbf{x}(t) - \mathbf{w}_j(t)], \tag{3.7}$$

where $b$ is the best matching unit (Eq. 3.6), and $h_b(t)$ is a Gaussian neighborhood function:

$$h_b(t) = \exp\left(\frac{-\|r_b - r_i\|^2}{2\sigma^2(t)}\right), \tag{3.8}$$

where $r_b$ is the location of $b$ on the map grid and $\sigma(t)$ is the neighborhood radius at time $t$.

The learning rate $\epsilon(t)$ is a decreasing function of time between $[0, 1]$, for instance the exponentially decreasing learning rate function defined as:

$$\epsilon(t) = \epsilon_0 \left(\frac{\epsilon_T}{\epsilon_0}\right)^{\frac{t}{T}}, \tag{3.9}$$

where $\epsilon_0$ is the initial learning rate, $\epsilon_T$ the final learning rate, $t$ is time and $T$ the training length in terms of epochs.

The competitive network can be trained with a batch variant of the SOM algorithm, where the whole data set to the network before any adjustments are made. The updating is carried out by replacing the prototype vector $\mathbf{w}_j$ with a weighted average over the samples:

$$\mathbf{w}_j(t+1) = \frac{\sum_{i=1}^{n} h_{b(i)}(t)\mathbf{x}_i}{\sum_{i=1}^{n} h_{b(i)}(t)}. \tag{3.10}$$

The batch version of the learning algorithm has shown fast performance for a variety of learning tasks (Kohonen, 2013). However, a potential drawback of the SOM is the fixed topology that may constrain mapping accuracy. For a more detailed discussion on this matter and examples of implementation, we refer the reader to a recent review of the SOM by Kohonen (2013).

### Neural Gas

The neural gas (NG, Martinetz, 1993) is an unsupervised algorithm for finding optimal data representations based on prototype vectors. Similarly to the SOM, the NG consists of a competitive layer with a fixed number $N$ of neurons fully connected to the input (Fig. 3.1.b). In the NG, however, the topology of the network is not fixed but rather develops throughout the learning process.

At each time step $t$, a vector-valued sample $\mathbf{x}$ is randomly chosen from a distribution $p(\mathbf{x})$ and presented to the network. Then, the distance order of the prototype neurons to the given sample $\mathbf{x}$ is determined so that each neuron $(k = 0, ..., n-1)$ is adapted according to:

$$\mathbf{w}_{i_k}(t+1) = \mathbf{w}_{i_k}(t) + \epsilon \cdot \exp(-k/\lambda) \cdot [\mathbf{x} - \mathbf{w}_{i_k}(t)], \tag{3.11}$$

where $i_0$ denotes the index of the closest neuron, $i_1$ the index of the second closest neuron and $i_{n-1}$ the index of the neuron most distant from $\mathbf{x}$, $\epsilon$ is the adaptation step size and $\lambda$ is the neighborhood range. The learning rate $\epsilon$ and the range $\lambda$ decrease with increasing $t$.

After a sufficient number of adaptation steps, the neurons will cover the data space with a minimum representation error. In fact, since the structure of the network is not constrained by a fixed topology, the NG has been shown to minimize the quantization error. The adaptation step of the NG (Eq. 3.11) can be interpreted as gradient descent on the energy function $E$ (Eq. 3.2).

Like in the SOM, in the NG the number of neurons must be decided a priori and cannot change over time. Furthermore, the number of epochs must also be pre-defined, with the neighborhood size decreasing as time increases. Therefore, these models have been shown to be unsuitable for continuous learning or the learning of non-stationary input distributions.

## 3.2.2 Growing Self-Organizing Networks

Growing networks represent one approach to address the limitations of static self-organizing networks by creating (or removing) neurons to support the correct for-

mation of topological maps. In the Growing Neural Gas (GNG) neurons are added at fixed intervals to minimize local errors, while the Growing When Required (GWR) adds neurons whenever the activity of the network is not sufficiently high. In this section, we provide a comparison between these growing models.

### Growing Neural Gas (GNG)

The GNG algorithm proposed by Fritzke (1995) represents an incremental extension of SOM and the NG. The GNG has the ability to add new neurons to an initially small network by evaluating local statistical measures gathered during previous adaptations and to create connections between existing neurons. The network topology is generated incrementally using competitive Hebbian learning (Martinetz, 1993), i.e. for each input, an edge connection is generated between the neuron that best matches the input and the second-best matching one.

The GNG network starts with a set of $N = 2$ neurons in the input space. At each training iteration, the algorithm is given a random input signal $\mathbf{x}$ drawn from the input distribution $p(\mathbf{x})$. The closest neuron $b$ and the second closest unit $s$ of $\mathbf{x}$ in $N$ are found and if the connection $(b, s)$ does not exist, it is created. The local error of $b$ is updated by $\Delta E_b = \|\mathbf{x} - \mathbf{w}_b\|^2$ and $w_s$ is moved towards $\mathbf{x}$ by a fraction $\epsilon_b$. The weight of all the topological neighbors of $b$ are also moved towards $\mathbf{x}$ by a fraction $\epsilon_i$.

If the number of given inputs is a multiple of a parameter $\lambda$, a new neuron is created halfway between those two neurons that have maximum accumulated error. A connection-age-based mechanism leads to the removal of rarely used connections and neurons without connections. The algorithm stops when a criterion is met, i.e. some performance measure, network size, or a given number of training epochs. The complete GNG training algorithm is provided in Appendix B.

### Growing When Required (GWR)

Different to the GNG which creates new neurons at a fixed growth rate, the GWR network proposed by Marsland et al. (2002) creates new nodes whenever the activity of trained neurons is smaller than a given threshold. As a criterion for neural growth, the training algorithm considers the amount of network activation at time $t$ computed as a function of the distance between the current input $\mathbf{x}(t)$ and its best-matching neuron $\mathbf{w}_b$:

$$a(t) = \exp(-\|\mathbf{x}(t) - \mathbf{w}_b\|). \tag{3.12}$$

Additionally, the algorithm considers the number of times that neurons have fired so that recently created neurons are properly trained before creating new ones. For this purpose, the network implements a firing counter $\eta \in [0, 1]$ used to express how frequently a neuron has fired based on a simplified model of how the efficacy of an habituating synapse reduces over time (Stanley, 1976). The firing counter is given by

$$\eta(t) = \eta_0 - \frac{S(t)}{\alpha} \cdot (1 - \exp(-\alpha_t/\tau)), \tag{3.13}$$

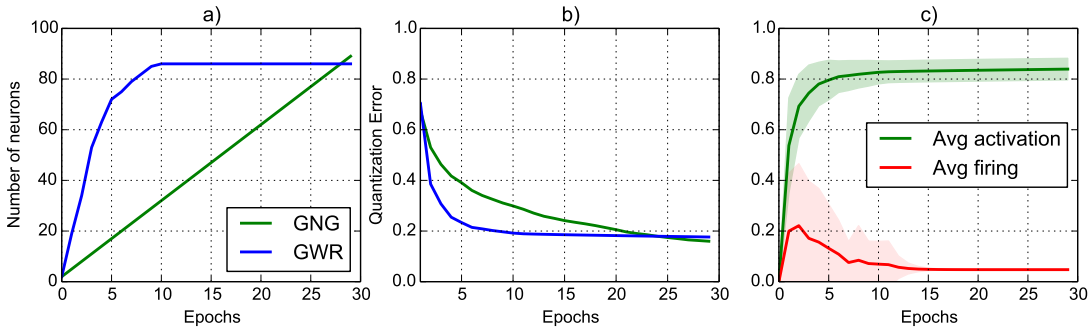



Figure 3.2: Comparison of GNG and GWR growth behavior: a) number of neurons, b) quantization error, and c) GWR average activation and firing counter through 30 training epochs for the Iris dataset (150 four-dimensional samples).

where $\eta_0$ is the resting value, $S(t)$ is the stimulus strength, and $\tau$ and $\alpha$ are constants that control the behavior of the curve.

The use of an activation threshold and firing counters to modulate the growth of the network leads to create a larger number of neurons at early stages of the training and then tune the weights of existing neurons through subsequent training epochs. This behavior is particularly convenient for incremental learning scenarios since neurons will be created to promptly distribute in the input space, thereby yielding fast convergence through iterative fine-tuning of the topological map. The GWR algorithm will then iterate over the training set until a given stop criterion is met, e.g. a maximum network size or a maximum number of iterations. The learning procedure for GWR is illustrated in Appendix B.

A comparison between GNG and GWR learning in terms of the number of neurons, quantization error (average discrepancy between the input and representative neurons in the network), and parameters modulating network growth (average network activation and firing rate) is shown in Fig. 3.2 over 30 training epochs for the Iris dataset[1]. Such a learning behavior is particularly convenient for incremental learning scenarios since neurons will be created to promptly distribute in the input space, thereby allowing a faster convergence through iterative fine-tuning of the topological map. It has been shown that GWR-based learning is particularly suitable for novelty detection and cumulative learning in robot scenarios (Marsland et al., 2002, 2005).

The standard GNG and GWR learning algorithms do not account for temporal sequence processing. Therefore, there is a motivation to extend these networks with recurrent connectivity while preserving desirable learning properties such as computational efficiency and network convergence.

[1] `http://archive.ics.uci.edu/ml/datasets/Iris`

## 3.3 Recurrent Self-Organizing Networks

The efficient processing of time-varying stimuli plays a crucial role in biological systems. Therefore, there is a strong motivation to design artificial systems that account for the processing of sequences, from low-level signals to complex high-level cognitive functions. Popular learning tasks with sequential data have been action recognition, DNA analysis, and natural language processing.

The self-organizing networks described in Section 3.2 have been designed for the spatial domain and do not naturally account for the extrapolation of temporal relations from time-varying input samples. As an attempt to learn sequentially ordered patterns, data windows in terms of serialized concatenations of a fixed number of samples from the input were used (e.g., Martinetz, 1993). However, this approach may lead to loss of information, the curse of dimensionality, and inappropriate metrics (Strickert and Hammer, 2005). Therefore, temporal extensions of these networks have been proposed that implement recurrent connectivity so that neural activation in the map is driven by multiple time steps.

The first example was the Temporal Kohonen Map (TKM, Chappell and Taylor, 1993), equipped with recurrent neurons in terms of leaky integrators. The computation of the distance of a neuron $\mathbf{w}_i$ from the input sequence $(\mathbf{x}_1, ..., \mathbf{x}_t)$ at time $t$ with similarity measure $d_W$ is

$$\tilde{d}_i(t) = \alpha \cdot d_W(\mathbf{x}(t), \mathbf{w}_i) + (1 - \alpha) \cdot \tilde{d}_i(t-1), \tag{3.14}$$

where $\alpha \in (0; 1)$ controls the rate of signal decay, expressing the quality of the representation of the current input and the exponentially weighted past. However, in the TKM there is no explicit back-reference to previous map activity, i.e. the context is only implicitly represented by the weights. Therefore the sequence representation domains are restricted to the superposition of values from the domain of the processed sequence entries.

To provide a less restricted recurrence, in the RecSOM (Voegtlin, 2002) the distance of a neuron from the input sequence at time $t$ is computed as

$$d_i(t) = \alpha \cdot d_W(\mathbf{x}(t), \mathbf{w}_i) + \beta \cdot \|\mathbf{c}_i - R_{t-1}\|, \tag{3.15}$$

$$R_{t-1} = (\exp(-\tilde{d}_1(t-1)), ..., \exp(-\tilde{d}_N(t-1))), \tag{3.16}$$

where $\mathbf{c}_i$ are the context descriptors of each neuron, $R_{t-1}$ is the context vector of the previous time step, $N$ is the number of neurons in the map, and $\|\cdot\|$ denotes the Euclidean distance. This preserves the information available within the activation at the last timestep. However, this is computationally expensive due to the high-dimensional contexts attached to each neuron.

A more compact model was introduced by a SOM for structured data (Hagenbuchner et al., 2003), where an additional context vector is used for each neuron, but only the last winner index is stored as information of the previous map state such that

$$\tilde{d}_i(t) = \alpha \cdot d_W(\mathbf{x}(t), \mathbf{w}_i) + \beta \cdot d_G(\mathbf{I}_{t-1} - \mathbf{c}_i), \tag{3.17}$$

where $\mathbf{I}_{t-1}$ denotes the index of the winner neuron at $t-1$ and $d_G$ is the grid distance measure. However, this recurrent activation cannot be used for arbitrary lattice shapes since it relies on fixed grid distances to update the winning neuron and its neighbours.

Different approaches with context learning have been proposed that use a compact reference representation for an arbitrary lattice topology. Context learning as proposed by Strickert and Hammer (2005) combines a compact back-reference with a weighted contribution of the current input and the past. Each neuron is equipped with a weight vector $\mathbf{w}_i$ and a temporal context $\mathbf{c}_i$ (with $\mathbf{w}_i, \mathbf{c}_i \in \mathbb{R}^d$), the latter representing the activation of the entire map at the previous timestep. The recursive activation function of a sequence is given by the linear combination

$$\tilde{d}_i(t) = \alpha \cdot d_W(\mathbf{x}(t), \mathbf{w}_i) + (1-\alpha) \cdot d_W(\mathbf{C}(t), \mathbf{c}_i), \tag{3.18}$$

$$\mathbf{C}_i = \beta \cdot \mathbf{w}_{I(t-1)} + (1-\beta) \cdot \mathbf{c}_{I(t-1)}, \tag{3.19}$$

where $\alpha, \beta \in (0,1)$ are fixed parameters, $\mathbf{C}(t)$ is a global context vector, and $I(t-1)$ denotes the index of the winner neuron at time $t-1$.

The training is carried out by adapting the weight and the context vector towards the current input and context descriptor according to:

$$\triangle\, \mathbf{w}_i = \epsilon_i \cdot h_\sigma(d_N(i, I_t)) \cdot (\mathbf{x}(t) - \mathbf{w}_i), \tag{3.20}$$

$$\triangle\, \mathbf{c}_i = \epsilon_i \cdot h_\sigma(d_N(i, I_t)) \cdot (\mathbf{C}(t) - \mathbf{c}_i), \tag{3.21}$$

where $\epsilon_i$ is the learning rate, $h_\sigma$ is usually a Gaussian function, and $d_N : N \times N \to \mathbb{R}$ is a neighborhood function that defines the topology of the network. After the training, Strickert and Hammer (2005) showed that $\mathbf{C}(t)$ converges to the optimal global temporal context vector $\mathbf{C}^{opt}(t)$ such that

$$\mathbf{C}^{opt}(t) = \sum_{j=1}^{t-1} (1-\beta) \cdot \beta^{t-1-j} \cdot \mathbf{x}(j). \tag{3.22}$$

Context learning can be applied to lattices with arbitrary topology as well as to incremental approaches that vary the number of neurons over time. For instance, a GNG model equipped with context learning (MergeGNG, Andreakis et al., 2009) uses the activation function defined by Eq. 3.18 and 3.19 to compute winner neurons and creates new neurons with a temporal context. Furthermore, this formulation of context learning can be extended to equip each neuron with an arbitrary number of context descriptors, leading to a reduced temporal quantization error (Estévez and Hernández, 2011). This is due to an increase in memory depth and temporal resolution following the idea of a Gamma memory model (de Vries and Príncipe, 1992). The computation of winner neurons in a network with a K-order Gamma memory is as follows:

$$d_i(t) = \alpha_w \cdot \|\mathbf{x}(t), \mathbf{w}_i\|^2 + \sum_{k=1}^{K} \alpha_k \cdot \|\mathbf{C}_k(t) - \mathbf{c}_k^i\|^2, \tag{3.23}$$

$$\mathbf{C}_k(t) = \beta \cdot \mathbf{c}_k^{I_{t-1}} + (1-\beta) \cdot \mathbf{c}_{k-1}^{I_{t-1}} \quad \forall K = 1, ..., K, \tag{3.24}$$

where $\alpha, \beta \in (0; 1)$ are constant values that modulate the influence of the current input and the past, and $\mathbf{c}_0^{I_{t-1}} \equiv \mathbf{w}^{I_{t-1}}$ with random $\mathbf{c}_k^{I_0}$ at $t = 0$. This results in a mean memory depth $D = K/(1-\beta)$ with temporal resolution $R = 1-\beta$. Therefore, both depth and resolution are modulated by the value of $\beta$. When $K = 1$, this approach reduces to the standard context learning mechanism (Eq. 3.18). Since the definition of the context descriptors is recursive, setting $\alpha_w > \alpha_1 > \alpha_2 > ... > \alpha_{K-1} > \alpha_K > 0$ should reduce the propagation of errors from early filter stages to higher-order contexts.

Experiments with Gamma-SOM and Gamma-GNG networks were reported by Estévez and Hernández (2011); Estévez and Vergara (2012), showing reduced temporal quantization error with respect to traditional context learning models for trained networks using multiple context descriptors.

## 3.4 Summary

We provided an overview on a set of neurobiologically-motivated neural networks implementing mechanisms of self-organization for the formation of topological maps driven by the distribution of the input. In addition to improving the understanding of experience-driven cortical organization via the development of simplified computational models, self-organizing networks have been successfully applied to a large number of data-analysis problems. From early formulations of topology-preserving neural networks to more elaborated models of information processing, self-organizing systems have been shown to represent a powerful and computationally convenient mechanism to deal with high-dimensional input and able to extrapolate underlying data structure in the spatiotemporal domain.

Although so far computational models of self-organization have mainly focused on small-scale systems, state-of-the-art neural architectures can be used as a stepping-stone towards the modeling of more complex systems and high-level cognitive functions, especially in the light of a more extensive understanding of cortical maps in the mammalian brain. In the next chapters, we propose a set of learning architectures for the recognition of complex action sequences from multiple visual and auditory cues. We use hierarchically-arranged self-organizing networks for the robust development of action representations with increasing level of abstraction, resembling neural mechanisms of hierarchical processing in the cortex discussed in Section 2.1. Furthermore, we show how Hebbian-like learning can be used for the development of associative connectivity between two self-organizing systems in an unsupervised fashion, and how this mechanism contributes to the emergence of multimodal action representations from audiovisual input.

# Chapter 4

# Self-Organizing Neural Integration of Visual Action Cues

## 4.1 Introduction

The visual recognition of articulated human movements is fundamental to a wide range of artificial systems oriented towards human-robot communication, action classification and action-driven perception. These challenging tasks may generally involve the processing of a large amount of visual information and learning mechanisms for generalizing a set of training actions and classifying novel samples. To operate in natural environments, a crucial property of artificial vision systems is the efficient and robust recognition of actions, also under noisy conditions caused by, for instance, systematic sensor errors or temporarily occluded persons. As discussed in Section 2.1, neurophysiological studies support a highly flexible and adaptive biological system with separate neural pathways for the distinct processing of pose and motion features at multiple levels and the subsequent integration of these visual cues for action perception. Computational implementations of simplified biological self-organizing models have shown motivating results on the recognition of dynamic patterns (see Chapter 3). With the use of extended models of conventional input-driven self-organization, it is possible to obtain progressively generalized representations of sensory inputs and learn inherent spatiotemporal dependencies of input sequences.

In this chapter, we propose a set of neurobiologically-motivated neural network architectures to achieve noise-tolerant action recognition from videos in real time. Our approach consist of a series of hierarchically-arranged self-organizing networks for processing action features in the spatiotemporal domain. The architectures are based on three assumptions consistent with neurophysiological evidence from the mammalian visual system:

1. Complex motion is analysed in parallel by two separated pathways and subsequently integrated to provide a joint percept (Vangeneugden et al., 2009);

2. Both pathways contain hierarchies to extrapolate shape and optic-flow features with increasing complexity, from low- to high-level representations of

the visual stimuli (Giese and Poggio, 2003; Hasson et al., 2008);

3. Input-driven self-organization is crucial for the cortex to tune the neurons according to the distribution of the inputs (Willshaw and von der Malsburg, 1976; Nelson, 2000).

Under these assumptions, we carry out action learning and classification through a two-pathway hierarchy of growing self-organizing networks that cluster pose and motion cues separately. We first extract pose and motion features from sequences of depth map videos and then cluster action features in terms of prototype pose-motion trajectories. Multi-cue trajectories from matching action frames are subsequently combined to provide action dynamics in the joint feature space. For our action recognition task, we propose two types of growing self-organizing networks extended for classification: the associative GNG (Parisi et al., 2014c) and the associative GWR (Parisi et al., 2015b). In both cases, visual representations obtained in an unsupervised fashion are associated with symbolic labels. This technique allows to predict action labels for novel visual samples. Furthermore, in contrast to static self-organizing models, growing networks dynamically change their topological structure to better match the input space and allow to train the system with novel actions without the need to re-train from scratch. To compare the performance of these two architectures (GNG vs GWR), we collected a dataset of full-body actions with 10 actions that we introduce in Section 4.2. Furthermore, we provide an evaluation of the performance of our architectures with respect to the state of the art in action recognition from depth map videos with experiments on a public benchmark of domestic daily actions. Finally, we show how this hierarchical approach can be extended to account for the rudimentary recognition of transitive actions, i.e. body sequences that also involve the use of objects such as grasping a bottle and drinking. Analogies to biological findings and limitations of these three learning architectures are discussed in Chapter 8.

## 4.2 KT Full-Body Action Dataset

The Knowledge Technology (KT) action dataset is composed of 10 full-body actions performed by 13 subjects with a normal physical condition (Parisi et al., 2014c). The dataset contains the following actions: *standing, walking, jogging, picking up, sitting, jumping, falling down, lying down, crawling,* and *standing up.*

Videos were captured in a home-like environment with a Kinect sensor installed $1,30$ meters above the ground. Depth maps were sampled with a VGA resolution of $640 \times 480$ and an operation range from 0.8 to 3.5 meters at 30 frames per second. To avoid biased execution, subjects did not receive explicit instructions on how to perform the actions nor regarding the purpose of the study. Snapshots of actions are shown in Fig. 4.1 as raw depth images, segmented body silhouettes, skeletons, and body centroids. (For more video sequences of actions, see Appendix C.)

From the raw depth map sequences, 3D body joints were estimated on the

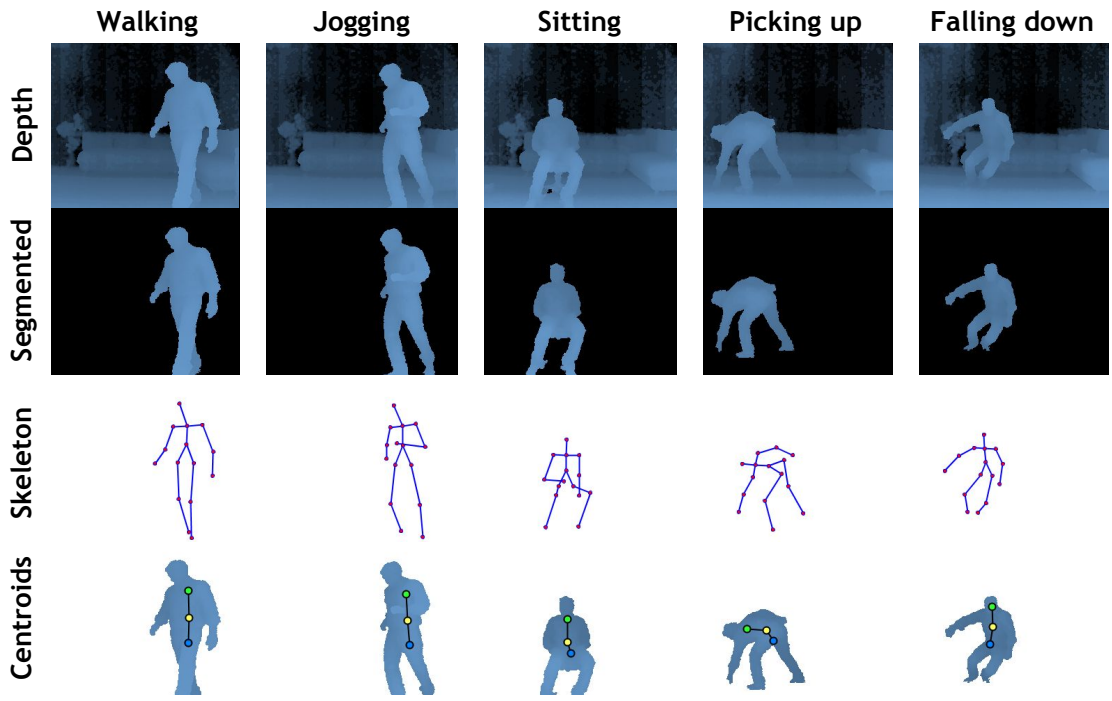



Figure 4.1: Snapshots of actions from the KT action dataset visualized as raw depth images, segmented body, skeleton, and body centroids.

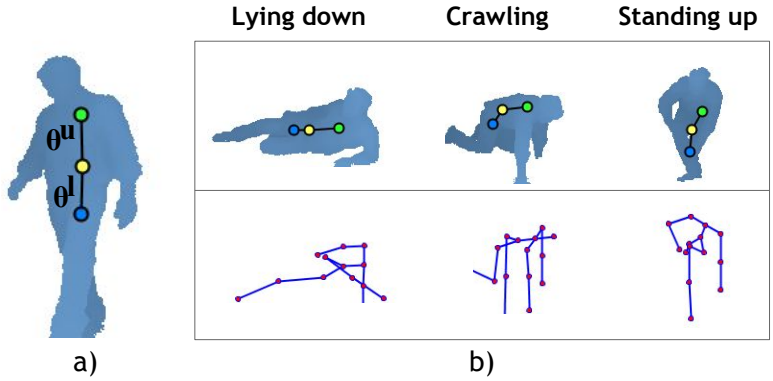



Figure 4.2: Action representations: (a) Three centroids with body slopes $\theta^u$ and $\theta^l$, and b) comparison of body centroids (top) and noisy skeletons (bottom).

basis of the tracking skeleton model provided by OpenNI.[1] In previous research, we proposed a simple model to track a spatially extended skeletons with two centroids and a global body orientation (Parisi and Wermter, 2013). The centroids were estimated as the centers of mass that follow the distribution of the main body masses on each posture. For this thesis, we extended our previous model to describe articulated actions more accurately by considering three body centroids (Fig. 4.2): $C_1$ for upper body with respect to the shoulders and the torso; $C_2$ for middle body

[1] OpenNI SDK. `http://www.openni.org/openni-sdk/`

with respect to the torso and the hips; and $C_3$ for lower body with respect to the hips and the knees, with each centroid computed as a point sequence of real-world coordinates $C = (x, y, z)$. As shown in Fig. 4.2, three body centroids are enough to represent significant posture characteristics while maintaining a low-dimensional feature space. Furthermore, this low-dimensional representation increases tracking robustness for situations of partial occlusion with respect to a skeleton model comprising a larger number of body joints.

To attenuate sensor noise, we used the median value of the last 3 estimated points (yielding action features at 10 frames per second). We then estimated upper and lower orientations $\theta_u$ and $\theta_l$ given by the slope angles of the line segments $\{C_1, C_2\}$ and $\{C_2, C_3\}$ respectively. As shown in Fig. 4.2.a, $\theta_u$ and $\theta_l$ describe the overall body pose according to the orientation of the torso and the legs, which allows to capture significant body features such as the characteristic posture of actions. This technique has been shown to provide a more reliable estimation of overall body posture than using skeletons, specially in cases of self-occlusion and unusual body postures (Fig. 4.2.b).

We computed the body velocity $S_i$ as the difference in pixels of the upper centroid $C_1$ between two consecutive frames. This centroid was chosen based on the motivation that the orientation of the torso is the most characteristic reference during the execution of full-body actions (Papadopoulos et al., 2014). We then encode $S_i$ as horizontal and vertical speed with respect to the image plane, respectively expressed as $h_i = \sqrt{(S_i^x)^2 + (S_i^z)^2}$ and $v_i = S_i^y$. The former refers to the target moving along the width and depth axis, i.e. closer, further, right, and left. The latter represents the speed with respect to height, e.g. negative if the target is moving down.

For each action frame $i$, we computed the following pose-motion vector:

$$\mathbf{f}_i = (\theta_i^u, \theta_i^l, h_i, v_i). \tag{4.1}$$

Thus, each action $A_j$ will be composed of a set of sequentially ordered pose-motion vectors such that:

$$A_j = \{(\mathbf{f}_i, l_j) : l_j \in L\}, \tag{4.2}$$

where $l_j$ is the action class label and $L$ is the set of class labels. Action labels were manually annotated for video sequences.

## 4.3 Two-Stream Hierarchical Processing

Neurophysiological studies have shown that the mammalian visual system processes biological motion in two separate neural pathways (Giese and Poggio, 2003; Vangeneugden et al., 2009). The ventral pathway recognizes sequences of snapshots of body postures, while the dorsal pathway recognizes movements in terms of optic-flow patterns. Both pathways comprise hierarchies that extrapolate visual features with increasing complexity of representation. Neurons in the macaque and human superior temporal sulcus (STS) are sensitive to both motion and posture for representing similarities among actions, thus suggesting contributions from

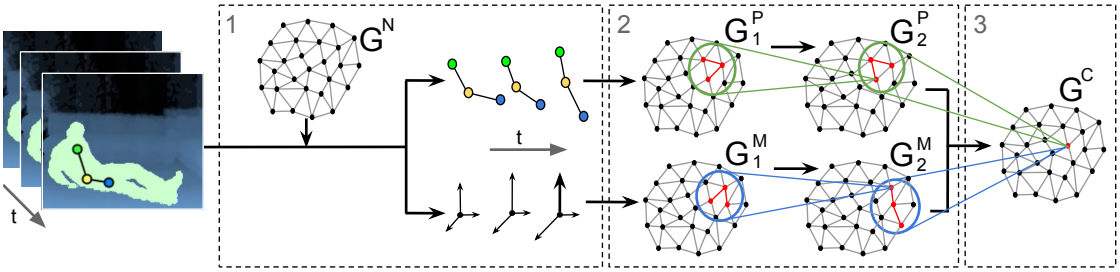



Figure 4.3: Three-stage GNG hierarchical pose-motion processing: 1) noise detection and removal; 2) hierarchical processing of pose-motion trajectories in two parallel streams; 3) classification of multi-cue trajectories (Parisi et al., 2014c).

converging cues received from the ventral and dorsal pathways (Oram and Perrett, 1996; Beauchamp et al., 2003). As discussed in Section 2.1.2, there is a consensus that posture and motion together play a key role in biological motion perception (Garcia and Grossman, 2008; Thirkettle et al., 2009). These findings have served to the development of architectures using learned prototype patterns to recognize actions, consistent with the idea that STS neurons integrate both body pose and motion.

In this section, we propose a learning framework for recognizing human full-body actor-independent actions. We first extract pose and motion features from depth map video sequences and then cluster actions in terms of prototypical pose-motion trajectories. Multi-cue samples from matching frames are processed separately by a two-stream hierarchy of GNG networks (Fritzke, 1995). The GNG is an unsupervised, incremental clustering algorithm able to dynamically change its topological structure to represent the input space. Clustered trajectories from the parallel streams are combined to provide joint action dynamics. We process the samples under the assumption that action recognition is selective for temporal order (Giese and Poggio, 2003). Therefore, positive recognition of an action occurs only when trajectory samples are activated in the correct temporal order. In order to assign labels to clustered trajectories, we extend the GNG with two labelling functions. Noisy samples are automatically detected and removed from the training and the testing set to increase recognition accuracy. We present and discuss experimental results on the KT action dataset with 10 full-body actions.

### 4.3.1 Learning Architecture

Our GNG-based architecture consists of three main stages: 1) detection and removal of noisy samples from the dataset; 2) hierarchical processing of samples from matching frames by two separate processing streams in terms of activation trajectories; and 3) classification of action segments as multi-cue trajectories. An overall overview of the framework is depicted in Fig.4.3.

### Hierarchical Learning

The motivation underlying hierarchical learning is to use trajectories of neural activation from one network as input for the training for a subsequent network. This mechanism allows to obtain progressively specialized neurons coding inherent spatiotemporal dependencies of the input, consistent with the assumption that the recognition must be selective for temporal order (Bertenthal and Pinto, 1993; Giese and Poggio, 2003). Therefore, positive recognition of action segments occurs only when neurons along the hierarchy are activated in the correct order of learned movement sequences.

The second stage is composed of a two-stream hierarchy for processing pose-motion cues separately. Hierarchical training is carried out as follows. We first train a network $G$ with a training set $T$. After the training is completed, the subsequent network $G^*$ will be trained with a new training set that is obtained by computing trajectories of best-matching neurons from $G$ for samples of $T$. Given a training sequence $X$ and the trained weight vectors $\mathbf{w}_j$ of $G$, the set of trajectories $\Omega(X)$ is given by

$$\Omega(X) = \{\mathbf{x}_i \in X : \mathbf{w}_{b(\mathbf{x}_i)}, \mathbf{w}_{b(\mathbf{x}_{i-1})}, ..., \mathbf{w}_{b(\mathbf{x}_{i-q+1})}\}, i \in [q..m], \tag{4.3}$$

where $b(\mathbf{x}_i) = \arg\min_{i \in W} \|\mathbf{x}_i - \mathbf{w}_j\|$, $q$ is the length of the trajectory, and $m$ is the number of samples of $X$. After the training of the higher level network is completed, each neuron in $G^*$ will encode a sequence-selective action segment from $q$ consecutive frames. This mechanism allows to obtain specialized neurons coding the spatiotemporal structure of the input.

In our architecture (Fig.4.3), we first train the networks $G_1^P$ and $G_1^M$ with the denoised training sets $P$ and $M$ respectively. After this training phase, chains of prototype neurons of training samples produce time-varying trajectories on each network. After this step, we train $G_2^P$ and $G_2^M$ with the training sets of concatenated trajectories of best-matching neurons as defined by Eq. 4.3.

The network layer $\mathrm{G}^C$ integrates pose-motion features by training the network with a new set $\Psi$ containing the concatenation of the activation trajectories of from $G_2^P$ and $G_2^M$ such that:

$$\Psi = \{\Omega(\mathcal{P}) \frown \Omega(\mathcal{M})\}, \tag{4.4}$$

where $\mathcal{P}$ and $\mathcal{M}$ are the set of neural activations from the pose and motion stream respectively. After the training of $G^C$ is completed, each neuron will encode a sequence-selective prototype action segment, thereby integrating changes in the configuration of a person's body pose over time.

### Noise Detection

Pose-motion vectors $\mathbf{f}_i$ as described in Section 4.2 are susceptible to tracking errors due to occlusion or systematic sensor errors, which may introduce noise in terms of values highly detached from the dominating point clouds. We assume inconsistent changes in body velocity to be caused by tracking errors rather than actual motion.

Therefore, we remove noisy motion samples to create smoother inter-frame transitions. First, the network $G^N$ is trained using only the motion samples. Second, the training motion samples are processed again to obtain the set of quantization errors $E$ from the trained network, which contains the distances from the best-matching neurons of all motion sample. We then compute the empirically defined threshold that considers the distribution of the samples as $th = 2\sigma(E)\sqrt{\mu(E)}$, where $\sigma(E)$ is the standard deviation of $E$ and $\mu(E)$ is its mean.

For each motion sample, if its distance from the best-matching neuron is greater than $th$, then the sample is considered to be noisy and its associated vector $\mathbf{f}_i$ is removed from the training set. We then obtain a new denoised training set from which we create two distinct sets with sequentially ordered pose and motion features, formally defined as $P = \{(\theta_i^u, \theta_i^l)\}$ and $M = \{(h_i, v_i)\}$ respectively.

### 4.3.2 Action Classification

For assigning labels to clustered trajectories with $G^C$, we extend the GNG algorithm with two labelling functions: one for the training phase and one for predicting the label of unseen samples at recognition time. First, we define a function $l : N \rightarrow L$ where $N$ is the set of neurons and $L$ is the set of class labels. According to the minimal-distance strategy (Beyer and Cimiano, 2011), the sample $\psi_k \in \Psi$ adopts the label $l_j$ of the best-matching neuron:

$$l(\psi_k) = l_j = l(\arg\min_{\psi_i \in \Psi} \|\psi_k - \psi_i\|^2). \tag{4.5}$$

At recognition time, our goal is to predict class labels from unseen samples in terms of pose-motion trajectory prototypes. Therefore, we define a prediction function $\varphi : \Psi \rightarrow L$ inspired by a single-linkage strategy in which a new sample $\psi_{new}$ is labeled with $l_j$ associated to the neuron $i$ that minimizes the distance to this new sample:

$$\varphi(\psi_{new}) = \arg\min_{l_j}(\arg\min_{\psi_i \in \Psi} \|\psi_{new} - \psi_i\|^2). \tag{4.6}$$

The adopted labelling techniques have shown to achieve best classification accuracy among other offline labelling strategies (Beyer and Cimiano, 2011).

The hierarchical flow is composed of 3 networks with each subsequent network neuron encoding a window of 3 samples from the previous one. Therefore, this classification algorithm returns the first action label $l_{new}$ after 7 new samples $\hat{f} \in F$ as defined by Eq. 4.1. Then, applying the temporal sliding window scheme, we get a new action label for each new sample. For instance, operating at 10 frames per second, we would get the first action label after $7/10 = 0.7$ seconds.

### 4.3.3 Results and Evaluation

We evaluated our approach on the KT action dataset described in Section 4.2. The KT action dataset is composed of 10 full-body actions performed by 13 subjects.

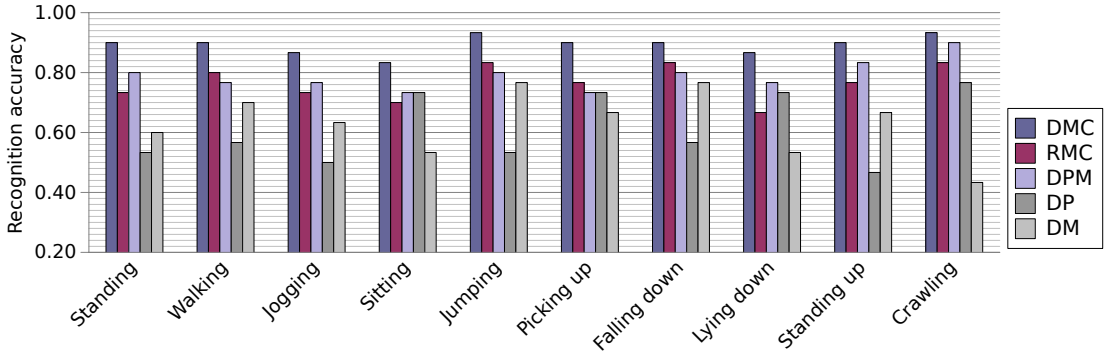



Figure 4.4: Recognition accuracy under 5 different processing conditions: Denoised multi-cue (*DMC*), denoised pose-motion vector (*DPM*), raw multi-cue (*RMC*), denoised pose-only (*DP*), and denoised motion-only (*DM*) (Parisi et al., 2014c).

For our experiments, we divided the data equally into training and test set, i.e. 30 sequences of 10 seconds for cyclic actions such as *standing, walking, jogging, sitting, lying down, and crawling*, and 30 repetitions for each goal-oriented action such as *picking up, jumping, falling down, and standing up*. Both the training and test sets contained data from all participants.

We used the following GNG training parameters: learning step sizes $\epsilon_b = 0.05$, $\epsilon_n = 0.005$, node insertion interval $\lambda = 350$, error reduction constant $\alpha = 0.5$, and error reduction factor $d = 0.995$ (see Appendix B for the detailed training algorithm). Maximum network size and the number of training epochs varied for each of the six GNG networks and were experimentally adjusted based on the network performance for different input distributions.

We evaluated the recognition accuracy of the framework under 5 different processing conditions: denoised multi-cue (DMC) and raw multi-cue (RMC) samples, denoised "pose only" (DP) and denoised "motion only" (DM) samples, and joint pose-motion vectors (DPM) as defined in Eq. 1 processed by a single stream. As can be seen in Fig. 4.4, the use of denoised multi-cue trajectory prototypes yields the best average recognition result (89%). The removal of noise from the data sets increases average recognition accuracy by 13%. The DMC approach exhibits average improvements over DP and DM of 28% and 26% respectively.

Our results also show that DMC exhibits increased accuracy over the learning of joint pose-motion vectors (DPM) by 10%. This is partly due to the fact that the DPM approach forces the early convergence of the networks in the joint pose-motion space, while DMC and RMC learn a sparse representation of disjoint pose-motion prototypes that are subsequently combined to provide joint action dynamics. The reported results for actor-independent action recognition were obtained with low latency providing real-time characteristics.

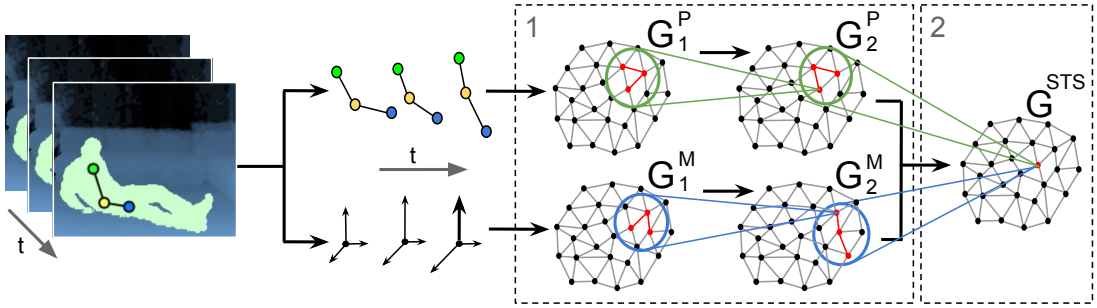



Figure 4.5: GWR-based architecture for pose-motion processing: 1) hierarchical processing of pose-motion features in parallel; 2) integration of neuron trajectories in the joint pose-motion feature space (Parisi et al., 2015b).

## 4.4 Hierarchical GWR Model

In the previous section, we explored the use of hierarchical self-organization for integrating pose-motion cues using GNG learning. The unsupervised learning algorithm was extended with two labelling functions for classification purposes. In this section, we use GWR networks that can create new neurons whenever the activity of the best-matching neuron of the input is not sufficiently high, leading to a more efficient convergence with respect to GNG networks that use a fixed insertion interval. In the GNG model, an extra network was used to automatically detect outliers in the training and test set. However, the removal of noisy cues via an additional specialized network lacks neurobiological support and adds complexity to the model. With the use of an extended GWR learning mechanism, we show that this process can be embedded naturally into the self-organizing hierarchy for the clustering of action cues and allows to remove noisy samples also during live classification.

We present our hierarchical GWR architecture for the classification of action samples and report a series of experiments on the KT action dataset and a benchmark of domestic actions CAD-60 (Sung et al., 2012).

### 4.4.1 GWR-based Learning Architecture

Our architecture consists of a two-stream hierarchy of GWR networks that processes extracted pose and motion features in parallel and subsequently integrates clustered neuronal activation trajectories from both streams. This latter network resembles the response of STS model neurons encoding sequence-selective prototypes of action segments in the joint pose-motion domain. An overall overview of the architecture is depicted in Fig. 4.5.

Different to the growing process of the GNG (Fritzke, 1995), GWR-based learning creates new nodes whenever the activity of trained neurons is smaller than a given threshold. Additionally, the training algorithm considers the number of times that a neuron has fired so that recently created neurons are properly trained before

creating new ones. For this purpose, the network implements a firing counter to express how frequently a neuron has fired based on a simplified model of how the efficacy of an habituating synapse reduces over time. As discussed in Section 3.2.2, the use of an activation threshold and firing counters to modulate the growth of the network leads to the creation of a larger number of neurons at early stages of the training and then a tuning of the weights of existing neurons through subsequent training iterations (epochs). This behavior is particularly convenient for incremental learning scenarios since neurons will be created to promptly distribute in the input space, thereby yielding fast convergence through iterative fine-tuning of the topological map. The GWR algorithm will then iterate over the training set until a given stop criterion is met, e.g. a maximum network size or a maximum number of iterations.

In our architecture, hierarchical learning is carried out as described in Section 4.3.1. At the first stage of our hierarchy, each stream is composed of two GWR networks to process pose and motion features separately. We therefore compute two distinct datasets with sequentially-ordered pose and motion features, denoted as $P$ and $M$ respectively. Since $P$ and $M$ are processed by different network hierarchies, they can differ in dimensionality. Following the notation introduced in Fig. 1, we train the networks $G_1^P$ and $G_1^M$ with samples from $P$ and $M$ respectively. After this step, we train $G_2^P$ and $G_2^M$ with the training sets of concatenated trajectories of best-matching neurons as defined by Eq. 4.3. The STS stage consists of the integration of prototype activation trajectories from both streams by training the network $G^{STS}$ with two-cue trajectory samples. The network layer $\mathrm{G}^{STS}$ integrates pose-motion features by training the network with the concatenation of vectors $\Psi = \{\Omega(\mathcal{P}) \frown \Omega(\mathcal{M})\}$, where $\mathcal{P}$ and $\mathcal{M}$ are the activation trajectories from $G_2^P$ and $G_2^M$ respectively. After the training of $G^{STS}$ is completed, each neuron will encode a sequence-selective prototype action segment, thereby integrating changes in the configuration of a person's body pose over time.

For the purpose of action classification, we extend the unsupervised GWR-based learning with two labelling functions: one for the training phase and one for returning the label of unseen samples as described in Section 4.3.2. We train the $G^{STS}$ network with the labeled training pairs so that symbolic labels are attached to neurons representing temporally-ordered visual representations.

### Noise Detection

The presence of noise in the sense of outliers in the training set has been shown to have a negative influence on the formation of faithful topological representations using SOMs (Parisi and Wermter, 2013), whereas such an issue is partially addressed by incremental networks. For instance, incremental networks such as GNG and GWR are equipped with a mechanism to remove rarely activated nodes and connections that may represent noisy input. In contrast to GNG, however, the learning strategy of the GWR shows a quick response to changes in the distribution of the input by creating new neurons to match it. The insertion threshold $a_T$ modulates the number of neurons that will be added, e.g. for high values of $a_T$

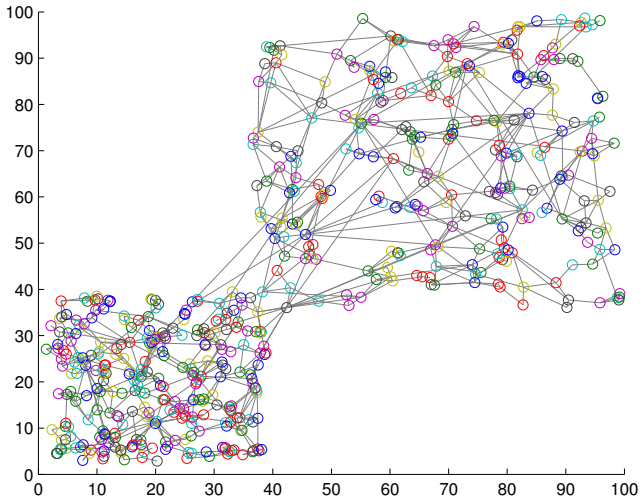



Figure 4.6: A GWR network trained with a normally distributed training set of 1000 samples resulting in 556 nodes and 1145 connections (Parisi et al., 2015b).

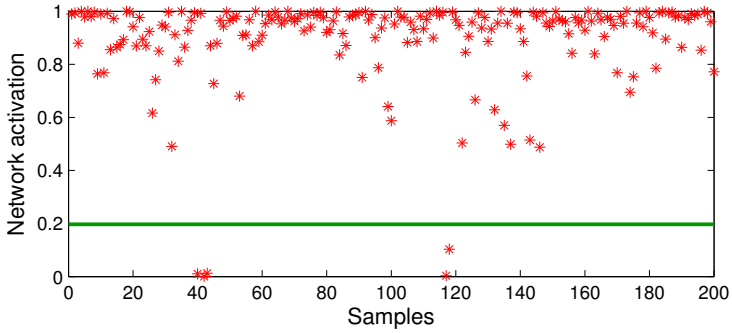



Figure 4.7: Activation values for the network trained in Fig. 4.6 with a test set of 200 samples containing noise. Noisy samples line under novelty threshold $a_{new} = 0.1969$ (green line) (Parisi et al., 2015b).

more nodes will be created. The network is also equipped with a mechanism to avoid slight input fluctuations to perturb the learning convergence and the creation of unnecessary nodes. The GWR takes into account the number of times that a neurons has been activated, so that neurons that have been activated more times, are trained less. Therefore, an additional threshold modulates the firing counter of neurons so that during the learning process less trained neurons are updated, whereas new neurons are created only when existing neurons do not sufficiently represent the input. A number of experiments have shown that the GWR is well-suited for novelty detection (Marsland et al., 2005), which involve the identification of inputs that do not fit the learned model.

In line with this mechanism, we use the activation function to detect noisy input after the training phase. The activation function will be equal to 1 in response

to input that perfectly matches the model, i.e. minimum distance between the weights of the neuron and the input, and will decrease exponentially for input with a higher distance. If the response of the network to the novel input is below a given novel activation threshold $a_{new}$, then the novel input can be considered noisy in the sense that it is not represented by well-trained prototype neurons, and thus discarded. The threshold value $a_{new}$ can be empirically selected by taking into account the response distribution of the trained network with respect to the training set. For each novel input $\mathbf{x}_{new}$, we compute:

$$\exp(-\|\mathbf{x}_{new} - \mathbf{w}_b\|) < \bar{A} - u \cdot \sigma(A), \tag{4.7}$$

where $\mathbf{w}_b$ is the best-matching neuron of $\mathbf{x}_{new}$, $\bar{A}$ and $\sigma(A)$ are respectively the mean and the standard deviation of the set of activations $A$ from the training set, and $u$ is a constant value that modulates the influence of fluctuations in the activation distribution.

Fig. 4.6 shows a GWR network trained with 100 input vectors with two normally distributed clusters. Over its 500 iterations, the network created 556 neurons and 1145 connections ($a_T = 0.95, u = 4$). The activation values for a test set of 200 samples (also normally distributed) containing artificially introduced noise are shown in Fig. 4.7. It is observable how noisy samples lie below the computed activation threshold $a_{new} = 0.1969$ (Eq. 4.7) and can, therefore, be discarded. We use this noise detection procedure to all the networks in our architecture with the aim to attenuate noise in the training data and prevent the forced classification of input that are not represented by the trained model.

### 4.4.2 Results and Evaluation

As aforementioned, we evaluated our approach both on our KT full-body action dataset (described in Section 4.2) and the public action benchmark CAD-60 (Sung et al., 2012). We now provide details on learning parameters for the GWR-based training and recognition, and a comparative evaluation.

#### KT Action Dataset

The KT full-body action dataset is composed of 10 full-body actions performed by 13 subjects. Similar to previously reported results (Section 4.3.3), we divided the data equally into training and test set, i.e. 30 sequences of 10 seconds for cyclic actions such as *standing, walking, jogging, sitting, lying down, and crawling*, and 30 repetitions for each goal-oriented action such as *picking up, jumping, falling down, and standing up*. Both the training and test sets contain data from all participants.

Our experiments show that our new approach outperforms the previous one with an average accuracy rate of 94% (5% higher the than GNG-based architecture described in Section 4.3 using an extra network for noise detection, and 18% higher than the same architecture without noise detection). We show the confusion matrix for both approaches in Fig. 4.8 and 4.9 (with each row of the matrix

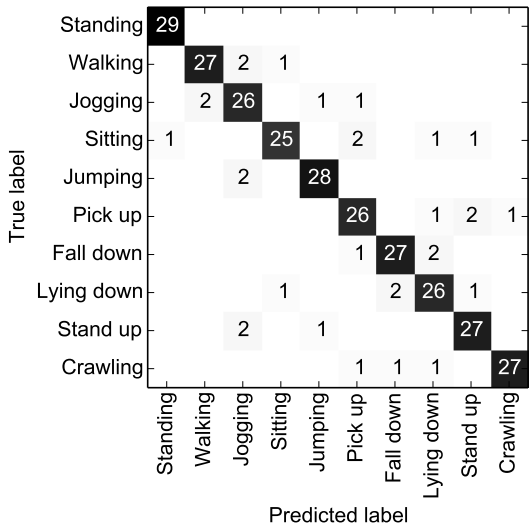



Figure 4.8: Confusion matrix for GNG-based architecture (Parisi et al., 2015b).

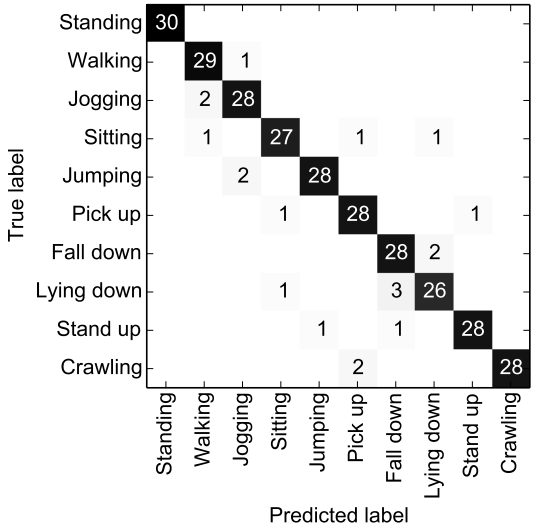



Figure 4.9: Confusion matrix for GWR-based architecture with embedded noise detection (Parisi et al., 2015b).

being an instance of the actual actions and each column an instance of the predicted actions.) We can observe from the matrices that all the actions are slightly classified more accurately with respect to Parisi et al. (2014c). The most often misclassified actions are *sitting* and *lying down*. In the first case, the action was confused with *walking* and *picking up*. This misclassification was mainly caused by skeleton tracking errors, i.e. when sitting down, the self-occlusion of joints may compromise the estimation of the overall body pose. The action *lying down* was, instead, misclassified as *falling down*. This is likely caused by the similar body poses of the two actions, despite the contribution of motion to disambiguate actions with similar poses.

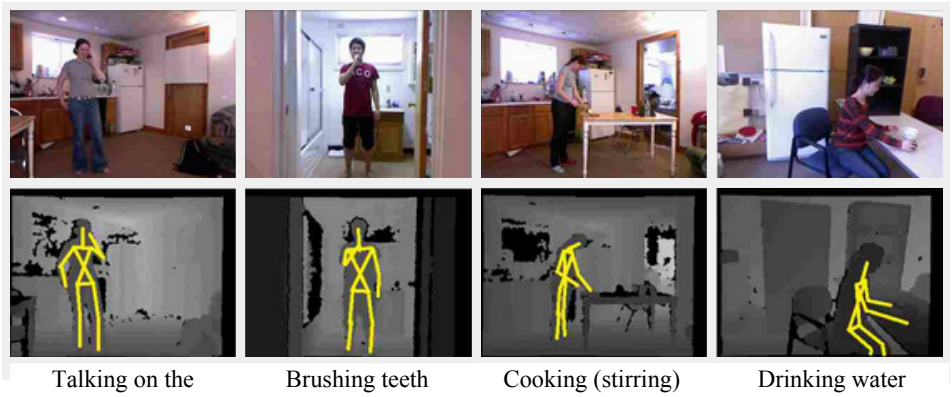



Figure 4.10: Daily actions from the CAD-60 dataset (RGB and depth images with skeleton). Adapted from (Sung et al., 2012).

### CAD-60

The Cornell activity dataset CAD-60 (Sung et al., 2012) is composed of 60 RGB-D videos of four subjects (two males, two females, one left-handed) performing 12 activities: *rinsing mouth, brushing teeth, wearing contact lens, talking on the phone, drinking water, opening pill container, cooking (chopping), cooking (stirring), talking on couch, relaxing on couch, writing on whiteboard, working on computer*. The activities were performed in 5 different environments: office, kitchen, bedroom, bathroom, and living room. The videos were collected with a Kinect sensor with distance ranges from 1.2 m to 3.5 m and a depth resolution of 640×480 at 15 frames per second. The dataset provides raw depth maps and RGB images, and skeleton data. An example of the actions and the resulting skeletons is shown in Fig. 4.10. The dataset provides skeleton data composed of 15 extracted joints for the following body parts: *head, neck, torso, shoulders, elbows, hands, hips, knees,* and *feet*.

For our approach, we used the set of 3D positions without the *feet*, leading to 13 joints (i.e., 39 input dimensions). Instead of using world coordinates, we encoded the joint positions using the center of the hips as frame of reference to obtain translation invariance. We then computed joint motion as the difference of two consecutive frames for each pose transition. We added a mirrored version of all action samples to obtain invariance to actions performed with either the right or the left hand.

For our evaluation on the CAD-60 dataset, we adopted a similar scheme as the one reported by Sung et al. (2012) using all the 12 activities plus a random action with *new person* strategy, i.e. the first 3 subjects for training and the remaining for test purposes. In Table 4.1, we show a comparison of our results with the state of the art on the CAD-60 dataset with precision and recall as evaluation metrics, and ranked by the $F_1$-score computed as:

$$F_1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall}. \tag{4.8}$$

Table 4.1: Our approach evaluated on the 12 activities from CAD-60 and comparison with other approaches.

| **Algorithm** | **Precision (%)** | **Recall (%)** | **F-score (%)** |
|---|---|---|---|
| Sung et al. (2012) | 67.9 | 55.5 | 61.1 |
| Ni et al. (2013) | 75.9 | 69.5 | 72.1 |
| Koppula et al. (2013) | 80.8 | 71.4 | 75.8 |
| Gupta et al. (2013) | 78.1 | 75.4 | 76.7 |
| Gaglio et al. (2014) | 77.3 | 76.7 | 77 |
| Zhang and Tian (2012) | 86 | 84 | 85 |
| Zhu et al. (2014) | 93.2 | 84.6 | 88.7 |
| **Our approach** | **91.9** | **90.2** | **91** |
| Faria et al. (2014) | 91.1 | 91.9 | 91.5 |
| Shan and Akella (2014) | 93.8 | 94.5 | 94.1 |

We obtained 91.9% precision, 90.2% recall, and 91% F-score, indicating that our model exhibits a good positive predictive value and very satisfactory sensitivity to classified actions. (For the precision and recall of each action and environment see Appendix D.) To be noted is that we separated the actions into 5 different environments for a consistent and more informative comparison with other approaches using the same dataset, whereas the specific properties of the environments were not known to the model and had no effect on the segmentation of the skeleton joints, therefore not influencing the classification process.

The best state-of-the-art result is 93.8% precision, 94.5% recall, and 94.1% F-score (Shan and Akella, 2014). In their work, the authors identified prior to learning a number of key poses to compute spatiotemporal action templates, which makes this approach highly data dependent. Each action must be segmented into atomic action templates composed of a set of $n$ key poses, where $n$ depends on the action's duration and complexity. Furthermore, experiments with low-latency (close to real time) classification have not been reported. The second approach with slightly better results than ours is the work by Faria et al. (2014) with 93.2% precision, 91.9% recall, and 91.5% F-score. In their work, the authors used a dynamic Bayesian Mixture Model to classify motion relations between body poses. However, they used the raw depth images to estimate their own skeleton model (and did not use the one provided by the CAD-60 benchmark dataset). Therefore, differences in the tracked skeleton may exist that hinder a quantitative comparison with our classification method.

Table 4.2: Training results on the two datasets – For each trained network along the hierarchy, the table shows the resulting number of nodes ($N$) and connections ($C$), and the activation threshold ($a$).

| | | | | | | |
|---|---|---|---|---|---|---|
| **KT Action Dataset** | $G_1^P$ | $N = 225$<br>$C = 435$<br>$a = .1865$ | $G_2^P$ | $N = 183$<br>$C = 338$<br>$a = .1934$ | $G^{STS}$ | $N = 118$<br>$C = 378$<br>$a = .2932$ |
| | $G_1^M$ | $N = 254$<br>$C = 551$<br>$a = .1732$ | $G_2^M$ | $N = 192$<br>$C = 353$<br>$a = .1910$ | | |
| **CAD-60** | $G_1^P$ | $N = 289$<br>$C = 403$<br>$a = .1778$ | $G_2^P$ | $N = 214$<br>$C = 445$<br>$a = .1898$ | $G^{STS}$ | $N = 137$<br>$C = 309$<br>$a = .2831$ |
| | $G_1^M$ | $N = 302$<br>$C = 542$<br>$a = .1698$ | $G_2^M$ | $N = 239$<br>$C = 495$<br>$a = .1991$ | | |

### Training Parameters

We now report the GWR parameters for the training sessions. We set the following values: insertion thresholds $a_T = 0.90$, learning rates $\epsilon_b = 0.3$ and $\epsilon_n = 0.006$, maximum age $a_{max} = 50$, firing counter parameters $h_0 = 1$, $\tau_b = 0.3$, $\tau_n = 0.1$.

Each network stopped training after 500 epochs over the whole dataset. These parameters were empirically found to let the model learn spatiotemporal dependencies with the best accuracy in terms of classification labels returned by the last network $G^{STS}$. For a single network, the number of neurons converged already after 100 epochs, and weight vectors of neurons showed little modification after 400 epochs. If we consider the 2 networks per stream in the first stage of the hierarchy and the integration network in the second stage, it took overall 1500 epochs to obtained a trained neuron in the $G^{STS}$ network.

In Table 4.2, we show the resulting properties of the networks along the hierarchy after the training sessions on the two datasets. In both cases, it can be observed that the number of nodes ($N$) and connections ($C$) is lower for higher levels of the hierarchy. The lower numbers indicate that in the STS level neurons encode more complex spatiotemporal dependencies with respect to the first level (in which only uni-cue spatial relations are considered), but with a smaller number

of specialized neurons. To be noted is that the number of neurons did not depend on the dimensionality of the input, but rather on the distribution of the data. From Table 4.2 it can also be seen that the activation threshold ($a$) increases towards higher levels of the hierarchy. In the first level, the activation function yielded larger fluctuations due to outliers and input data that were rarely presented to the network during the training. Conversely, activations of training samples matching the model get higher as neurons become more specialized. These results indicate that noise from the training data was not propagated along the hierarchy, but rather detected and discarded, which resulted in a larger $a$-value.

## 4.5 Towards Learning Transitive Actions

The recognition of transitive actions (actions that involve the interaction with an object) is an important part of daily human activities. Humans possess an outstanding capability to infer the goal of actions from the interaction with objects. The study of transitive actions such as grasping and holding has often been the focus of research in neuroscience and psychology (Fleischer et al., 2013; Nelissen et al., 2005; Gallese et al., 1996). Nevertheless, this task has remained an open challenge for computational models of action recognition.

Neurophysiological studies suggest that only when information about the object identity is added to the semantic information of the action, then the actions of other individuals can be completely understood (Saxe et al., 2004). From the computational perspective, an important question regards the potential links between representations of body postures and manipulated objects and, in particular, how these two representations can be integrated.

In this section, we propose a possible extension of the hierarchical architectures previously discussed in this chapter to account for the learning of action–object mappings from RGB-D videos (Mici et al., 2016). The architecture consists of two separate pathways that process body action features and object features in parallel and subsequently it integrates prototypes of actions and the identity of recognized objects. Experimental results have shown that the proposed integration of body actions and objects significantly increases the classification accuracy of action sequences.

### 4.5.1 Proposed Architecture

An overview of the architecture is depicted in Fig. 4.11. For the processing of body postures, a set of local features describing the upper-body joints are extracted and fed into the 2-layer neural architecture with GWR networks. The input for the object recognition module is the RGB image of the object. Objects are represented as compact feature vectors and are fed into a SOM network. The last layer learns the combination of body postures and objects involved in an action.

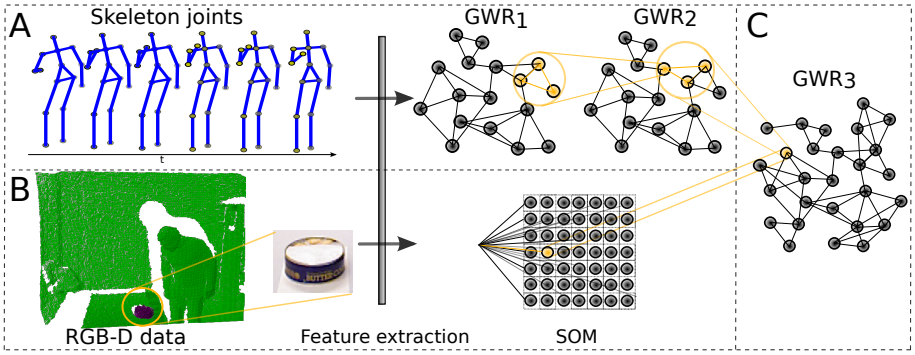



Figure 4.11: Overview of the proposed architecture for transitive action recognition (Mici et al., 2016).

### Object Recognition

Objects are extracted from RGB action sequences, where the region of interest is automatically extracted through a point cloud-based table top segmentation. For the representation of objects, we use SIFT features (Jégou et al., 2010) that yield invariance to translation, rotation and scaling transformations and, to some extent, robustness to occlusions. For the problem of object category recognition, we use *dense* SIFT descriptors on regular grids across each image. Since object representations are compared using the Euclidean distance as a metric, we compute a fixed-dimensional vector for each image. For this purpose, we selected the vector of locally aggregated descriptors (VLAD) (Lowe, 2004), which results in a feature vector with a high discriminative power.

For learning objects, we train a SOM network on a set of training objects (Fig. 4.11.*B*). We attach symbolic labels to each neuron based on the majority of input samples that have matched with each neuron during the training phase. At recognition time, for each input image the best-matching neuron from the trained network will be computed, so that the knowledge of the category of objects can be transferred to the higher layer of the architecture in terms of a symbolic label.

### Body Motion Recognition

For the recognition of body motion sequences, we train a hierarchical GWR architecture (Fig. 4.11.*A*). We first train the $GWR_1$ network with the sequences of body postures. After the training is completed, the $GWR_2$ network is trained with neural activation trajectories from $GWR_1$. Thus, for each input sample $\mathbf{x}_i$, the best-matching neuron in $GWR_1$ network is computed as in Eq. 3.6. The weights of the neurons activated within a temporal sliding window of length $q$ are concatenated and fed as input to $GWR_2$. The input data for training $GWR_2$ is of the form:

$$\psi(\mathbf{x}_i) = \{b(\mathbf{x}_i), b(\mathbf{x}_{i-1}), ..., b(\mathbf{x}_{i-q+1}), i \in [q..m]\}, \tag{4.9}$$

where $m$ is the number of training samples. While the first network learns a set of prototype body postures, the second network will learn temporally-ordered prototype sequences from $q$ consecutive samples.

#### Integration of Action-Object Representations

The last network in hierarchy, $GWR_3$, will integrate the information from the converging streams and learn action–object mappings (Fig. 4.11.$C$). For this purpose, we create a new dataset by concatenating the set of activation trajectories from the $GWR_2$ network and the object's symbolic label from the SOM trained with the set $O$ of training objects. The resulting training data consists of pairs $\phi_u$ of the following form:

$$\phi_u = \{b(\psi(\mathbf{x}_i)), ..., b(\psi(\mathbf{x}_{i-q-1})), l_b(\mathbf{y}), \mathbf{x}_i \in T, \mathbf{y} \in O, u \in [q..m-q]\}, \quad (4.10)$$

where $l_b(\mathbf{y})$ represents the label attached to the best-matching neuron of the object recognition module for the object input $\mathbf{y}$. Each neuron in $GWR_3$ is assigned with an action label adopting the same labeling strategy as in the SOM, meaning that neurons take the label of the best-matching input samples. After the training of $GWR_3$ is completed, each neuron will encode a prototype segment of the action in terms of action–object pairs.

### 4.5.2 Results and Evaluation

#### Data Collection

We collected a dataset of the following daily activities: *picking up*, *drinking* (from a mug or a can), *eating* (cookies) and *talking on a phone*. The actions were performed by 6 participants that were given no explicit indication on the purpose of the study nor instructions on how to perform the actions.

The dataset was collected with an Asus Xtion depth sensor that provides a synchronized RGB-D image (color and depth map). The tracking of skeleton joints was computed with the OpenNI framework (Fig. 4.12). Action labels were manually annotated from a ground-truth of sequence frames. We added a mirrored version of all action samples to obtain invariance to actions performed with either the right or the left hand. The depth sensor was also used for acquiring the objects dataset. Since object recognition should be view-invariant, RGB images were acquired with the camera positioned in two different heights and from objects in different views with respect to the sensor. Object labels were manually annotated for the training sequences, and the labels output from the object recognition module were used for the test sequences.

#### Training and Evaluation

In order to evaluate the generalization capabilities of our architecture, we conducted experiments with 10-fold cross-validation, meaning that data were split

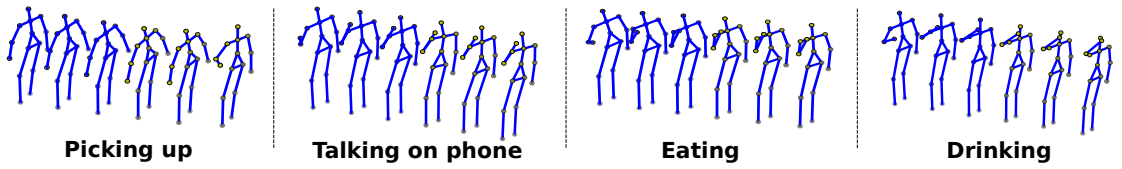



Figure 4.12: Examples of sequences of skeleton joints taken from our action dataset (Mici et al., 2016).

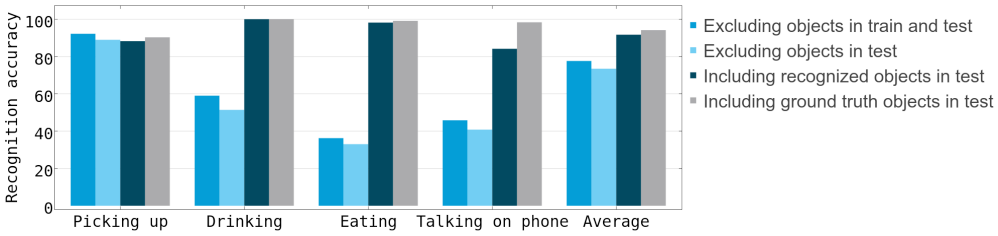



Figure 4.13: Evaluation of the recognition accuracy on the test data set under different conditions (Mici et al., 2016).

into 10 sets of random samples, from which 60% for training and 40% for testing. The reported results were averaged over the 10 folds.

We set the following GWR training parameters: learning rates $\epsilon_b = 0.1$, $\epsilon_n = 0.01$, firing threshold $h_T = 0.1$, insertion thresholds $a_T = \{0.5, 0.4, 0.3\}$ for $GWR_1$, $GWR_2$, and $GWR_3$ respectively, maximum age $a_{max} = 100$, firing counter parameters $h_0 = 1$, $\tau_b = 0.3$ and $\tau_n = 0.1$. Each GWR network was trained for 50 epochs over the whole dataset of actions. The number of neurons created in each GWR network given a training set with $\approx$ 18.600 frames were $\approx$ 480 for $GWR_1$, $\approx$ 600 for $GWR_2$, while for $GWR_3$ the number varied from $\approx$ 700 to $\approx$ 1000 depending on the inclusion or exclusion of the objects (as explained in Fig. 4.13). For training the SOM, we used a $20 \times 20$ map of neurons in a hexagonal network lattice with a Gaussian neighboring function and 50 training epochs over the whole dataset of objects.

We evaluated the recognition accuracy of the architecture under three conditions: (1) completely excluding the object identity in both training and testing, (2) including the objects in training while excluding them in testing phase, and (3) no exclusion in both phases. In the third case, the label given by the SOM-based object classifier was used during testing. Additional experiments were run using the objects' ground-truth labels for comparison. Results are reported in Fig. 4.13, where it is possible to see a significant improvement of the action classification performance for the third condition. When the object *can* is replaced with the object *mug*, the final classification accuracy of the action *drinking* is not affected – this is a desirable generalization capability of our architecture. Furthermore, the relatively low recognition rates in the second condition suggest that the identity of the object is crucial for distinguishing between the actions *drinking*, *eating* and

*talking on phone*, while for the action *picking up* the situation does not vary drastically in either case. Together, these results suggest that the identity of objects plays a fundamental role for the effective recognition of daily actions that involve the interaction with objects.

## 4.6 Summary

In this chapter, we presented a series of neurobiologically-motivated approaches that learns to recognize actions from depth map video sequences. The proposed neural network architectures rely on three assumptions that are consistent with evidence on neural mechanisms for action recognition: 1) pose and motion action features are processed in two distinct pathways, respectively the ventral and the dorsal stream, and then action cues are integrated to provide a joint percept (Vangeneugden et al., 2009); 2) hierarchies within each pathway process features with increasing complexity (Giese and Poggio, 2003; Hasson et al., 2008); and 3) visual information is arranged according to input-driven self-organization (Willshaw and von der Malsburg, 1976; Nelson, 2000).

Our architectures consist of a two-pathway hierarchy of growing self-organizing networks that process pose-motion features in parallel and subsequently integrate action cues to provide movement dynamics in the joint feature space. Hierarchical learning was carried out using prototype trajectories composed of neuron activation patterns. The learning mechanism of the networks allows to attenuate noise and detect noisy novel samples. For classification purposes, we extended the standard implementations of unsupervised GNG and GWR learning with two labelling functions for associating symbolic labels to learned visual representations.

We conducted a series of experiments with a full-body action dataset and a public benchmark dataset of daily actions. Additionally, we presented an extended architecture for the learning of action–object mappings from RGB-D videos. The reported results motivate the embedding of our learning systems into mobile robot platforms to conduct further evaluations in more complex scenarios, where the robust recognition of actions plays a key role. For instance, the learning of multimodal action representations from multiple sensors (see Chapter 5 and 6), the detection of dangerous events for assistive robotics such as falls, and the recognition of actions with learning robots in HRI scenarios (see Chapter 7).

# Chapter 5

# Self-Organizing Emergence of Multimodal Action Representations

## 5.1 Introduction

As humans, our daily perceptual experience is modulated by an array of sensors that convey different types of modalities such as visual, auditory, and somatosensory information (see Chapter 2). A number of computational models have been proposed that aim to effectively integrate multisensory information, in particular, audiovisual input. These approaches generally use unsupervised learning for obtaining visual representations of the environment and then link these features to auditory cues. Vavrečka and Farkaš (2014) and Morse et al. (2015) presented connectionist architectures that learn to bind visual properties of objects (e.g., spatial location, shape and color) to proper lexical features. Unimodal representations are obtained by neural network self-organization and multimodal representations develop through the activation of unimodal modules via associative connections. The development of associations between co-occurring stimuli for multimodal binding has been strongly supported by neurophysiological evidence (Fiebelkorn et al., 2009). However, these approaches do not naturally scale up to learn more complex spatiotemporal patterns such as action–word mappings.

Action words do not label actions in the same way that nouns label objects (Gentner, 1982). While nouns typically refer to objects that can be perceived as distinct units, action words refer instead to spatiotemporal relations within events that may be performed in many different ways with high spatial and temporal variance. Words for actions and events appear to be among children's earliest vocabulary (Bloom, 1993; Hirsch-Pasek et al., 2000). Infants are able to learn action–word mappings using cross-situational statistics, thus in the presence of sometimes unavailable ground-truth action words (Smith and Yu, 2008). Furthermore, action words can be progressively learned and improved from linguistic and social cues so that novel words can be attached to existing visual representations. This hypothe-

sis is supported by neurophysiological studies evidencing strong links between the cortical areas governing visual and language processing, and suggesting high levels of functional interaction of these areas for the formation of multimodal representations of audiovisual stimuli (Foxe et al., 2000; Raij et al., 2000; Belin et al., 2000, 2002; Pulvermüller, 2005).

It has been argued that the superior temporal sulcus (STS) in the mammalian brain may be the basis of an action-encoding network with neurons driven by the perception of dynamic human bodies and that for this purpose it receives converging inputs from earlier visual areas from both the ventral and dorsal pathways (see Section 2.1.2). Thus, the STS area is thought to be an associative learning device for linking different unimodal representations and accounting for the mapping of naturally occurring, highly correlated features such as body pose and motion, the characteristic sound of an action (Beauchamp et al., 2004; Barraclough et al., 2005) and linguistic stimuli (Belin et al., 2002; Wright et al., 2003; Stevenson and James, 2009). Multimodal representations of actions in the brain play an important role for a robust perception of complex action patterns, with the STS representing a multisensory area in the brain network for social cognition (Allison et al., 2000; Adolphs, 2003; Beauchamp, 2005; Beauchamp et al., 2008).

In Chapter 4, we focused on the development of robust action representations from different visual cues such as body posture and motion, and object features. In this chapter, we investigate how congruent multimodal representations of actions can naturally emerge from the co-occurrence of audiovisual stimuli. In particular, we propose an approach where associative links between unimodal representations are incrementally learned in a self-organizing manner. For this purpose, we extend our proposed spatiotemporal hierarchy for the integration of pose-motion action cues as presented in Parisi et al. (2015b) to include an associative network layer where action–word mappings develop from co-occurring audiovisual inputs using asymmetric inter-layer connectivity. Each network layer comprises a self-organizing neural network that employs neurobiologically-motivated Hebbian-like plasticity and habituation for stable incremental learning (Marsland et al., 2002).

The proposed architecture is novel in two main aspects: First, our learning mechanism does not require manual segmentation of training samples. Instead, spatiotemporal generalizations of actions are incrementally obtained and mapped to symbolic labels using the co-activation of audiovisual stimuli. This allows us to train the model in an incremental fashion also in the presence of occasionally unlabeled samples. Second, we let asymmetric inter-layer connectivity emerge taking into account the spatiotemporal dynamics of sequences so that symbolic labels are linked to temporally-ordered representations in the visual domain. This kind of connectivity allows the bidirectional retrieval of audiovisual inputs, i.e. it is possible to retrieve action words from processed visual patterns and, conversely, to activate congruent visualizations of learned actions from recognized action words.

We conducted a set of experiments with the KT action dataset containing 10 full-body actions (see Section 4.2) using body pose-motion cues as visual features and action labels obtained from automatic speech recognition. Experimental results showed that we achieve state-of-the-art recognition performance without the

need to manually segment training samples, and that this performance is not drastically compromised as the number of available labeled samples is decreased.

## 5.2 Associative Action–Word Mappings

Our neural architecture consists of a self-organizing hierarchy with four network layers for the unsupervised processing of visual action features and the development of associative connections between learned action representations and symbolic labels. An overall diagram of the architecture is shown in Fig. 5.1.

Network layers 1 and 2 comprise a two-stream hierarchy for the processing and subsequent integration of body pose and motion features, resembling the ventral and the dorsal pathway respectively for the processing of complex motion patterns (Giese and Poggio, 2003). The integration of pose and motion cues is carried out in network layer 3 (or $G^{STS}$) to provide movement dynamics in the joint feature space (Parisi et al., 2015b). Hierarchical learning from contiguous Growing When Required (GWR) networks (Marsland et al., 2002) shapes a functional hierarchy that processes spatiotemporal visual patterns with an increasing level of complexity by using neural activation trajectories from lower-level layers for training higher-level layers. For learning multimodal representation of actions, network layer 4 (or $G^{STSm}$) implements a self-organizing algorithm where action–word mappings are developed by binding co-occurring audiovisual inputs using bidirectional inter-layer connectivity. For this purpose, we extended the traditional GWR learning algorithm with a mechanism for semi-supervised label propagation and enhanced synaptic connectivity for learning prototype neural activation patterns in the temporal domain. The proposed learning algorithm is referred to as Online Semi-Supervised GWR (OSS-GWR).

The self-organizing associative connectivity between $G^{STSm}$ and the Action Words Layer (AWL) will yield an incremental formation of congruent action–word mappings for the bidirectional retrieval of audiovisual patterns.

### 5.2.1 A Self-Organizing Spatiotemporal Hierarchy

Our learning model consists of hierarchically-arranged GWR networks (Marsland et al., 2002) that obtain progressively generalized representations of sensory inputs and learn inherent spatiotemporal dependencies. The GWR network is composed of a set of neurons with their associated weight vectors linked by a set of edges. During the training, the network dynamically changes its topological structure to better match the input space following competitive Hebbian learning (Martinetz, 1993). The learning procedure for GWR is illustrated by Algorithm 1 (except for Steps 3, 7.c, 8.c, 9, and 10 that are implemented by the OSS-GWR only).

The motivation underlying hierarchical learning is to obtain progressively specialized neurons coding spatiotemporal dependencies of the input. This is consistent with neurophysiological evidence supporting increasingly large temporal receptive windows in the mammalian cortex (Taylor et al., 2015; Hasson et al., 2008;

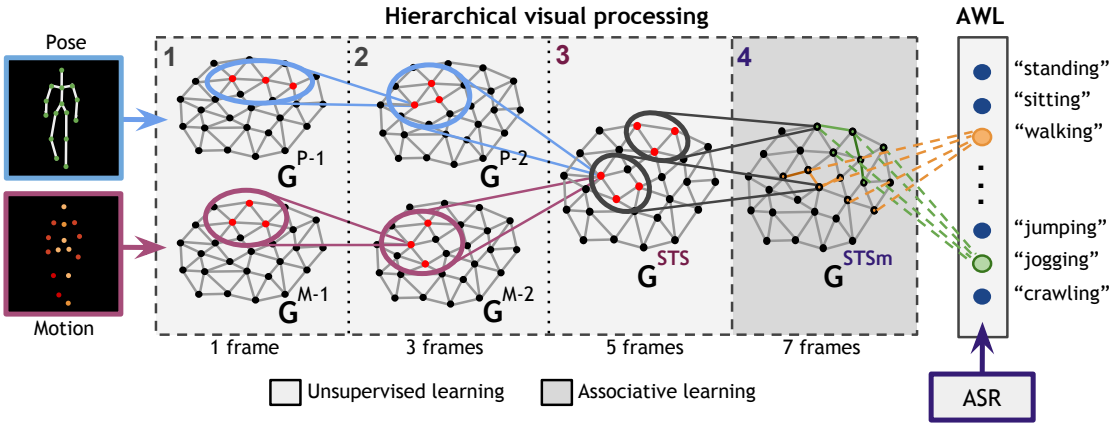



Figure 5.1: Diagram of our learning architecture with GWR networks and the number of frames required for hierarchical processing. Layers 1-3: parallel spatiotemporal clustering of visual features and self-organizing pose-motion integration ($G^{STS}$). Layer 4: Self-organization of $G^{STS}$ representations and associative learning for linking visual representations in $G^{STSm}$ to the Action Words Layer (AWL) obtained from automatic speech recognition (ASR) (Parisi et al., 2016b).

Lerner et al., 2011), where neurons in higher areas encode information accumulated over longer timescales. In our architecture, hierarchical learning is carried out by training a higher-level network with neural activation trajectories from a lower-level network. These trajectories are obtained by computing the best-matching neurons for the current input sequence with respect to the trained network with $N$ neurons, so that a set of trajectories of length $q$ is given by

$$\Omega^q(\mathbf{x}(t)) = \{\mathbf{w}_{b(\mathbf{x}(t))}, \mathbf{w}_{b(\mathbf{x}(t-1))}, ..., \mathbf{w}_{b(\mathbf{x}(t-q+1))}\}, \tag{5.1}$$

with $b(\mathbf{x}(t)) = \arg\min_{j \in N} \|\mathbf{x}(t) - \mathbf{w}_j\|$ computing the index of the neuron that minimizes the distance to the current input.

The overall hierarchical flow is illustrated in Fig. 5.2. The low-level networks $G^{P\text{-}1}$ and $G^{M\text{-}1}$ learn a set of time-independent primitives that will be used for higher-level representations and should exhibit robust activation regardless of temporal disruptions of the input. The networks $G^{P\text{-}2}$ and $G^{M\text{-}2}$ process activation trajectories of 3 neurons from the previous layer and the integration of the input is carried out in $G^{STS}$ over activation trajectories of 3 neurons from layer 2. The network layer $G^{STS}$ integrates pose-motion features by training the network with the concatenation of vectors $\Psi = \{\Omega^q(\mathcal{P}) \frown \Omega^q(\mathcal{M})\}$, where $\mathcal{P}$ and $\mathcal{M}$ are the activation trajectories from $G^{P\text{-}2}$ and $G^{M\text{-}2}$ respectively. Network layer $G^{STSm}$ processes activation trajectories of 3 neurons from $G^{STS}$, thereby representing visual information over a temporal window of 7 frames (see Section 4.3 for a detailed discussion). After the training is completed, neurons in $G^{STSm}$ encode sequence-selective prototype action segments, following the assumption that the recognition of actions must be selective for temporal order (Giese and Poggio, 2003; Hasson et al., 2008).

**Algorithm 1** OSS-GWR - In layers 1, 2, and 3 of our architecture, we use GWR learning, while in layer 4 ($G^{STSm}$) we use OSS-GWR.

1: Create two random neurons with weights $\mathbf{w}_1$ and $\mathbf{w}_2$.
2: Initialize an empty set of spatial connections $E = \emptyset$.
3: [OSS-GWR only] Initialize an empty set of temporal connections $P = \emptyset$ and a set of label–to–action references $V = \emptyset$.
4: At each iteration $t$, generate an input sample $\mathbf{x}(t)$ with sample label $\xi$
5: Select the best and the second-best matching neuron such that:
$b = \arg\min_{n \in A} \|\mathbf{x}(t) - \mathbf{w}_n\|, \quad s = \arg\min_{n \in A/\{b\}} \|\mathbf{x}(t) - \mathbf{w}_n\|$.
6: Create a connection $E = E \cup \{(b, s)\}$ if it does not exist and set its age to 0.
7: If $(\exp(-\|\mathbf{x}(t) - \mathbf{w}_b\|) < a_T)$ and $(\eta_b < f_T)$ then:
a: Add a new node $r$ $(A = A \cup \{r\})$ with $\mathbf{w}_r = 0.5 \cdot (\mathbf{x}(t) + \mathbf{w}_b)$, $\eta_r = 1$,
b: Update edges: $E = E \cup \{(r, b), (r, s)\}$ and $E = E/\{(b, s)\}$,
c: [OSS-GWR only] Initialize neuron label (Eq. 5): $\lambda(r) = \gamma^{\text{new}}(b, \xi)$.
8: If no new neuron is added:
a: Update the best-matching neuron $\mathbf{w}_b$ and its neighbors $i$:
$\Delta\mathbf{w}_b = \epsilon_b \cdot \eta_b \cdot (\mathbf{x}(t) - \mathbf{w}_b), \quad \Delta\mathbf{w}_i = \epsilon_n \cdot \eta_i \cdot (\mathbf{x}(t) - \mathbf{w}_i)$,
with the learning rates $0 < \epsilon_n < \epsilon_b < 1$.
b: Increment the age of all edges connected to $b$ by 1.
c: [OSS-GWR only] Update neuron label (Eq. 6): $\lambda(b) = \gamma^{\text{update}}(b, s, \xi)$.
9: [OSS-GWR only] Create a temporal connection $P_b^{b(t-1)}$ if it does not exist, increase it by a value of 1 and decrease all the others if $P_n^{b(t-1)} > 0$.
10: [OSS-GWR only] Create a label–to–action reference $V_b^{\lambda(b)}$ if it does not exist and update it (Eq. 8): $V_b^{\lambda(b)} = \Lambda^{\lambda(b)}(b)$.
11: Reduce the firing counters of the best-matching neuron and its neighbors $i$:
$\Delta\eta_b = \tau_b \cdot \kappa \cdot (1 - \eta_b) - \tau_b, \quad \Delta\eta_i = \tau_i \cdot \kappa \cdot (1 - \eta_i) - \tau_i$,
with constant $\tau$ and $\kappa$ controlling the curve behavior.
12: Remove all edges with ages larger than $\mu_{max}$ and remove neurons without edges.
13: If the stop criterion is not met, repeat from step 4.

## 5.3 GWR-based Associative Learning

For the $G^{STSm}$ layer, we extended the standard GWR algorithm with: 1) semi-supervised label propagation functions so that prototype neurons can be attached to symbolic labels also in the absence of labeled samples, and 2) enhanced synaptic connectivity in the temporal domain for learning activation patterns of consecutively activated neurons. The detailed learning algorithm for the proposed Online Semi-Supervised GWR (OSS-GWR) is illustrated by Algorithm 1.

### 5.3.1 Semi-Supervised Label Propagation

For the semi-supervised propagation of labels, we attach labels to neurons by taking into account local connectivity and neural activation patterns. In this way,

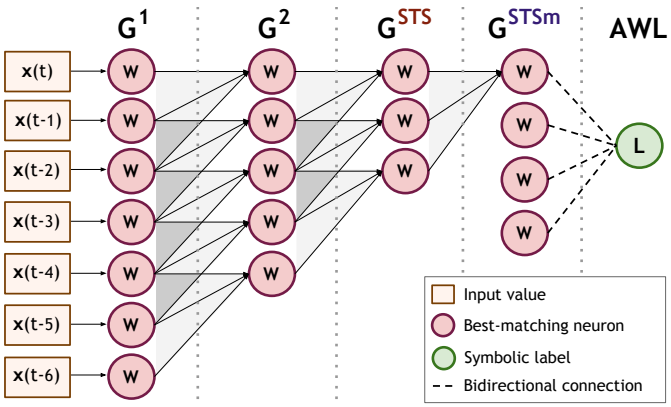



Figure 5.2: Hierarchical learning of the last 7 inputs for processing neural activations with a sliding window scheme and asymmetric inter-layer connectivity between $G^{STSm}$ and AWL used for bidirectional retrieval of audiovisual patterns. A neuron in $G^{STSm}$ encodes action segments of 7 inputs. Action labels are predicted from 4 neurons in $G^{STSm}$ (10 inputs), while for each action word in AWL, one onset neuron in $G^{STSm}$ is computed (Parisi et al., 2016b).

only labels attached to well-trained neurons are propagated to unlabeled neighbors (Algorithm 1, Steps 7.c and 8.c). For this purpose, we defined two labelling functions: $\gamma^{\text{new}}$ for when a new neuron is created, and $\gamma^{\text{update}}$ for when a neuron is updated.

Provided that $b$ is the index of the best-matching neuron of the training sample $\mathbf{x}(t)$ with label $\xi$ and that we denote a missing sample label with the value $-1$, the label of a new neuron $\lambda(\mathbf{w}_r)$ is assigned according to

$$\gamma^{\text{new}}(b, \xi) = \begin{cases} \xi & \xi \neq -1 \\ \lambda(\mathbf{w}_b) & \text{otherwise} \end{cases} \tag{5.2}$$

For updating the label of an existing neuron, we also consider whether the current training sample is labeled. If this is not the case, then the best-matching neuron $b$ will take the label of its closest neighbor $s$, provided that the two neurons have been sufficiently trained as expressed by their firing counters. Given the index of the second-best matching neuron $s$ of $\mathbf{x}(t)$, the update labelling function for $\lambda(\mathbf{w}_b)$ is defined as

$$\gamma^{\text{update}}(\xi, b, s) = \begin{cases} \xi & \xi \neq -1 \\ \lambda(\mathbf{w}_s) & (\xi = -1) \wedge (\pi_s^b \geq \pi_T) \\ \lambda(\mathbf{w}_b) & \text{otherwise} \end{cases} \tag{5.3}$$

$$\pi_j^i = \frac{E_j^i}{1 + \eta_i + \eta_j}, \tag{5.4}$$

with $E_j^i = 1$ if the neurons $i$ and $j$ are connected and 0 otherwise. Thus, this function yields greater values for interconnected, well-trained neurons, i.e. that

Table 5.1: Training parameters for the S-GWR and the OSS-GWR used for the classification task of the Iris dataset (results in Fig. 5.3).

| **Parameter** | **Value** |
|---|---|
| Insertion threshold | $a_T = \{0.35, 0.75\}$ |
| Firing threshold | $f_T = 0.1$ |
| Learning rates | $\epsilon_b = 0.1$, $\epsilon_n = 0.01$ |
| Firing counter | $\tau_b = 0.3$, $\tau_i = 0.1$, $\kappa = 1.05$ |
| Training epochs | 20 |
| Labeling threshold (OSS-GWR only) | $\pi_T = 0.5$ |

have smaller firing counters. The value $\pi_T$ is used as a threshold to modulate the propagation of a label from $s$ to $b$.

We evaluated our semi-supervised labelling strategy on a classification task using the Iris benchmark dataset[1] containing 3 classes with 50 four-dimensional samples each. The goal of our experiment was to compare the classification performance of the proposed OSS-GWR with respect to the traditional GWR extended for classification (S-GWR, Parisi et al. 2015b) using a decreasing percentage of available labeled samples in the training set. The average accuracy was estimated over 10 runs by removing labels at random positions for each percentage of available labels (from 0% to 100%).

The training algorithm used for this experiment is illustrated by Algorithm 1, excluding Steps 3, 9, and 10 which are used in the $G^{STSm}$ layer only, while the training parameters are listed in Table 5.1. Fig. 5.3 shows the average recognition accuracy for two different insertion thresholds $a_T = \{0.35, 0.75\}$ used to modulate the number of neurons created by the network, which also has an impact on the classification performance. In a smaller network, a prototype neuron will represent a greater number of samples. Thus, it is more likely that a neuron representing a dense cluster of samples with the same label will be assigned the correct one. It can be seen that the OSS-GWR outperforms S-GWR for the classification task as soon as not all labels are available. Larger deviations from the average accuracy can be observed due to the fact that for each run labels were removed from randomly selected samples and the distribution of missing labels can strongly influence the final outcome, particularly when few samples were labeled. Furthermore, the number of neurons created at each run varied, i.e. $\approx 16$ for $a_T = 0.35$ and $\approx 100$ for $a_T = 0.75$. This is due to the fact that the weight vectors for the two neurons initializing the networks were randomly selected from the training samples.

These results show that the proposed labelling strategy (Eq. 5.2, 5.3, and 5.4) yields higher classification performance in the absence of sample labels. The over-

[1] `http://archive.ics.uci.edu/ml/datasets/Iris`

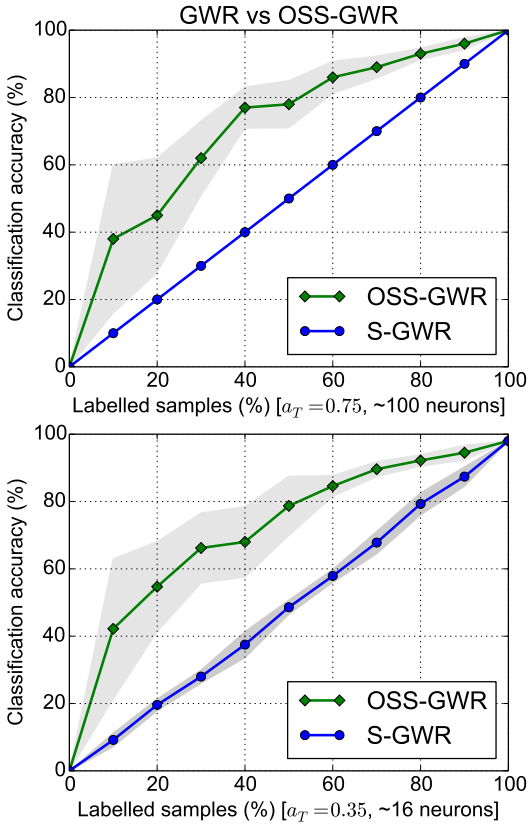



Figure 5.3: Average classification accuracy with a decreasing percentage of available labels in the training set for trained S-GWR and OSS-GWR networks with $\approx 100$ neurons (top) and $\approx 16$ neurons (bottom) (Parisi et al., 2016b).

all approach is said to be online since the algorithm incrementally propagates labels during the training process (Beyer and Cimiano, 2011), in contrast to offline methods where labels are used after the unsupervised training of a network has finished.

### 5.3.2 Sequence-Selective Synaptic Links

Next, we enhanced standard GWR connectivity by taking into account latent temporal relations of the input, so that connections between neurons that are consecutively activated can be created and incrementally updated. In other words, when the two neurons $i$ and $j$ are activated at time $t-1$ and $t$ respectively, the synaptic link $P_j^i$ between them is strengthened. At each iteration, the link $P_b^{b(t-1)}$ between the best-matching neuron $b$ and the previous winner neuron $b(t-1)$ is increased by a value of 1, while the synaptic links between $b(t-1)$ and the other neurons are decreased if $P_n^{b(t-1)} > 0$ for $n \in A/\{b\}$ (Algorithm 1, Step 9). This approach results in the efficient learning of the temporal structure of the input in terms of neural activation trajectories. The highest value of $P_n^b$ will encode the most frequent transition, and thus allowing to estimate a prediction of $b(t+1)$

provided $b(t)$.

Sequence selectivity driven by asymmetric connections has been argued to be a feature of the cortex for neurons encoding optic flow patterns, where an active neuron preactives neurons encoding future patterns, while it inhibits neurons encoding other patterns (Mineiro and Zipser, 1998). This mechanism can be used for iteratively retrieving prototype neurons that encode an action sequence given the onset neuron for that action.

## 5.4 Bidirectional Retrieval of Audiovisual Inputs

We now describe the asymmetric connectivity between the $G^{STSm}$ layer and the Action Words layer (AWL) which allows the bidirectional retrieval of audiovisual patterns. We will show how it is possible to predict action words from processed visual patterns and, conversely, how to activate congruent visualizations of learned actions from recognized action words.

### 5.4.1 Action–to–Word Patterns

During the learning phase, unsupervised visual representations of actions in $G^{STSm}$ are linked to symbolic action labels $\lambda \in L$, with $L$ being the set of possible words. Action words in AWL will then have a one-to-many relation with neurons in $G^{STSm}$, while neurons can be linked to only one label in $L$. The development of connections between $G^{STSm}$ and AWL depends on the co-activation of audiovisual inputs. More specifically, the connection between a neuron in $G^{STSm}$ and a symbolic label in AWL will emerge if the neuron is activated within a time window in which the label is also activated by an auditory signal. In case no auditory stimulus occurs during the training of neurons in $G^{STSm}$, the sample label will be given the value $-1$ to indicate a missing label. Symbolic labels attached to neurons will be updated according to the semi-supervised label propagation rules (Eq. 5.2 and 5.3).

Given a previously unseen sequence of visual inputs, we want to predict the correct action word by comparing the novel input to prototype action sequences in $G^{STS}$ and then return action labels attached to the best-matching neurons. The hierarchical flow of the visual input is composed of four networks, each of them processing activation trajectories of 3 neurons from the previous layer as introduced in Section 4.3. Thus, each neuron in $G^{STS}$ represents a prototype sequence encoding 7 consecutive frames (Fig. 5.2). By applying a temporal sliding window scheme, we get a new action label for each processed frame. To improve the robustness of the label prediction process, we return an action word from 4 neurons consecutively activated in $G^{STS}$ (10 frames). Given a set of 4 labels obtained from the last 4 activated neurons from visual input, we output the statistical mode of the set, i.e. the most frequent label in the set is returned as the predicted action word. If we assume visual input at 10 frames per second, an action word will be predicted for 1 second of video.

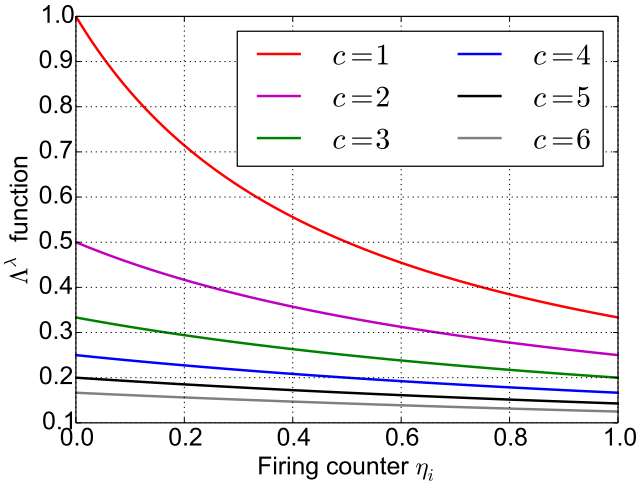



Figure 5.4: Values of the $\Lambda^\lambda$ function (Eq. 5.5) for different firing counters $\eta_i$ and sequence counters $c(\lambda_j, t)$ in the range 1 to 6 expressing the sequential order of processed samples. It can be seen that greater values are given to neurons activated at the beginning of the sequence, with an increasing response for well-trained neurons (smaller firing counter) (Parisi et al., 2016b).

### 5.4.2 Word–to–Action Patterns

For the development of connectivity patterns from AWL to $G^{STSm}$, we take into account the temporal order of consecutively activated neurons, yielding the learning of onset neurons in $G^{STSm}$ to be linked with an action label, and from which it is possible to retrieve temporally-ordered prototype sequences for an action word. For a labeled neuron $b$ in $G^{STSm}$ activated at time $t$, its connection strength with the symbolic label $\lambda$ becomes:

$$\Lambda^\lambda(b) = \frac{1}{2 \cdot \eta_b + c(\lambda, t)}, \tag{5.5}$$

with $c(\lambda, t)$ being a sequence counter that will increase by 1 when $\lambda(b) = \lambda(b-1)$ and reset to zero when this condition does not hold. Thus, this function expresses the relation between the firing counter $\eta_b$ of the neuron $b$ and its sequential order within the set of neural activations with the same label, yielding greater connection strengths for well-trained neurons that activate at the beginning of a sequence. The $\Lambda^\lambda$ function for different neuron firing counters is depicted in Fig. 5.4 for a temporal window of 6 neural activations.

Word–to–action connectivity patterns are stored in the label–to–action reference $V$ and updated at each training iteration so that $V_b^{\lambda(b)} = \Lambda^{\lambda(b)}(b)$ (Algorithm 1, Step 10). The neuron in $G^{STSm}$ with the maximum value of $\Lambda^\lambda$ can then be selected as the onset neuron of an action label $\lambda$ representing the first element of a prototype sequence.

We expect that word–to–action connections will develop according to the $\Lambda^\lambda$ function (Eq. 5.5) for each action label. Thus, when an action label $\lambda(j)$ is recog-

nized from speech, the onset neuron in $G^{STSm}$ of that action can be selected as the neuron that maximizes $\Lambda^{\lambda(j)}$, and consequently as the first element of a sequence used to generate prototype visual representations of actions. The index of the onset neuron $\mathbf{w}_v(t)$ for an action label $\lambda(j)$ is defined as:

$$v(t) = \arg\max_n V_n^{\lambda(j)}. \tag{5.6}$$

Next, we can retrieve the next neuron of a prototype action sequence by selecting the maximal temporal synaptic connectivity:

$$v(t+1) = \arg\max_n P_n^{v(t)}, \tag{5.7}$$

from which we can reconstruct a temporally-ordered sequence of arbitrary length by retrieving the weight vectors for a number of timesteps into the future. For instance, the sequence $(\mathbf{w}_{v_t}, ..., \mathbf{w}_{v_{t+3}})$ will generate visual output for a temporal window of 10 frames (1 second). This mechanism can be used in practice to visually assess how well the model has learned action dynamics and whether it has accounted for effectively binding action words to visual representations.

## 5.5 Experiments and Evaluation

We present our experimental set-up and results on a dataset of 10 full-body actions that has been previously used to report recognition performance with manual segmentation for ground-truth labelling (Parisi et al., 2014c, 2015b). For the reported experiments of this approach, action labels were recorded from speech so that action–word mappings of training samples can result from co-occurring audio-visual inputs using unsupervised learning and our strategy for label propagation. To evaluate our system, we compared newly obtained results with reported results using hierarchical GWR-based recognition (Parisi et al., 2015b). We conducted additional experiments with different percentages of available labeled samples during the training, ranging between 100% (all samples are labeled) and 0%.

### 5.5.1 Audiovisual Inputs

Our action dataset is composed of 10 full-body actions performed by 13 subjects (see Section 4.2). Videos were captured in a home-like environment with depth maps sampled at 30 frames per second. The dataset contains the following actions: *standing, walking, jogging, picking up, sitting, jumping, falling down, lying down, crawling,* and *standing up*. From raw depth map sequences, 3D body joints were estimated on the basis of the tracking skeleton model and actions were represented by three body centroids (Fig. 5.5): $C_1$ for upper body with respect to the shoulders and the torso; $C_2$ for middle body with respect to the torso and the hips; and $C_3$ for lower body with respect to the hips and the knees, with each centroid computed as a point sequence of real-world coordinates $C = (x, y, z)$. To attenuate sensor noise, we used the median value of the last 3 estimated points (yielding action features at

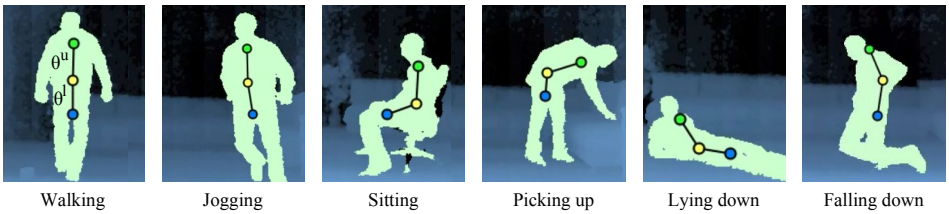



Figure 5.5: Representation of actions from the KT dataset. We estimate three centroids: $C_1$ (green), $C_2$ (yellow) and $C_3$ (blue) for upper, middle and lower body respectively. The segment slopes $\theta^u$ and $\theta^l$ describe the posture in terms of the overall orientation of the upper and lower body (Parisi et al., 2016b).

10 frames per second). We then estimated upper and lower orientations $\theta_u$ and $\theta_l$ given by the slope angles of the line segments $\{C_1, C_2\}$ and $\{C_2, C_3\}$ respectively. As shown in Fig. 5.5, the values $\theta_u$ and $\theta_l$ describe the overall body pose according to the orientation of the torso and the legs, which allows capturing significant body features such as the characteristic posture of actions. We computed the body velocity $S_i$ as the difference in pixels of the upper centroid $C_1$ between two consecutive frames.

For recording action labels, we used automatic speech recognition from Google's cloud-based ASR enhanced with domain-dependent post-processing (Twiefel et al., 2014). The post-processor translates each sentence in a list of candidate sentences returned by the ASR service into a string of phonemes. To exploit the quality of the well-trained acoustic models employed by this service, the ASR hypothesis is converted to a phonemic representation employing a grapheme-to-phoneme converter. The word from a list of in-domain words is then selected as the most likely word candidate. An advantage of this approach are the hard constraints of the results, as each possible result can be mapped to an expected action word. Reported experiments showed that the sentence list approach obtained the best performance for in-domain recognition with respect to other approaches on the TIMIT speech corpus[2] with a sentence-error-rate of 0.521. The audio recordings were performed by speaking the name of the action in a time window of 2 seconds during its execution, i.e. for each repetition in the case of *jumping, picking up, falling down*, and *standing up*, and every 2 seconds for cyclic actions (*standing, walking, jogging, sitting, lying down, crawling*). This approach has the advantage of assigning labels to continuous video streams without the manual segmentation of visual features from specific frames.

### 5.5.2 Results and Evaluation

For a consistent comparison with previous results, we adopted similar feature extraction and evaluation schemes. We divided the data equally into training and test set, i.e., 30 sequences of 10 seconds for each cyclic action (*standing, walking,*

[2]TIMIT Acoustic-Phonetic Continuous Speech Corpus: `https://catalog.ldc.upenn.edu/LDC93S1`

*jogging, sitting, lying down, crawling*) and 30 repetitions for each goal-oriented action (*jumping, picking up, falling down, standing up*). Both the training and the test sets contained data from all subjects.

For the learning in the $G^{STSm}$ layer, we used the following training parameters: insertion threshold $a_T = 0.9$, learning rates $\epsilon_b = 0.3$, $\epsilon_n = 0.006$, firing counter parameters $\tau_b = 0.3$, $\tau_i = 0.1$, $\kappa = 1.05$, maximum age for edges $\mu_{max} = 500$, labelling threshold $\pi_T = 0.5$ (OSS-GWR only). These parameters were empirically found with respect to best accuracy in terms of classification performance. Similar to Parisi et al. (2015b), each network was trained for 500 epochs over the entire training set. Once a layer was trained, its weights were set fixed and the next higher-level layer was trained. If we consider the 4 network layers of the architecture, it took overall 2000 epochs to obtain a trained neuron in the $G^{STSm}$ network. Layers $G^{STSm}$ and AWL were trained together according to Algorithm 1.

Experimental results showed an average classification accuracy of $93,3\%$, comparing with the state-of-the-art results of 94% reported by Parisi et al. (2015b) which required the manual segmentation of training samples for assigning ground-truth labels. The confusion matrices for the novel OSS-GWR and the S-GWR approaches tested on a set of 10 actions are shown in Fig. 5.6 and 5.7 respectively (with the rows of the matrix being the instances of actual actions and columns being the instances of predicted actions). The matrices show that there is a significant similarity on which samples were misclassified, suggesting that misclassification depends more on the visual features than on issues related to the associative learning mechanism via the co-occurrence of audiovisual inputs. For example, actions that are similar with respect to body posture (e.g. *walking* and *jogging*, *falling down* and *lying down*), tend to be mutually misclassified. The reason for this is that although the defined features used to learn relevant properties of actions should be sufficient to univocally describe spatiotemporal patterns over different timescales, tracking inaccuracies from the depth sensor may have a negative impact on the extraction of reliable pose-motion cues. While it is possible to embed the detection of sensor noise in low-level networks (Parisi et al., 2015b), it is non-trivial to detect inaccurate samples that belong to the feature space, e.g. caused by the (self-)occlusion of body joints. In this case, tracking errors will propagate from low to higher-level layers and lead to the misclassification of samples.

An additional experiment consisted of decreasing the percentage of labeled action samples. Since visual representations are progressively learned without supervision, we expect that the absence of training action labels will not have a catastrophic impact on the correct development of associative connections of audiovisual input (as would be expected for a strictly supervised method). For this purpose, we trained our system with a similar scheme as in the first experiment, but this time we omitted action words from ASR of randomly chosen samples and varied the percentage of available labels from 100% to 0%. Here, with *sample* we do not refer to a single data point (as in the experiment from Section 5.3.1), but rather to a set of data points represented by the number of frames for the duration of the audio time window, i.e. 20 frames. The average classification accuracy with different percentages of omitted audio samples for randomly selected samples over

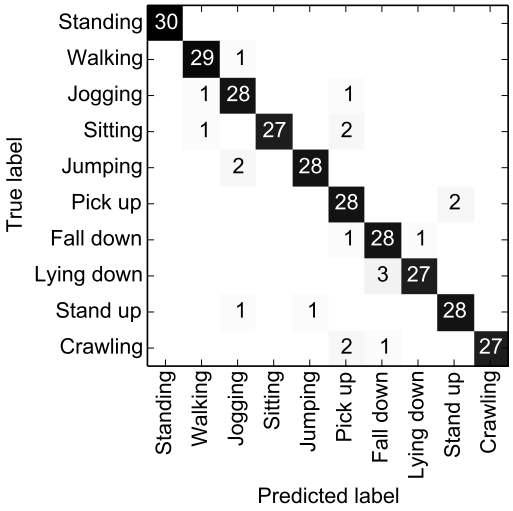



Figure 5.6: Confusion matrix for the OSS-GWR approach on the KT action dataset without manual segmentation (Parisi et al., 2016b).

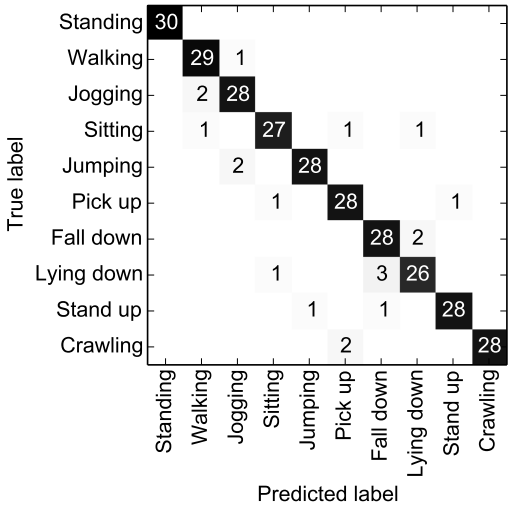



Figure 5.7: Confusion matrix for the S-GWR approach tested on the KT action dataset with samples manually segmented (Parisi et al., 2016b).

10 runs is displayed in Fig. 5.8. We can observe that although a decreasing number of available labeled samples during the training phase has a negative impact on the classification performance, this decline is not proportional to the number of omitted action words. As soon as 10% of labeled samples are available during the training, the system shows an accuracy of 58.5%, and accuracy values above 85% can be observed for 50% or more available labeled samples. On the other hand, we found that the timing at which these action words are presented to the AWL layer over the training epochs does have a significant impact on the performance. In fact, best results were obtained if action words are presented when visual representations have reached a certain degree of stability, while associative connections created at

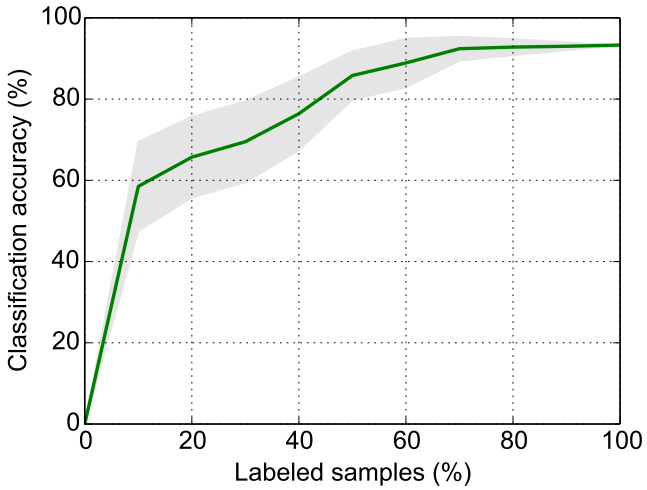



Figure 5.8: Average classification accuracy over 10 runs for a decreasing percentage of available action labels (Parisi et al., 2016b).

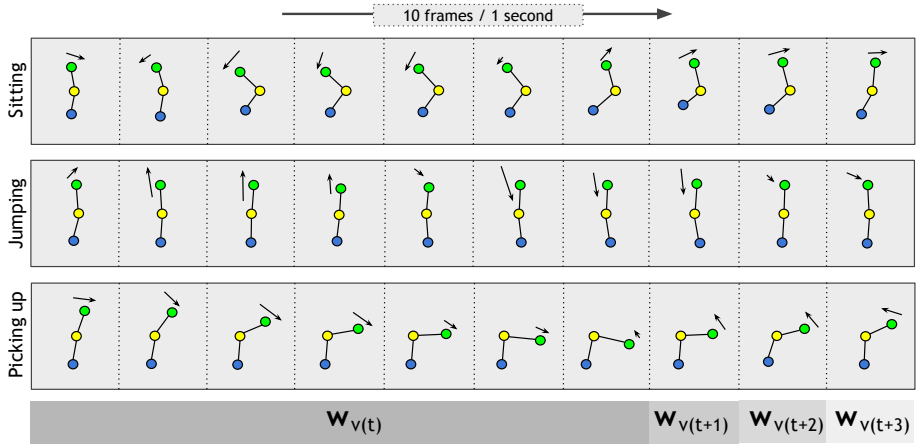



Figure 5.9: Example of learned visual representations generated from speech recognition for the action words *sitting*, *jumping*, and *picking up*. The figure shows the three body centroids and the motion intensity of the upper-body centroid (black arrow) for a window of 10 frames (1 second) starting from the action onset neuron (Parisi et al., 2016b).

early stages of visual development may not be as reliable.

To gain insight into how well the associative layer has learned action dynamics, we generated learned action representations from action words in the absence of visual input. The visualizations were generated from the recognized action words by computing onset neurons in $G^{STSm}$ via the associative connections from AWL (Eq. 5.5). For each onset neuron, one-step prediction was made using the temporal connectivity (Eq. 5.6) to compute snapshots of 10 frames (1 second of action). The visual representations of the actions *sitting*, *jumping*, and *picking up* for a time window of 1 second are shown in Fig. 5.9, where we displayed the

three body centroids and the motion intensity of the upper-body centroid (black arrow). From these visualizations, we can argue that the associative layer successfully learns temporally-ordered representations of visual input sequences from onset neurons, and therefore that our model accounts for the bidirectional retrieval of audiovisual input.

## 5.6 Summary

We presented a hierarchical neural architecture for learning multimodal action representations from a set of training audiovisual inputs. In particular, we investigated how associative links between unimodal representations can emerge in a self-organizing manner from the co-occurrence of multimodal stimuli. Visual generalizations of action sequences were learned using hierarchically-arranged GWR networks for the processing of inputs with increasingly larger temporal windows.

Multimodal action representations in terms of action–word mappings were obtained by incrementally developing bidirectional connections between learned visual representations and action labels from automatic speech recognition. For this purpose, we proposed an associative network with asymmetric inter-layer connectivity that takes into account the spatiotemporal dynamics of action samples and binds co-occurring audiovisual inputs. For this associative layer, we implemented an extended GWR learning algorithm (the OSS-GWR) that accounts for the propagation of action labels in a semi-supervised training scenario and that learns neural activation patterns in the temporal domain through enhanced synaptic connectivity. Experiments with a dataset of 10 full-body actions showed that our system achieves state-of-the-art classification performance without requiring the manual segmentation of training samples. Together, these results show that our neural architecture accounts for the bidirectional retrieval of audiovisual inputs, also in the scenario where a number of action labels is omitted during the training phase.

Our implementation of bidirectional action–word connections roughly resembles a phenomenon found in the human brain, i.e. spoken action words elicit receptive fields in the visual area (Barraclough et al., 2005; Miller and Saygin, 2013). In other words, learned visual representations of actions can be activated in the absence of visual input, in this case from recognized speech. These visualizations can be generated by computing the onset neuron in the $G^{STSm}$ layer via the developed associative connections to AWL, so that temporally-ordered action snapshots can be obtained from neural activation patterns learned by synaptic connectivity in the temporal domain. We have shown that this property can be used in practice to assess how well the model accounts for learning congruent visual representations of actions from pose-motion features.

Our results encourage the leverage of the proposed architecture in several directions. For instance, so far we have assumed that the training labels provided from speech are correct. On the other hand, several developmental studies have shown that human infants are able to learn action–word mappings also in the presence

of missing, ambiguous or sometimes contradictory referents using cross-situational statistics (Smith and Yu, 2008). Thus, it would be interesting to evaluate the robustness of the system if the available labels are sometimes inaccurate or in contradiction with previously learned labels. Furthermore, another limitation of our model is the use of domain-dependent ASR. Although this approach yields the reliable recognition of a set of action words (Twiefel et al., 2014), it has the disadvantage that a specific set of words has to be defined a priori. Therefore, new action words cannot be learned during the training process. We plan to address this constraint by accounting for learning new lexical features so that the action vocabulary can be dynamically extended during training sessions. It has been shown that lexical features can be learned using recursive self-organizing architectures (Strickert and Hammer, 2005), obtaining action word representations from a phonemic representation of recognized audio. This extension would comprise a hierarchical stream for processing audio features and, similar to the visual hierarchy, higher-lever representations of speech (words) would be learned from lower-level representations (e.g., phonemes). Such a processing scheme would be in line with neurophysiological evidence supporting the hierarchical processing of aural features in the auditory cortex with increasing temporal receptive windows (Lerner et al., 2011). By considering the aforementioned extensions, the mechanism responsible for developing associative connections should be robust to situations in which action words recognized from speech may not be reliable. Therefore, an additional labelling scheme should be considered that takes into account cross-statistical properties of labels to guarantee a congruent audiovisual mapping.

Finally, our results motivate the extension of our approach for scenarios that require more complex audiovisual inputs, for instance by considering the recognition of transitive actions. This challenging task would require accounting for the learning of action-object relations to be described by more flexible action words, e.g. labelling both the action and the object being used. An interesting question would then be how multiple different modules develop bidirectional connections in order to provide a congruent perceptual experience.

The manual labeling of training sequences is expensive and hinders the automatic, continuous learning of novel information. Thus, research in the direction of neurocognitive architectures aimed at developing robust multimodal representations from more natural interactions would provide a significant benefit for learning agents in order to trigger proper action-driven behavior in complex environments.

# Chapter 6

# Action Learning and Assessment with Recurrent Self-Organization

## 6.1 Introduction

The efficient processing of sequential input plays a crucial role in biological systems. An example of this is the mammalian visual cortex, a hierarchical brain structure able to efficiently compute incoming visual stimuli at different spatial and temporal scales (see Chapter 2). In combination with other brain areas, the capability of the visual cortex to efficiently compute spatiotemporal structure of the input results in highly skilled mechanisms of visual perception, e.g. the robust discrimination of complex biological motion patterns. Similarly, artificial systems processing sequential input from cluttered environments must account for the robust computation of the underlying spatiotemporal structure of incoming stimuli.

In Chapters 4 and 5, we demonstrated that hierarchies of self-organizing neural networks can learn spatiotemporal action features with increasing complexity of representation. The main advantage of this method over traditional supervised learning approaches is that visual representations are learned in an unsupervised fashion. However, since the conventional definition of self-organizing networks such as the SOM (Kohonen, 1990), the GNG (Fritzke, 1995) and the GWR (Marsland et al., 2002) do not account for the processing of time-varying inputs, the temporal processing of features in our neural architectures was explicitly modeled in terms of neurons in higher-level layers computing the concatenation of neural activation trajectories from lower-level layers, which increases the dimensionality of neural weights along the hierarchy. The high-dimensionality of neural weights may compromise the accuracy of the metric used to compute best-matching neurons, in our case the Euclidean distance, and therefore the ability of the network to develop correct topological maps. As discussed in Section 3.3, different temporal extensions of self-organizing networks have been proposed that implement recurrent connectivity so that neural activation in the map is driven by multiple time steps. In particular, context learning as proposed by Strickert and Hammer (2005) combines a compact back-reference with a weighted combination of the current input and

the previous network activation.

In this chapter, we propose an extension the GWR network with context learning for processing sequential input. In Section 6.2, we introduce a temporal GWR for learning body motion sequences and assessing the quality of novel sequences with respect to learned motion templates. Reported experiments show how this extension of the GWR outperforms previous temporal self-organizing models and how recurrent self-organization can be used to provide visual feedback in real time when incorrect body motion is detected. In Section 6.3, we propose a deep neural architecture with hierarchically-arranged recurrent GWR networks for learning action features with increasingly larger spatiotemporal receptive fields. Visual representations obtained through unsupervised learning are incrementally associated to symbolic action labels for the purpose of action classification. We show how this novel architecture outperforms previous self-organizing approaches for action recognition with the KT action dataset (see Section 4.2) and the Weizmann action benchmark, especially when the number of ground-truth labels available during the training is decreased.

## 6.2 Human Motion Assessment

The correct execution of well-defined movements plays a crucial role in physical rehabilitation and sports. While there is an extensive number of well-established approaches for human action recognition, the task of assessing the quality of actions and providing feedback for correcting inaccurate movements has remained an open issue in the literature (see Section 2.2).

In this section, we present a learning-based method for efficiently providing feedback on a set of training movements captured by a depth sensor. We propose a novel recursive neural network that uses growing self-organization for the efficient learning of body motion sequences. The quality of actions is then computed in terms of how much a performed movement matches the correct continuation of a learned sequence. The proposed system provides visual assistance to the person performing an exercise by displaying real-time feedback, thus enabling the user to correct inaccurate postures and motion intensity. We evaluate our approach with a data set containing 3 powerlifting exercises performed by 17 athletes using a temporal extension of both the SOM and the GWR networks. Experimental results show that our neural architecture accounts for computing visual feedback in real time, and that processing motion intensity is crucial for exercises with variations of speed or lockouts. We evaluated the detection of mistakes both on a single- and multiple-subject scenario.

For our approach, we tracked the position of a person based on a simplified 3D model of the human skeleton using a set of $K$ joint coordinates $\mathbf{j}_i = (x_{j_i}, y_{j_i}, z_{j_i}), 1 \leq i \leq K$, so that at each timestep $t$ the body posture is represented as the collection of $K$ joints $\mathbf{p}(t) = (\mathbf{j}_i(t), ..., \mathbf{j}_K(t))$. We computed motion intensity from posture sequences with the inter-frame difference between consecutive joint pairs. The Kinect's skeleton model (Fig. 6.1), although not faithful to human

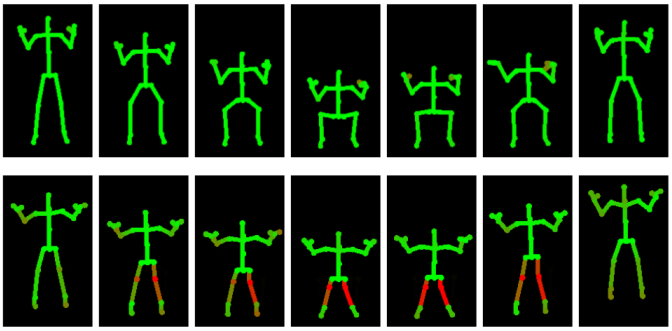


Figure 6.1: Visual feedback for correct squat sequence (top), and a sequence containing *knees in* mistake (bottom, joints and limbs in red) (Parisi et al., 2015a).

anatomy, provides reliable estimations of the joints' position over time. This allows us to extract significant properties of postural dynamics.

### 6.2.1 Proposed Architecture

Our architecture consists of two hierarchically arranged layers with self-organizing networks processing posture and motion sequences (Fig. 6.2). The first layer is composed of two GWR networks, $G^P$ and $G^M$, that learn a dictionary of posture and motion feature vectors respectively. This hierarchical scheme has the advantage of using a fixed set of learned features to compose more complex patterns in the second layer, where the recursive network $G^I$ is trained with sequences of posture-motion activation patterns from the first layer to learn the spatiotemporal structure of the input.

From a dataset $\mathbf{X}$ with $n$ samples, we compute the best-matching neuron of each input with respect to a trained network with $N$ neurons, so that a sequence of input activations from the training set is given by

$$\Omega(\mathbf{X}) = \{\mathbf{w}_{b(\mathbf{x}_1)}, \mathbf{w}_{b(\mathbf{x}_2)}, ..., \mathbf{w}_{b(\mathbf{x}_n)}\}, \tag{6.1}$$

with $b(\mathbf{x}_i) = \arg\min_{i \in N} \|\mathbf{x}_i - \mathbf{w}_j\|$ computing the index of the neuron (or prototype vector) that minimizes the distance to the current input. We denote the dataset of posture and motion vectors as $\mathbf{P}$ and $\mathbf{M}$ respectively. The training dataset for $G^I$, $\mathbf{I}$, is given by the horizontal concatenation of the set of activations over $\mathbf{P}$ and $\mathbf{M}$, i.e. $\mathbf{I} = \{\Omega(\mathbf{P}) \frown \Omega(\mathbf{M})\}$.

### 6.2.2 Merge-GWR

To learn the spatiotemporal structure of the input in $G^I$, we extend the traditional GWR algorithm (Marsland et al., 2002) for efficient context learning (Strickert and Hammer, 2005). We adopt the distance function as defined by Eq. 3.18 and 3.19 such that

$$d_n(t) = \alpha \cdot \|\mathbf{x}(t) - \mathbf{w}_n\|^2 + (1 - \alpha) \cdot \|[\beta \cdot \mathbf{w}_{\hat{b}} + (1 - \beta) \cdot \mathbf{c}_{\hat{b}}] - \mathbf{c}_i\|^2, \tag{6.2}$$

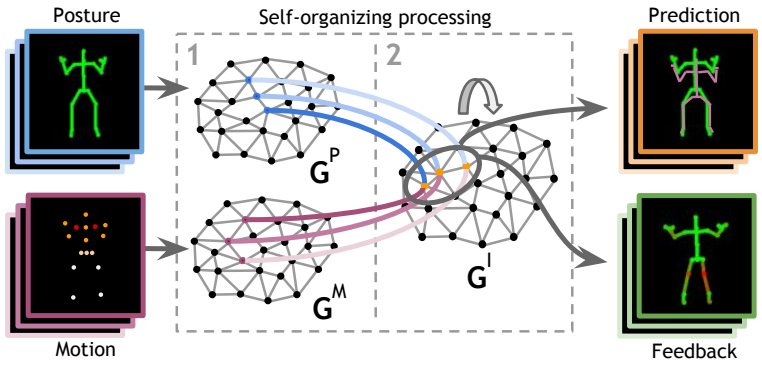



Figure 6.2: Multilayer learning architecture with incremental self-organizing networks. In Layer 1, two GWR networks learn posture and motion features respectively. In Layer 2, a recursive GWR learns spatiotemporal dynamics of body motion. This mechanism allows to predict the ideal continuation of a learned sequence and compute feedback as the difference between its expected behavior and its current execution (Parisi et al., 2016a).

where $\alpha$ and $\beta$ are constant values that modulate the influence of the current input and the past, and $\hat{b}$ is the index of the winner neuron at the previous time step. Specifically for our recursive GWR model, the update functions of the weight and context neurons become

$$\Delta \mathbf{w}_i = \epsilon_i \cdot \eta_i \cdot (\mathbf{x}(t) - \mathbf{w}_i), \tag{6.3}$$

$$\Delta \mathbf{c}_i = \epsilon_i \cdot \eta_i \cdot ([\beta \cdot \mathbf{w}_{\hat{b}} + (1 - \beta) \cdot \mathbf{c}_{\hat{b}}] - \mathbf{c}_i), \tag{6.4}$$

where $\epsilon_i$ is the learning rate and $\eta_i$ is the firing counter. The complete training algorithm is illustrated in Algorithm 2.

The recursive GWR architecture avoids the drawback of our previous approach using a MSOM, where the number of neurons of the networks had to be decided a priori. Furthermore, since the GWR does not have a fixed lattice topology, it can better represent the feature space.

## Time Series Analysis

We compared the performance of our MGWR on a time series analysis task with other two well-established models of recursive self-organization: Merge Neural Gas (MNG) (Strickert and Hammer, 2005) and Merge Growing Neural Gas (MGNG) (Andreakis et al., 2009). For the analysis, we used the Mackey Glass time series, a continuous and chaotic function that has been used to evaluate the temporal quantization of recursive models. It is defined by the differential equation $\frac{dx}{d\tau} = bx(\tau) + \frac{ax(\tau-d)}{1+x(\tau-d)^{10}}$ and depending on the values of the parameters, it displays a range of pseudo-periodic dynamics. For evaluation purposes, it is generally used with $a = 0.2$, $b = -0.1$, and $d = 17$. Similar to previous comparison schemes in the literature (Voegtlin, 2002), all the models were evaluated in terms of their

**Algorithm 2** Merge-GWR.

1: Create two random neurons $A = \{\mathbf{w}_1, \mathbf{w}_2\}$ with context vectors $\mathbf{c}_1$, $\mathbf{c}_2$
2: Initialize an empty set of connections $E = \emptyset$
3: Initialize an empty global context $\mathbf{C}_0 = 0$
4: At each iteration, generate an input sample $\mathbf{x}_t$.
5: Select the best and second-best matching neurons (Eq. 6.2):
$b = \arg\min_{i \in A} d_i(t)$, $s = \arg\min_{i \in A/\{b\}} d_i(t)$.
6: Update global context $\mathbf{C}_t = \beta \cdot \mathbf{w}_{\hat{b}} + (1 - \beta) \cdot \mathbf{c}_{\hat{b}}$
7: Create a connection $E = E \cup \{(b, s)\}$ if it does not exist and set its age to 0.
8: If $(\exp(-d_b(t)) < a_T)$ and $(\eta_b < f_T)$ then:
Add a new node $r$ $(A = A \cup \{r\})$:
$\mathbf{w}_r = 0.5 \cdot (\mathbf{w}_b + \mathbf{x}_t)$, $\mathbf{c}_r = 0.5 \cdot (\mathbf{C}_t + \mathbf{x}_t)$, $\eta_r = 1$,
Update edges between neurons:
$E = E \cup \{(r, b), (r, s)\}$ and $E = E/\{(b, s)\}$.
9: If no new node is added, update weight and context of the winning node and its neighbors $i$:
$\Delta\mathbf{w}_b = \epsilon_b \cdot \eta_b \cdot (\mathbf{x}_t - \mathbf{w}_s)$, $\Delta\mathbf{w}_i = \epsilon_n \cdot \eta_i \cdot (\mathbf{x}_t - \mathbf{w}_i)$,
$\Delta\mathbf{c}_b = \epsilon_b \cdot \eta_b \cdot (\mathbf{C}_t - \mathbf{c}_b)$, $\Delta\mathbf{c}_i = \epsilon_i \cdot \eta_i \cdot (\mathbf{C}_t - \mathbf{c}_i)$.
10: Increment the age of all edges connected to $b$ of 1.
11: Reduce the firing counters of the best-matching neuron and its neighbors $i$:
$\eta_b = \eta_b + (\tau_b \cdot \kappa \cdot (1 - \eta_b) - \tau_b)$, $\eta_i = \eta_i + (\tau_i \cdot \kappa \cdot (1 - \eta_i) - \tau_i)$,
with $\tau$, $\kappa$ constants controlling the curve behavior.
12: Remove all edges with ages larger than $\mu_{max}$ and remove nodes without edges.
13: If the stop criterion is not met, repeat from step 4.

temporal quantization error (TQE) for 30 steps in the past with 150,000 elements of the series. The TQE for the map at time $t$ is defined as:

$$e(t) = \sum_{i=1}^{N} \left( \sum_{j:I(j)=i} \|\mathbf{x}^{j-t} - \sum_{j:I(j)=i} \mathbf{x}^{j-t}/\gamma_i\|^2/\gamma_i \right)^{1/2} /N, \tag{6.5}$$

where $N$ is the number of neurons, $\gamma_i$ is the number of timesteps in which neuron $i$ becomes the winner.

For MGWR learning, we used the following training parameters: insertion threshold $a_T = 0.95$, learning rates $\epsilon_b = 0.01$, and $\epsilon_n = 0.001$, maximum age $a_{max} = 200$, firing counter parameters $\tau_b = 0.3$, $\tau_i = 0.1$, $\kappa = 1.05$, firing threshold $\eta_T = 0.1$, and context learning parameters $\alpha = 0.6$, $\beta = 0.7$ with 100 training epochs. The training parameters of MNG and MGNG were set according to previously reported experiments (Strickert and Hammer, 2005; Andreakis et al., 2009).

The TQE for the recursive models MSOM, MNG, MGNG and our MGWR is reported in Fig. 6.3, showing how the four models behave quite similar, with the MGWR slightly outperforming the others. The average TQE over 30 timesteps was MSOM= 0.0795, MNG= 0.0749, MGNG= 0.0721, and MGWR= 0.0697.

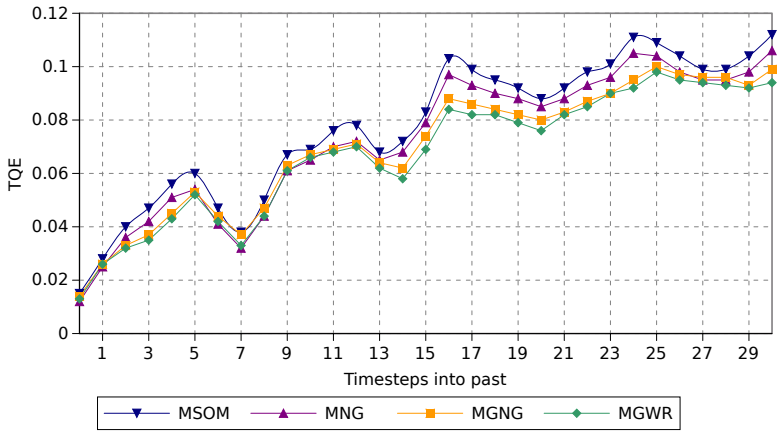



Figure 6.3: Temporal quantization error over 30 timesteps into past for the Mackey-Glass time series (Parisi et al., 2016a).

Although both the MSOM and MNG are not growing methods, the latter performs better since the topology of the MNG network is not fixed, thus yielding a smaller quantization error.

### 6.2.3 Feedback from Prediction

The underlying idea for assessing the quality of a sequence is to measure how much the current input sequence differs from a learned template. In other words, provided that the trained model $G^I$ is able to predict a training sequence with a satisfactory degree of accuracy, it is then possible to quantitatively compute how much a novel sequence follows this expected pattern.

We define a function that computes the difference of a current input sequence, $\Omega_t$ , from its expected input, i.e. the prediction of the next element of the sequence given $\Omega_{t-1}$:

$$\mathfrak{f}_\Omega(t) = \|\Omega_t - \mathfrak{p}(\Omega_{t-1})\|, \tag{6.6}$$

$$\mathfrak{p}(\Omega_{t-1}) = \mathbf{w}_p \text{ with } p = \arg\min_{j \in N} \|\mathbf{c}_j - \Omega_{t-1}\|. \tag{6.7}$$

Since the weight and context vectors of the prototype neurons lie in the same feature space as the input ($\mathbf{w}_i, \mathbf{c}_i \in \mathbb{R}^{|\Omega|}$), it is possible to provide joint-wise feedback computations. The recursive prediction function $\mathfrak{p}$ can be applied an arbitrary number of timesteps into the future. Therefore, after the training phase is completed, it is possible to compute $\mathfrak{f}_\Omega(t)$ in real time with linear computational complexity $\mathcal{O}(|A|)$, which depends on the number of neurons of a trained model.

We show the result of this prediction mechanism in Fig. 6.4. For this example, a network was trained with the *Finger to nose* routine, which consists of keeping your arm bent at the elbow and then touching your nose with the tip of your finger. When the person starts performing the routine after this training phase, we can see progressively fading violet lines representing the next 30 time

Figure 6.4: Movement prediction – Visual hints for future steps of a network trained for *Finger to nose* routine. Progressively fading violet lines indicate the correct order of execution (Parisi et al., 2015a).

steps, thereby providing visual assistance to successfully carry out the movement through spatiotemporal hints. The value 30 was empirically determined to provide a substantial reference to future steps while limiting visual clutter.

To compute feedback, we use the predictions estimated by $\mathfrak{p}$ as hints on how to perform a routine over 100 timesteps into the future, and then use $\mathfrak{f}_\Omega(t)$ to spot mistakes on novel sequences that do not follow the expected pattern for individual joint pairs. A mistake can then be detected when $\mathfrak{f}_\Omega(t)$ exceeds a given threshold $\mathfrak{f}^T$ over $i$ timesteps. Visual representations of these computations can then provide useful qualitative feedback to assist the user on the correct performance of the routine and the correction of mistakes (Fig. 6.1). Different from our previous model, our current approach learns also motion intensity to better detect temporal discrepancies. Therefore, it is possible to provide more accurate feedback on posture transitions and the correct execution of lockouts.

### 6.2.4 Experimental Results

We present our experimental results on a data-set of 3 powerlifting movements used for the training, validation, and test of the proposed system.

#### Powerlifting Dataset

The data collection took place at the Kinesiology Institute of the University of Hamburg, Germany, where 17 volunteering participants (9 male, 8 female) performed 3 different powerlifting exercises:

**E1)** *High bar back squat*: One repetition consists of crouching with a loaded barbell behind the back until the hips are lower than the knees and then standing up;

**E2)** *Deadlift*: Lift a loaded barbell off the ground to the hips, then lower back to the ground;

**E3)** *Dumbbell lateral raise*: Start with the arms at the side of the body then raise the dumbbells sidewards while keeping the elbows higher than the wrists.

For a thorough evaluation of our system, we also recorded a set of typical mistakes for each routine:

**E1)** M1) *Good morning*: Raising the hips without raising the chest with an excessively horizontal back angle;

M2) *Half squat*: Going only halfway down to the ground;

M3) *Knees in*: Bow the knees toward each other during the lift.

**E2)** M1) *No lockout*: The execution is carried out properly, but the lift is stopped before the lockout;

M2) *Rounded back*: The back is heavily rounded during the lift instead of being in a straight line.

**E3)** M1) *Low elbows*: Lateral lifts performed with the wrists being higher than the elbows.

We captured body motion of correct and incorrect executions with a Kinect v2 sensor[1] and estimated body joints using Kinect SDK 2.0 that provides a set of 25 joint coordinates at 30 frames per second. The participants executed the routines frontal to the sensor placed at 1 meter from the ground. We processed video sequences with Kinect SDK to segment body motion and extract 3D joint coordinates frame by frame.

We used the joints for *head, neck, wrists, elbows, shoulders, spine, hips, knees*, and *ankles*, for a total of 13 3D-joints (39 dimensions). We manually segmented single repetitions for all exercises. In order to obtain translation invariance, we subtracted the *spine_base* joint (the center of the hips) from all the joints in absolute coordinates.

## Evaluation

We evaluated our method for computing feedback with individual and multiple subjects. We divided the correct body motion data with 3-fold cross-validation into training and test sets and trained the models with data containing correct motion sequences. Each network was trained for 100 epochs. For the test phase, both the correct and incorrect movements were used with feedback threshold $\mathfrak{f}^T = 0.7$ over 100 frames.

Our expectation was that the output of the feedback function would be higher for sequences containing mistakes. We observed true positives (TP), false negatives (FN), true negatives (TN), and false positives (FP) as well as the measures true positive rate (TPR or sensitivity), true negative rate (TPR or specificity), and positive predictive value (PPV or precision). Results for single- and multiple-subject data on E1, E2, and E3 routines are displayed in Table 6.1 and 6.2 respectively, along with a comparison with the best-performing feedback function $\mathfrak{f}_b$ from Parisi et al. (2015a) in which only posture sequences were used.

[1]Microsoft Kinect 2.0 – `microsoft.com/en-us/kinectforwindows/develop/`

Table 6.1: Single-subject evaluation.

| | | TP | FN | TN | FP | TPR | TNR | PPV |
|---|---|---|---|---|---|---|---|---|
| E1 | $\mathfrak{f}_\mathfrak{b}$ | 35 | 10 | 33 | 0 | 0.77 | 1 | 1 |
| | $\mathfrak{f}_\Omega$ | 35 | 2 | 41 | 0 | 0.97 | 1 | 1 |
| E2 | $\mathfrak{f}_\mathfrak{b}$ | 24 | 0 | 20 | 0 | 1 | 1 | 1 |
| | $\mathfrak{f}_\Omega$ | 24 | 0 | 20 | 0 | 1 | 1 | 1 |
| E3 | $\mathfrak{f}_\mathfrak{b}$ | 63 | 0 | 26 | 0 | 1 | 1 | 1 |
| | $\mathfrak{f}_\Omega$ | 63 | 0 | 26 | 0 | 1 | 1 | 1 |

Table 6.2: Multi-subject evaluation.

| | | TP | FN | TN | FP | TPR | TNR | PPV |
|---|---|---|---|---|---|---|---|---|
| E1 | $\mathfrak{f}_\mathfrak{b}$ | 326 | 1 | 7 | 151 | 0.99 | 0.04 | 0.68 |
| | $\mathfrak{f}_\Omega$ | 328 | 1 | 13 | 143 | 0.99 | **0.08** | **0.70** |
| E2 | $\mathfrak{f}_\mathfrak{b}$ | 127 | 2 | 0 | 121 | 0.98 | 0 | 0.51 |
| | $\mathfrak{f}_\Omega$ | 139 | 0 | 0 | 111 | **1** | 0 | **0.56** |
| E3 | $\mathfrak{f}_\mathfrak{b}$ | 123 | 0 | 8 | 41 | 1 | 0.16 | 0.75 |
| | $\mathfrak{f}_\Omega$ | 126 | 0 | 15 | 31 | 1 | **0.33** | **0.80** |

The evaluation on single subjects shows that the system successfully provides feedback on posture errors with high accuracy. A drawback of our previous model was a limited memory due to the number of neurons being fixed a priori and a fixed network topology yielding a higher quantization error. In our current approach, the MGWR networks grow dynamically to better represent the spatiotemporal structure of the sequences. This allows us to reduce the temporal quantization error over longer timesteps (Fig. 6.3), so that more accurate feedback can be computed and thus reduce the number of false negatives and false positives (Table 6.1 and 6.2). Furthermore, since the networks can create new neurons according to the distribution of the input, each network can learn a larger number of possible executions of the same routine, thus being more suitable for training sessions with multiple subjects.

Tests with multiple-subject data show a significantly decreased performance, mostly due to a large number of false positives. This is not necessarily a flaw linked to the learning mechanism, but rather a consequence of the fact that people have different body configurations and, therefore, different ways to perform the same routine. A solution to attenuate this issue is to set different values for the feedback threshold $\mathfrak{f}^T$. For larger values, the system would tolerate more variance in the performance. On the other hand, one must consider whether a higher degree

of variance is desirable based on the application domain; for instance, rehabilitation routines may be tailored to a specific subject based on their specific body configuration and health condition.

Our experimental results encourage further work in the direction of embedding our system into an assistive robot which could interact with the user and motivate the correct performance of physical rehabilitation routines and sports training. This is supported by a number of studies in which robots were used for motivating the users to perform a set of health-related tasks (Dautenhahn, 1999; Kidd and Breazeal, 2007; Nalin et al., 2012). Furthermore, the assessment of motion plays a crucial role not only for the detection of mistakes on training sequences but also in the timely recognition of gait deterioration, e.g. linked to age-related cognitive declines. In this context, growing learning architectures are particularly suitable for this task, since they may adapt to the user through longer periods of time while still detecting significant changes in their motor skills.

## 6.3 Deep Self-Organizing Learning

### 6.3.1 Introduction

Computational models inspired by the hierarchical organization of the visual cortex have become increasingly popular for action recognition from videos, with deep neural network architectures producing state-of-the-art results on a set of benchmark datasets (see Chapter 2). Typically, visual models using deep learning comprise a set of convolution and pooling layers trained in a hierarchical fashion for yielding action feature representations with increasing degree of abstraction (Guo et al., 2016). This processing scheme is in agreement with neurophysiological studies supporting the presence of functional hierarchies with increasingly large spatial and temporal receptive fields along cortical pathways (see Section 2.2).

The training of deep learning models for action sequences has been proven to be computationally expensive and require an adequately large number of training samples for the successful learning of spatiotemporal filters. The supervised training procedure comprises two stages: a forward stage in which the input is represented with the current network parameters and the prediction error is used to compute the loss cost from ground-truth sample labels, and a backward stage which computes the gradients of the parameters and updates them using back-propagation through time (BPTT, Mozer 1995). While different regularization methods have been proposed to boost performance such as parameter sharing and dropout, on the other hand, the training process requires samples to be (correctly) labeled in terms of input-output pairs. Consequently, the question arises whether traditional deep learning models for action recognition can account for real-world learning scenarios, in which the number of training samples may not be sufficiently high and ground-truth labels may be occasionally missing or incorrect.

In Chapter 4 and 5, we showed that a deep architecture comprising a hierarchy of self-organizing networks can learn spatiotemporal action features with increasing

complexity of representation. The main advantage of this method over traditional supervised learning approaches is that visual representations are learned in an unsupervised fashion. For the purpose of classification, associative connections between these visual representations and symbolic labels are learned during the training phase. Remarkably, correct action–label mappings with state-of-the-art accuracy can be obtained also in the absence of a large percentage of ground-truth labels. On the other hand, experiments were conducted by feeding this self-organizing architecture with a set of hand-crafted action features, which goes against the idea of deep convolutional neural network architectures of learning significant action features by iteratively tuning internal representations. Furthermore, the temporal processing of features was modeled in terms of neurons in higher-level layers computing the concatenation of neural activation trajectories from lower-level layers, which increases the dimensionality of neural weights along the hierarchy. In this section, we address these two issues through the development of a self-organizing hierarchy with increasingly large spatiotemporal receptive fields in the spirit of deep learning with convolutional neural networks.

Similar to our hierarchical architectures presented in Chapter 4 and 5, the deep self-organizing architecture introduced here is composed of two distinct processing streams for pose and motion features, in correspondence to the ventral and the dorsal pathways respectively, and their subsequent integration in the STS area (see Section 2.1.2). We propose a novel temporal extension of the GWR equipped with recurrent connectivity, referred to as Gamma-GWR, that learns spatiotemporal properties of the input. Different from previously introduced models that learn action representations from hand-crafted 3D features, we use a hierarchy of Gamma-GWR networks to learn prototype action segments from video containing full-body silhouettes. For the purpose of classification, associative connections between visual action representations and action class labels are learned during the training phase. We evaluate our approach on the KT action dataset and the Weizmann action benchmark, showing that our model learns robust action-label mappings also in the case of occasionally absent or incorrect action class labels during training sessions.

### 6.3.2 Proposed Architecture

The proposed architecture is illustrated in Fig. 6.5. Each layer in the hierarchy comprises a Gamma-GWR network and a pooling mechanism for learning action features with increasingly large spatiotemporal receptive fields. In the last layer, we extend the Gamma-GWR to learn associative connections between visual representations and symbolic labels for the purpose of action classification.

#### Gamma-GWR

We introduce a temporal GWR network that equips each neuron with an arbitrary number of context descriptors to increase the memory depth and temporal resolution in the spirit of a Gamma memory model (de Vries and Príncipe, 1992). A

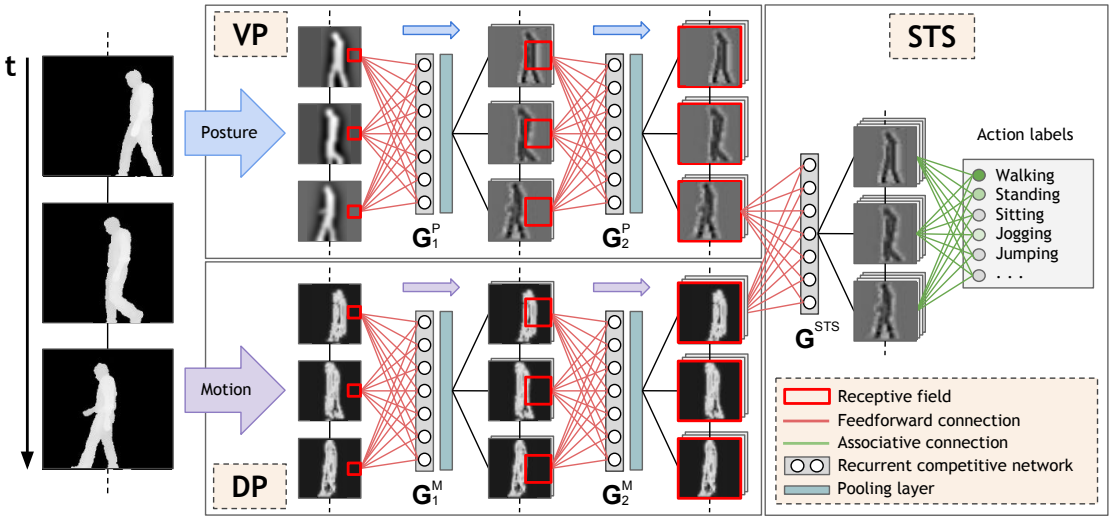



Figure 6.5: Diagram of our deep neural architecture with recurrent GWR networks for action recognition. Posture and motion action cues are processed separately in the ventral (VP) and the dorsal pathway (DP) respectively. At the STS stage, the recurrent network learns associative connections between prototype action representations and symbolic labels.

similar approach has been previously applied to SOM and GNG learning, showing good results in nonlinear time series analysis (Estévez and Hernández, 2011; Estévez and Vergara, 2012).

Following previous formulations of context learning (see Section 3.3), the activation of the network with a K-order Gamma memory becomes

$$d_i(t) = \alpha_w \cdot \|\mathbf{x}_t - \mathbf{w}_i\|^2 + \sum_{k=1}^{K} \alpha_k \cdot \|(\beta \cdot \mathbf{c}_k^{I_{t-1}} + (1-\beta) \cdot \mathbf{c}_{k-1}^{I_{t-1}}) - \mathbf{c}_k^i\|^2, \qquad (6.8)$$

for each $k = 1, ..., K$, where $\alpha, \beta \in (0; 1)$ are constant values that modulate the influence of the current input and the past, and $\mathbf{c}_0^{I_{t-1}} \equiv \mathbf{w}^{I_{t-1}}$ with random $\mathbf{c}_k^{I_0}$ at $t = 0$. It has been shown that the mean memory depth is $D = K/(1-\beta)$ and its temporal resolution is $R = 1 - \beta$ (Estévez and Vergara, 2012). Therefore, both depth and resolution are modulated by the value of $\beta$.

The proposed training algorithm is illustrated by Algorithm 3 (except for Steps 3, 9.c, and 10.b that are implemented by the AG-GWR only). Different from the standard GWR with activation function $a(t) = \exp(-\|\mathbf{x}(t) - \mathbf{w}_b\|)$ (see Section 3.2), in the Gamma-GWR this function is replaced with $a_t = \exp(-d_i(t))$ with $d_i(t)$ as defined by Eq. 6.8.

For $K = 1$, the Gamma-GWR is reduced to the Merge-GWR as described in Section 6.2.2.

**Algorithm 3** Associative Gamma-GWR (AG-GWR).

1: Start with a set of two random nodes, $A = \{\mathbf{w}_1, \mathbf{w}_2\}$ with context vectors $\mathbf{c}_k^i$ for $k = 1, ..., K, i = 1, 2$.
2: Initialize an empty set of connections $E = \emptyset$.
3: [AG-GWR only] Initialize an empty label matrix $H = \emptyset$.
4: Initialize $K$ empty global contexts $\mathbf{C}_k = 0$.
5: At each iteration, generate an input sample $\mathbf{x}_t$ with label $\xi$.
6: Select the best and second-best matching neurons (Eq. 6.8):
$b = \arg\min_{i \in A} d_i(t)$, $s = \arg\min_{i \in A/\{b\}} d_i(t)$.
7: Update context descriptors: $\mathbf{C}_k(t) = \beta \cdot \mathbf{c}_k^{I_{t-1}} + (1 - \beta) \cdot \mathbf{c}_{k-1}^{I_{t-1}}$.
8: Create a connection $E = E \cup \{(b, s)\}$ if it does not exist and set its age to 0.
9: If $(\exp(-d_b(t)) < a_T)$ and $(\eta_b < f_T)$ then:
a: Add a new node $r$ ($A = A \cup \{r\}$):
$\mathbf{w}_r = 0.5 \cdot (\mathbf{w}_b + \mathbf{x}_t)$, $\mathbf{c}_k^r = 0.5 \cdot (\mathbf{C}_k(t) + \mathbf{c}_k^i)$, $\eta_r = 1$,
b: Update edges between neurons:
$E = E \cup \{(r, b), (r, s)\}$ and $E = E/\{(b, s)\}$.
c: [AG-GWR only] Associate the sample label $\xi$ to the neuron $r$:
$H(r, \xi) = 1$, $H(r, l) = 0$, with $l \in L/\{\xi\}$.
10: If no new node is added:
a: Update weight and context of the winning node and its neighbors $i$:
$\Delta\mathbf{w}_b = \epsilon_b \cdot \eta_b \cdot (\mathbf{x}_t - \mathbf{w}_b)$, $\Delta\mathbf{w}_i = \epsilon_n \cdot \eta_i \cdot (\mathbf{x}_t - \mathbf{w}_i)$,
$\Delta\mathbf{c}_k^b = \epsilon_b \cdot \eta_b \cdot (\mathbf{C}_k(t) - \mathbf{c}_k^b)$, $\Delta\mathbf{c}_k^i = \epsilon_n \cdot \eta_i \cdot (\mathbf{C}_k(t) - \mathbf{c}_k^i)$.
b: [AG-GWR only] Update label values of $b$ according to the sample label $\xi$:
$H(b, \xi) = H(b, \xi) + 1$, $H(b, l) = H(b, l) - 0.1$, with $l \in L/\{\xi\}$.
11: Increment the age of all edges connected to $b$ of 1.
12: Reduce the firing counters of the best-matching neuron and its neighbors $i$:
$\eta_b = \eta_b + (\tau_b \cdot \kappa \cdot (1 - \eta_b) - \tau_b)$,
$\eta_i = \eta_i + (\tau_i \cdot \kappa \cdot (1 - \eta_i) - \tau_i)$,
with $\tau$, $\kappa$ constants controlling the curve behavior.
13: Remove all edges with ages larger than $\mu_{max}$ and remove nodes without edges.
14: If the stop criterion is not met, repeat from step 5.

### Pooling Layers

Along the hierarchical visual pathways, individual neurons specialize to increasingly complex stimulus features, thus yielding invariance to stimulus transformations such as changes in scale and position (Földiák, 1991). Early studies by (Hubel and Wiesel, 1962) postulated the idea of complex cell invariance with respect to spatial phase shifts carried out in terms of a linear summation of responses from phase-sensitive neurons.

Typically, computational models with deep architectures obtain invariant responses by alternating layers of feature detectors and nonlinear pooling neurons using the maximum (MAX) operation, which has been shown to achieve higher fea-

ture specificity and more robust invariance with respect to linear summation (Guo et al., 2016). The process of pooling became a standard operation in convolutional neural network models, where pooling layers generally follow convolutional layers with the aim to reduce the dimensions of the feature maps and network parameters, resulting in translation-invariant responses to features of previous layers (see Guo et al. (2016) for a review). Although robust invariance to translation has been obtained via MAX and average pooling, the MAX operator has shown faster convergence and improved generalization (Scherer et al., 2010).

Hosoya and Hyvärinen (2016) demonstrated that spatial pooling of visual cells can be implemented via linear transformations such as the principal components analysis (PCA) that retain several of the first principal components and ignore the remaining ones. PCA-pooling learns to suppress fine-grained structures of the input and thus extrapolates linear pooling of highly correlated parts of the stimulus. Therefore, it is argued that linear transformations of dimensionality reduction explain more neurally-plausible properties of the primary visual cortex (V1) complex cells, and that this process may also represent the basis of spatial pooling in higher visual areas.

In our architecture, the pooling layers are implemented via PCA over the multi-dimensional neuron weights, which maximizes the variance of projection along each component and thus minimizes the reconstruction error. The mathematical formulation of the PCA transformation is defined by a set of $k$ weight vectors $\mathbf{w}_{(k)}$ that map each row vector $\mathbf{x}_{(i)}$ of the input space to a new vector of principal components $\mathbf{t}_{(i)} = (t_1, ..., t_k)_{(i)}$ given by $t_{k(i)} = \mathbf{x}_{(i)} \cdot \mathbf{w}_{(k)}$ (Jolliffe, 2002). In our pooling layers, we compute the first component such that:

$$\mathbf{w}_{(1)} = \arg\max_{\|\mathbf{w}\|=1} \left\{ \sum_i (\mathbf{x}_{(i)} \cdot \mathbf{w})^2 \right\}, \tag{6.9}$$

where $\mathbf{x}_{(i)}$ are the neural weights activated by the input.

### Associative Learning and Classification

The aim of classification is to predict action labels from unseen action samples. For this purpose, the last network is equipped with an associative learning mechanism to map sample labels to prototype neurons representing action segments.

During the training phase, neurons in the $\text{G}^{\text{STS}}$ network can be assigned a label $l$ (with $l$ from a set $L$ of label classes) or remain unlabeled. The AG-GWR training algorithm for this network is illustrated by Algorithm 3. The associative matrix $H$ stores the frequency-based distribution of sample labels for each neuron in the network. Neurons in the $\text{G}^{\text{STS}}$ network are activated by the latest $K + 1$ input samples, i.e. from time $t$ to $t - K$. The label that we take into account is the one of the latest input at time $t$. When a new neuron $r$ is created and provided that $\xi$ is the label of the input sample $\mathbf{x}_t$, the associative matrix is updated according to $H(r, \xi) = 1$ and $H(r, l) = 0$, with $l \in L/\{\xi\}$. Instead, when an existing neuron $b$ is updated, we increase $H(b, \xi)$ by a value of 1 and decrease $H(b, l)$ of 0.1.

This labeling mechanism yields neurons associated to most frequent labels, thus also handling situations in which sample labels may be occasionally missing or incorrect. To predict the label $\lambda$ of a novel sample $\tilde{\mathbf{x}}_t$ after the training is completed, we return the label class with the highest value of the associative matrix for the best-matching neuron $b$ of $\tilde{\mathbf{x}}_t$ according to Eq. 6.8.

### 6.3.3 Experiments and Evaluation

We conducted experiments on two datasets: the KT action dataset and the Weizmann action benchmark (introduced in detail in the following sections). For these two datasets, we used the same action features from video frames containing segmented body silhouettes as shown in Fig. 6.6.a, and similar neural network training parameters.

#### Action Features

As input for our learning architecture, we use the difference of Gaussians (DoG) transformed versions of gray-level images containing segmented body silhouettes. This transformation emulates the preprocessing of visual input by the retina and the lateral geniculate nucleus (LGN) by applying an edge detector filter that approximates the sum of biologically-motivated Gabor filters (Lücke, 2009).

The DoG image is computed for each original image subtracting images convolved with Gaussians of different variances $\sigma_i^2$. The DoG operation with two Gaussian kernels $\sigma_1$ and $\sigma_2$ is defined as:

$$DoG = G_{\sigma_1} - G_{\sigma_2} = \frac{1}{\sqrt{2\pi}} \left( \frac{1}{\sigma_1} \exp^{-(x^2+y2)/2\sigma_1^2} - \frac{1}{\sigma_2} \exp^{-(x^2+y2)/2\sigma_2^2} \right), \qquad (6.10)$$

where $(x, y)$ are the pixels from the original input image. For our approach, we subtract the image convolved with two Gaussian kernels $\sigma_1^2 = 1$ and $\sigma_2^2 = 3$, with these values being consistent with the biologically measured ratio of $\frac{\sigma_1}{\sigma_2} \approx \frac{1}{3}$ from the cat's visual cortical simple cells (Somers et al., 1995). The resulting image sequence is illustrated in Fig. 6.6.b. Motion sequences are obtained as the pixel-level difference between consecutive transformed images containing body silhouettes.

#### Training Parameters

For our experiments, we used sequences of actions at 10 frames per second with resized images of $30 \times 30$ pixels containing only the segmented body silhouette. In the first layer, we select overlapping image patches of $3 \times 3$ pixels for both posture and motion sequences, i.e. a total of 784 patches for each $30 \times 30$ input image. The patches from posture and motion images are fed into the recurrent networks $\mathrm{G}_1^{\mathrm{P}}$ and $\mathrm{G}_1^{\mathrm{M}}$ respectively, both comprising a Gamma-GWR with 1 context descriptor ($K = 1$). In the second layer, the input is represented by the pooled activation from the first two networks, yielding a $28 \times 28$ representation for each

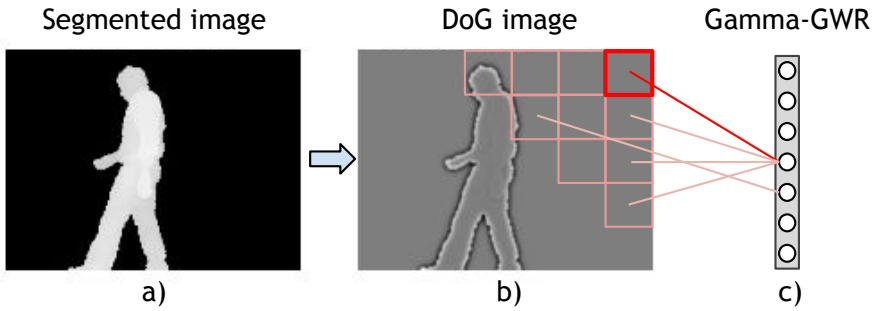



Figure 6.6: Body motion representation: (a) gray-scale segmented body silhouette; (b) DoG image divided into image patches; (c) recurrent competitive network for processing image patches.

processed frame. From this representation, we compute $4 \times 4$ patches that are fed into $G_2^P$ and $G_2^M$ with 3 context descriptors ($K = 3$) each. In the third layer, the pooled activation from the pose and the motion pathways are concatenated, producing $14 \times 7$ matrices for each frame that we use to train $G^{STS}$ with $K = 5$. If we consider the hierarchical processing of the architecture, the last network yields neurons that respond to the latest 10 frames, which correspond to 1 second of video.

Since the definition of the context descriptors is recursive, setting $\alpha_w > \alpha_1 > \alpha_2 > ... > \alpha_{K-1} > \alpha_K > 0$ has been shown to reduce the propagation of errors from early filter stages to higher-order contexts for the Gamma-SOM (Estévez and Hernández, 2011) and the Gamma-GNG (Estévez and Vergara, 2012). We assign decreasing values to $\alpha_i$ according to the following function:

$$\mathbf{p}_N = \left[ \frac{\alpha^i}{\sum_i \alpha^i} : \alpha^i = \frac{1}{N} - \exp\left(-(i+2)\right) \right], \quad i = 1, .., N \tag{6.11}$$

with $N = K + 1$, i.e. the number of context descriptors plus the current weight vector. For our three-layer architecture, this function yielded the following values:

| Network | $\mathbf{p}_N$ |
|---|---|
| $G_1^P$ / $G_1^M$ | $\mathbf{p}_2 = [0.536, 0.464]$ |
| $G_2^P$ / $G_2^M$ | $\mathbf{p}_4 = [0.318, 0.248, 0.222, 0.212]$ |
| $G^{STS}$ | $\mathbf{p}_6 = [0.248, 0.178, 0.152, 0.142, 0.139, 0.138]$ |

The training parameters were set based on the overall recognition accuracy over multiple training sessions, with the selected values being similar to the ones set in previous experiments (see Section 4.3 and 5.5). The training parameters for all the 5 networks are summarized in Table 6.3.

Table 6.3: Training parameters for the Gamma-GWR architecture.

| **Parameter** | **Value** |
|---|---|
| Insertion threshold ($a_T$) | $G_1^P$=0.7, $G_1^M$=0.6, $G_2^P$=0.6, $G_2^M$=0.5, $G^{STS}$=0.9 |
| Firing threshold | $f_T = 0.1$ |
| Learning rates | $\epsilon_b = 0.1$, $\epsilon_n = 0.001$ |
| Firing counter | $\tau_b = 0.3$, $\tau_i = 0.1$, $\kappa = 1.05$ |
| Context descriptor | $\beta = 0.7$ |
| Training epochs | 100 |

### Results on the KT Action Dataset

The KT action dataset that contains 10 full-body actions performed by 13 participants (see Section 4.3). The actions are *standing, walking, jogging, sitting, picking up, jumping, lying down, standing up, falling*, and *crawling*. Participants were recorded individually in a home-like environment with a Kinect sensor obtaining depth maps sampled at 30 frames per second. For a consistent comparison with previous results presented in Chapter 4 and 5, we adopted similar feature extraction and evaluation schemes with action sequences at 10 frames per second. We divided the data equally into training and test set, i.e., 30 sequences of 10 seconds for each cyclic action (*standing, walking, jogging, sitting, lying down, crawling*) and 30 repetitions for each goal-oriented action (*jumping, picking up, falling down, standing up*). Both the training and the test sets contained data from all subjects.

Experimental results showed an average classification accuracy of 97%, producing the best result on this dataset with respect to previously introduced approaches using hand-crafted action features (94% reported in Chapter 4 and 93, 3% reported in Chapter 5 without the manual segmentation of training samples for assigning ground-truth labels). The confusion matrix for the AG-GWR approach tested on a set of 10 actions is shown in Fig. 6.7. The matrix shows that there is a strong similarity on which samples were misclassified with respect to our previous models, suggesting again that misclassification depends more on the visual features than on issues related to the associative learning mechanism. Similar to the obtained results in previous chapters using 3D centroids as action features, actions that are similar with respect to body posture (e.g. *walking* and *jogging*, *falling down* and *lying down*), tend to be mutually misclassified. The reason for this is that although sequences of segmented body pose and motion should be sufficient to univocally describe action patterns, tracking inaccuracies and body segmentation errors may have a negative impact on the extraction of reliable pose-motion cues.

Similar to experiments in Chapter 5, we decreased the percentage of labeled action samples. We trained our system with as in the first experiment, but this time

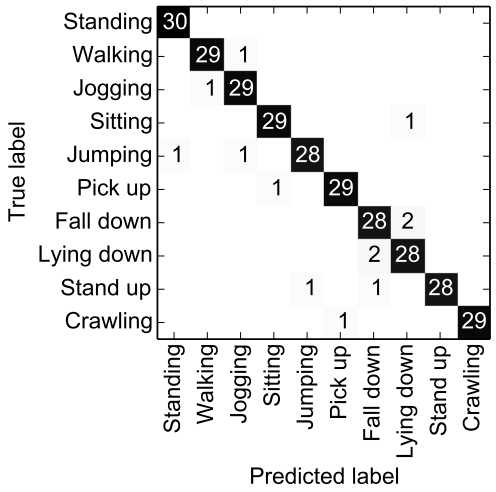



Figure 6.7: Confusion matrix for the AG-GWR approach tested on the KT action dataset.

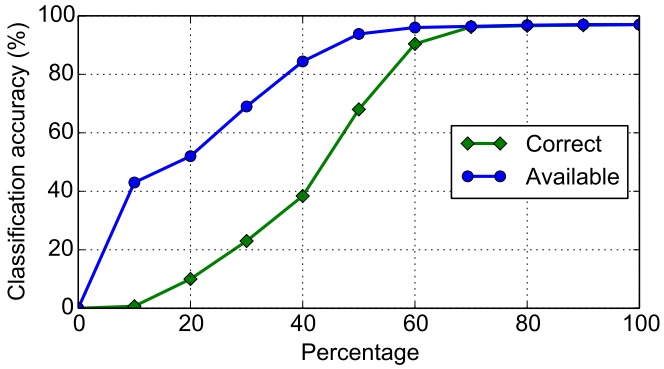



Figure 6.8: Average classification accuracy on the KT action dataset over 10 runs for a decreasing percentage of available and correct action labels.

we omitted action labels of randomly chosen samples and varied the percentage of available labels from 100% to 0%. The average classification accuracy with different percentages of omitted sample labels for randomly selected samples over 10 runs is displayed in Fig. 6.8. Although a decreasing number of available labeled samples during the training phase has a negative impact on the classification performance, this decline is not proportional to the number of omitted action labels. When 10% of labeled samples are available during the training, the system shows an accuracy of 43%, and accuracy values above 84% can be observed for 50% or more available labeled samples. Similar to the associative learning mechanism introduced in Section 5.3, we found that the timing at which these action labels are presented to AG-GWR layer over the training epochs does have a significant impact on the performance. The best results were obtained if action labels are

presented when visual representations have reached a certain degree of stability, while associative connections created at early stages of visual development may not be as reliable since the distribution of neurons will tend to significantly change during the rest of the training phase.

An additional experiment consisted of decreasing the percentage of correct sample labels. We changed correct ground-truth action labels of randomly chosen samples to incorrect ones, varying the percentage of correct labels from 100% to 0%. The average classification accuracy with different percentages of correct sample labels for randomly selected samples over 10 runs is displayed in Fig. 6.8. Different from the results obtained with different percentages of missing labels, incorrect labels had a stronger negative influence on the overall classification performance. This is because incorrect labels alter the frequency distribution of labeled samples. The overall accuracy decreases significantly when the percentage of correct labels is smaller than 60%.

### Results on the Weizmann Dataset

The Weizmann dataset (Gorelick et al., 2005) contains 90 low-resolution ($180 \times 144$) sequences with 10 actions performed by 9 subjects. The actions are *walk, run, jump, gallop sideways, bend, one-hand wave, two-hands wave, jump in place, jumping jack*, and *skip*. Sequences are sampled at $180 \times 144$ with a static background and are about 3 seconds long. For our experiments, we used aligned foreground body shapes by background subtraction included in the dataset (Fig. 6.10). To be consistent with other evaluation schemes in the literature, we evaluated our approach by performing *leave-one-out* cross-validation, i.e., 8 subjects were used for training and the remaining one for testing. This procedure was repeated for all 9 permutations and the results were averaged. Similar to Schindler and Van Gool (2008), we trimmed all sequences to a total of 28 frames, which is the length of the shortest sequence.

Similar to experiments on the KT action dataset, we decreased the percentage of available and correct labels (from 100% to 0%) from randomly chosen samples. The average classification accuracy with different percentages of omitted and incorrect labels over 10 runs is displayed in Fig. 6.8. As expected from previous experiments, incorrect labels had a stronger negative influence on the overall classification performance with respect to missing labels. Classification performance over 80% was obtained for at least 40% available labels, whereas in the case of incorrect labels, we obtained an overall performance under 40% for less than 50% correct labels.

Results for the recognition of 10-frame snippets are shown in Table 6.4. Our experiments yielded an overall accuracy of 98.7%, which is a very competitive result with respect to the state of the art of 99.64% reported by Gorelick et al. (2005). In their approach, the authors extract action features over a number of frames by concatenating 2D body silhouettes in a space-time volume. These features are then fed into simple classifiers: nearest neighbors and Euclidian distance. Schindler and Van Gool (2008) obtained an accuracy of 99.6% by combining pose and motion

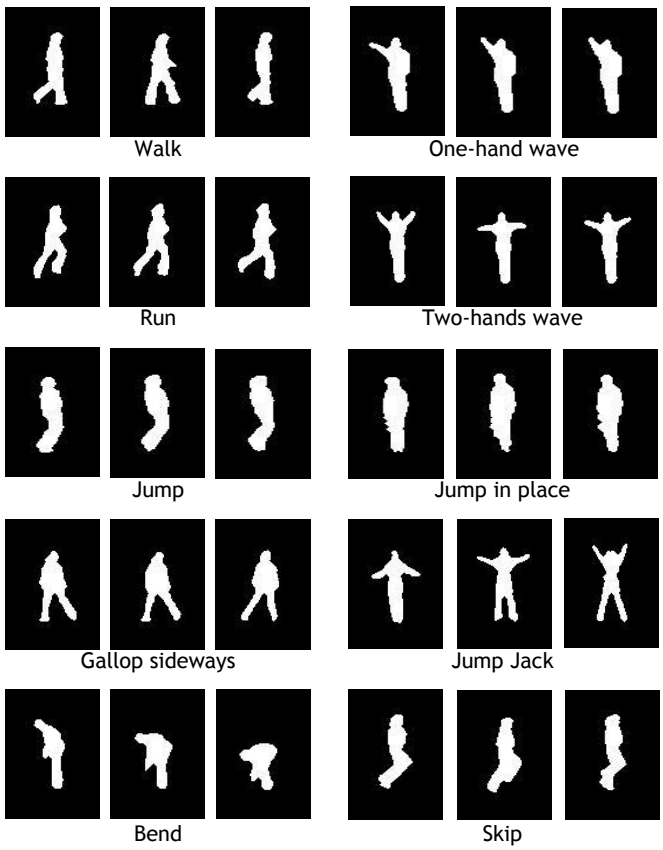



Figure 6.9: Three sample frames of body shapes by background subtraction for each action from the Weizmann dataset.

cues. In their two-pathway architecture, the filter responses are MAX-pooled and then compared to a set of learned templates. The similarities from both pathways are concatenated to a feature vector and classified by a bank of linear classifiers. Their experiments showed that local body pose and optic flow for a single frame are enough to achieve around 90% accuracy, with snippets of 5-7 frames (0.3-0.5 seconds of video) yielding similar results to experiments with 10-frame snippets. Our results outperform the overall accuracy reported by Jung et al. (2015) with three different deep learning models: convolutional neural network (CNN, 92.9%), multiple spatiotemporal scales neural network (MSTNN, 95.3%), and 3D CNN (96.2%). However, a direct comparison of the above-described methods with ours is hindered by the fact that they differ in the type of input and number of frames per sequence used during the training and the test phase.

Most of the results in the literature are reported at the level of correctly classified sequences. Therefore, we also evaluated our approach on full sequence classification to compare it to the state of the art. For each action sequence, we predicted labels from 10-frame snippets and then considered the prediction with the highest statistical mode as the output label for that sequence. Results at a sequence level are shown in Table 6.5.

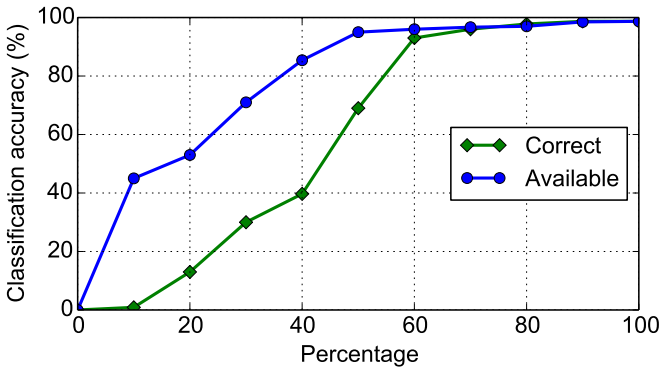



Figure 6.10: Average classification accuracy on the Weizmann dataset over 10 runs for a decreasing percentage of available and correct action labels.

Table 6.4: Results on the Weizmann dataset for 10-frame snippets. Results from Jung et al. (2015) with 3 different models: 1) CNN, 2) MSTNN, and 3) 3D-CNN.

| | **Accuracy (%)** |
|---|---|
| Gorelick et al. (2005) | 99.64 |
| Schindler and Van Gool (2008) | 99.6 |
| **Our approach** | **98.7** |
| Jung et al. (2015) | 92.9[1], 95.3[2], 96.2[3] |

Table 6.5: Results on the Weizmann dataset for full action sequences.

| | **Accuracy (%)** |
|---|---|
| Gorelick et al. (2005) | 100 |
| Blank et al. (2007) | 100 |
| Schindler and Van Gool (2008) | 100 |
| Fathi and Mori (2008) | 100 |
| **Our approach** | **100** |
| Jhuang et al. (2007) | 98.8 |
| Thurau and Hlavác (2008) | 94.4 |
| Niebles et al. (2008) | 90 |

## 6.4 Summary

Research efforts for tackling human action recognition and body motion assessment have produced a larger number of methodologies, ranging from rule-based systems to neural network architectures (see Section 2.2). In this Chapter, we showed that these two tasks can be approached using recurrent neural network self-organization.

In Section 6.2, we introduced a hierarchical self-organizing architecture for learning body motion sequences from 3D skeleton models. The quality of actions is computed in terms of how much a performed movement matches the correct continuation of a learned motion sequence template. For learning sequences and computing body motion prediction, we introduced a novel temporal extension of the GWR – the Merge-GWR – in the spirit of context learning (see Section 3.3). Our experiments showed that the Merge-GWR outperforms previous temporal self-organizing models, yielding a smaller temporal quantization error with respect to reported experiments on a time-series regression task. We evaluated our assessment system on a dataset with 3 powerlifting exercises, showing that the system can provide real-time visual feedback and detect motion mistakes in both single- and multiple-subject scenarios. Learning architectures comprising growing networks such as the GWR and the Gamma-GWR are particularly suitable for adapting to the user through longer periods of time while still detecting significant changes in their motor skills, e.g., gait deterioration linked to age-related cognitive decline. Future work could comprise the embedding of our system into an assistive robot which could interact with the user and motivate the correct performance of physical rehabilitation routines, sports training, and a set of health-related tasks (Dautenhahn, 1999; Kidd and Breazeal, 2007; Nalin et al., 2012). The detection of abnormal user behavior using self-organizing networks and the use of assistive robots will be discussed in Chapter 7.

In Section 6.3, we proposed a deep neural architecture with a hierarchy of recurrent GWR networks for learning action features with increasingly larger spatiotemporal receptive fields. Visual representations obtained through unsupervised learning are incrementally associated to symbolic action labels for the purpose of action classification. This is achieved through the use of an associative mechanism in the Gamma-GWR that attaches labels to prototype neurons based on their frequency. Different from previously introduced models in which action representations are learned from hand-crafted 3D features (see Chapter 4 and 5), we use a hierarchy of Gamma-GWR networks to learn prototype action segments from video containing full-body silhouettes. PCA-pooling is used to maximize the variance of projection of each layer, yielding invariance to scale and position along the hierarchy. Our experiments showed that this architecture outperforms previous self-organizing approaches on the KT action dataset. In order to compare our approach to state-of-the-art deep learning methods, we conducted experiments on the Weizmann action benchmark, showing competitive performance in different evaluation schemes. Additional experiments on both datasets showed that our learning architecture can also handle situations in which the number of available and correct ground-truth labels is decreased during the training phase.

# Chapter 7

# A Neurocognitive Robot for Multimodal Action Recognition

## 7.1 Introduction

In the previous chapters, we have illustrated a set of neural network architectures for the robust recognition of human body motion patterns. As the main contribution with respect to previous work in the literature, we have proposed a hierarchy of recurrent self-organizing networks with spatiotemporal receptive fields for learning action features with increasing complexity of representation (see Chapter 6). In contrast to traditional classification approaches that strongly rely on the ground-truth labeling of training samples, our approach associates unsupervised visual representations to available symbolic labels, with the availability and veracity of the labels not compromising the correct formation of visual action cues. The underlying motivation is to foster more flexible training procedures for artificial agents operating in real-world scenarios where, for instance, visual samples may be highly degraded due to sensor noise or action labels being occasionally unavailable or incorrect. As discussed in Chapter 2, the integration of multiple modalities is crucial for enhancing the perception of actions, especially in situations of uncertainty, with the aim to reliably operate in highly dynamic environments. However, experiments reported in Chapters 4, 5, and 6 were conducted with data collected in highly controlled environments, thus without considering a number of challenges introduced by agents operating in real-world scenarios. These challenging scenarios include, for instance, one of the modalities being occasionally unavailable or even in conflict to other sensor measurements.

In this chapter, we investigate aspects of multimodal integration for enhancing human-robot interaction and triggering sensory-driven robot behavior in dynamic environments. In particular, we propose two main scenarios. The first scenario consists of a robot-human assistance task, where a humanoid robot is used to monitor a user in a domestic environment with the aim to detect dangerous behavior such as falls. The humanoid integrates information from a depth sensor and a stereophonic microphone to actively track the user and report abnormal behavior

with respect to a set of learned domestic actions. In the second scenario, we propose a model of audiovisual integration for an interactive reinforcement learning task. In order to correctly perform the task, teacher-like feedback can be provided to the agent using both speech or hand gestures. Our model integrates dynamic audiovisual patterns and computes the level of confidence of the perceived feedback based on the available cues. Together, our experiments show that the integration of multiple modalities leads to a significant improvement of performance in both scenarios with respect to approaches relying on one single modality.

## 7.2 A Multimodal Approach for Abnormal Event Detection

We present a humanoid robot that tracks a person in daily activities and detects situations of danger such as falls. For this purpose, we use an array of sensors installed on the Nao and a multimodal controller that integrates heterogeneous sensory information to trigger Nao's motor behavior. The overall architecture of the system is shown in Fig. 7.1. Our system integrates multiple sensor modalities to enhance the perception of the robot through automatic speech recognition (ASR), sound source localization (SSL), and visual active tracking for fall detection. In the proposed scenario, the person can communicate with the Nao using speech commands. We use a depth sensor and a stereo microphone system to actively track the person by its motor abilities. In the case that the person is out of the field of view (FOV) of the depth sensor, SSL is used to locate the person and establish visual tracking. Information from Nao's sonar sensors is used to avoid obstacles in the environment. When the person asks for assistance or a fall is detected, the humanoid will approach the person and record the scene using the depth sensor's RGB channel of the depth sensor. This video recording can then be sent to the person's caregiver or relatives for further human evaluation.

Nao is a midsize humanoid robot developed by Aldebaran Robotics.[1] We extended the robot Nao with an ASUS Xtion Pro depth sensor installed on top of the head (Fig 7.1.a). The Xtion has an operation range between 0.8 and 3.5 meters with a VGA resolution ($640 \times 480$) at a maximum of 30 fps. In contrast to the Microsoft Kinect, the Xtion has reduced power consumption and also reduced weight. The main technical characteristics of the Xtion live sensor are listed in Table 7.1. For SSL, we used a Soundman OKM II binaural stereo microphone with omnidirectional polar pattern and a frequency range of 20Hz–20kHz. We installed the stereo microphone on the Xtion sensor with a distance of 14.5 cm between the right/left channels. We chose the Soundman microphones by comparing the SSL performance also with the stereo microphones embedded in the Nao and the Xtion (see Section 7.2.2). For ASR, we use a bluetooth headset (Sennheiser EZX 80) with an omnidirectional microphone that can be comfortably worn by the person and allows more robustness in noisy environments compared to the microphones

[1] Aldebaran Robotics - `http://www.aldebaran-robotics.com/`

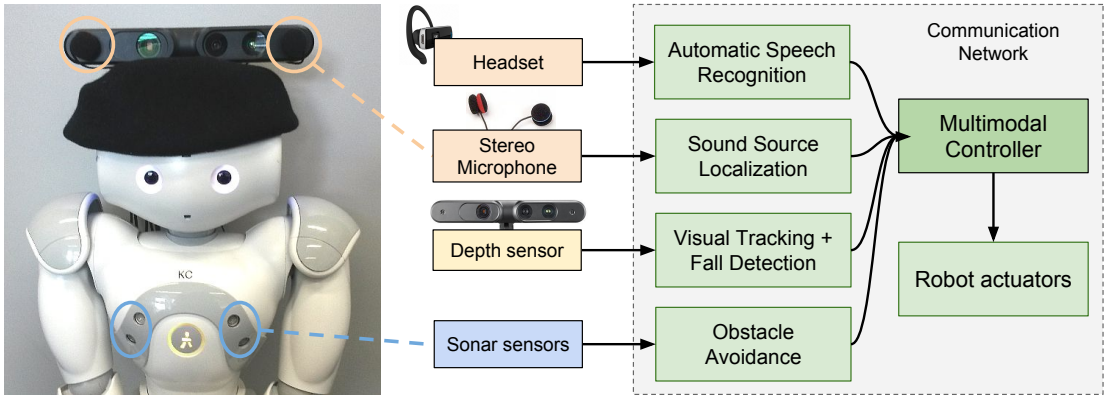



Figure 7.1: Overall architecture of our multimodal system – a) extended Nao with a depth sensor and stereo microphone, b) Communication network for covey sensor information to the multimodal controller for sensory-driven robot behavior.

Table 7.1: ASUS Xtion Live sensor specifications

| | |
|---|---|
| Depth Image Size | VGA (640x480) : 30 fps |
| Field of View | 58° horizontal, 45° vertical, 70° diagonal |
| Distance of use | 0.8 m to 3.5 m |
| Dimensions | $18 \times 3.5 \times 5$ cm |
| Power consumption | below 2.5 W |
| Interface | USB 2.0/3.0 |
| Weight | 227 g |

embedded in the Nao, especially when the robot is moving. We use Nao's sonar sensors to detect obstacles on the way. The sonar sensors have an effective cone of 60° with a resolution of 1 cm and a detection range from 0.25 to 2.55 meters.

### 7.2.1 Active Tracking

The Xtion depth sensor is characterized by a reduced FOV (58° horizontal, 45° vertical, 70° diagonal), limiting its use in expansive environments. This motivates the implementation of an active tracking system, which moves the sensor to keep the person in the scene. We use Nao's head to move the sensor and increase the horizontal FOV from 58° to 138° (Fig. 7.2). Nao will then smoothly pan its head by 10° degrees in the required direction, for a maximum pan angle of 40° degrees in each direction. As a strategy for active tracking, we define a bounding box in which the person can act without the sensor being moved (Fig. 7.3). We base the tracking of the person on a 3D skeleton model and consider the point of the upper-body torso as the reference of the person's position (see Section 4.2). When

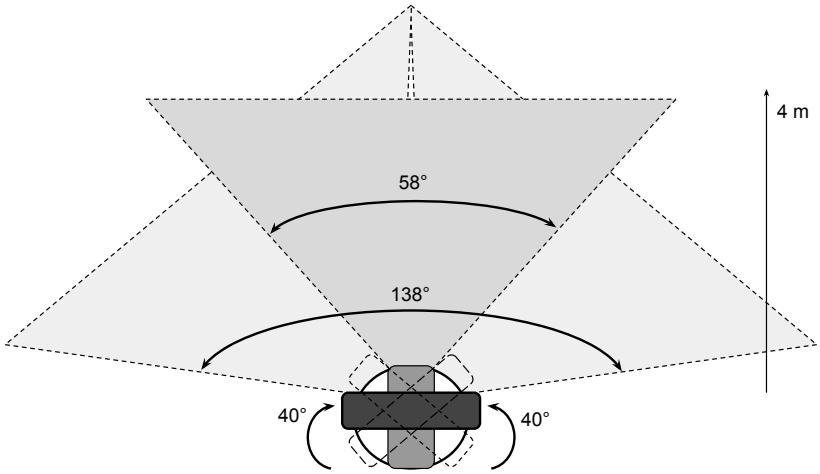



Figure 7.2: Nao with Xtion sensor: extended horizontal field of view from 58 to 138 degrees with a maximum head pan angle of 40 degrees in each direction (Parisi et al., 2016c).

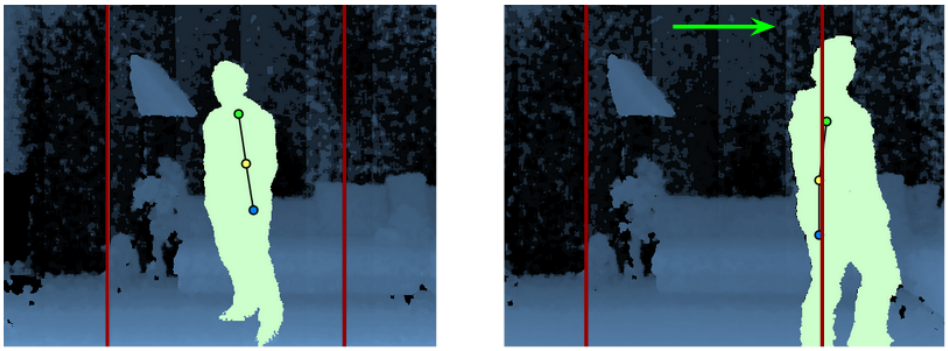



Figure 7.3: Threshold-based active tracking. When the upper-body centroid lies outside of the threshold, the tracking application will compute the needed operations to keep the person within the bounding box (red lines) (Parisi et al., 2016c).

the torso point lies beyond the threshold, the tracking application will compute the operations required to keep the person within the bounding box.

The tracking application is built on top of simple-openni[2], which wraps the OpenNI–NITE framework[3] for user identification, calibration and estimation of skeletal joints. We use this library with Processing IDE[4] with skeleton tracking provided by OpenNI. In this setting, we obtain the angle of the person with respect to the sensor as follows:

$$\alpha = \arctan([x - (x_{max}/2)]/z_{max}), \tag{7.1}$$

[2]simple-openni: `https://code.google.com/p/simple-openni/`
[3]OpenNI/NITE: `http://www.openni.org/software`
[4]Processing IDE: `http://processing.org/`

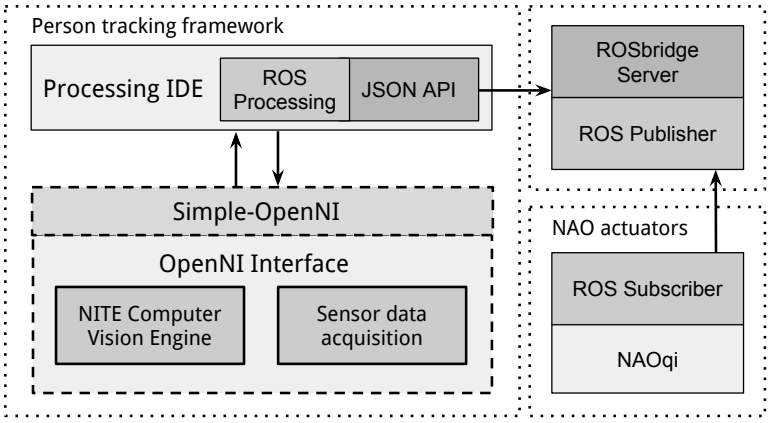



Figure 7.4: A diagram of the communication network for interfacing the tracking framework with Nao's actuators over ROS (Parisi et al., 2016c).

where $x$ is the position of the torso joint with respect to the horizontal image plane, $x_{max}/2$ is the center of this plane, and $z_{max}$ is the focal length (max. depth value).

All system modules for active tracking communicate over Robot Operating System (ROS), a framework for robot software development with operating system-like functionality on a heterogeneous computer cluster. It provides hardware abstraction, device drivers, libraries, visualizers, message-passing between processes and package management. A diagram of the overall architecture for active tracking is illustrated in Fig. 7.4. To interface our system modules, we use a ROS-based communication network implemented with *publisher-subscriber* nodes. We implement publisher nodes to continually broadcast a message over the network using a message-adapted class. The subscriber node will receive the messages on a given topic via a master node, which keeps a registry of publishers and subscribers. This specific architecture represents a robust interface to connect different applications, e.g. written in different programming languages, over a common network of communication. The tracking framework communicates to ROS over Rosbridge[5] and a modified version of ROSProcessing,[6] extended to publish ROS topics. Rosbridge provides a JSON API[7] to ROS functionality for non-ROS programs. The rosbridge_suite package is a collection of packages that implement the rosbridge protocol and provides a WebSocket transport layer. We program Nao to move its head according to the tracking application via NAOqi framework,[8] which allows homogeneous communication between different Nao modules (motion, audio, video), and ROS integration.

[5]rosbridge_suite: `http://wiki.ros.org/rosbridge_suite`

[6]ROSProcessing: `https://github.com/pronobis/ROSProcessing`

[7]JSON API: `http://jsonapi.org/`

[8]NAOqi framework: `https://community.aldebaran-robotics.com/doc/1-14/dev/naoqi/index.html`

### 7.2.2 Sound Source Localization

There are a number of auditory cues that can be used for sound-source localization. Most of these cues are derived from the spatial separation of sensors. Among these are the difference in the time at which sounds arrive at each microphone (time difference of arrival, TDOA), the difference in intensity (interaural intensity difference, IID), and spectral variations in the signals (Knapp and Carter, 1976). Any number of microphones greater than two can be used in principle, but the hardware and computational cost sharply rises with each additional microphone.

In our scenario, we require fast and reliable SSL. On the other hand, high accuracy is not an issue. Therefore, we choose a simple but reasonably accurate binaural solution which extracts the TDOA from a stereo signal using the cross-correlation algorithm (Schnupp et al., 2010). This algorithm shifts the signals from the individual microphones with respect to each other and determines the shift producing the greatest cross-correlation. That shift corresponds to the TDOA and thus to the angle of incidence.

It is possible to compute the angle of incidence for a given TDOA from the geometry of the system. However, since the estimate of the TDOA computed by the cross-correlation algorithm can be smeared by the acoustic properties of the environment, the robot body, and the ego-noise it produces, we opted for an empirical approach: We recorded $60\,s$ of recorded speech from 19 directions at $10°$ intervals between $-90°$ and $90°$ from the robot. We split each of the recordings into $0.25\,s$ snippets and computed the relative time shift maximizing the cross-correlation between the channels for each snippet. For each occurring time shift, we then selected that angle of incidence for which it occurred most often as the most likely angle of incidence. We did this for three different sets of microphones: the Nao's own microphones, those of the Xtion sensor, and the Soundman microphones.

Figure 7.5 shows histograms of maximizing time shifts for each angle. The TDOA estimated by the cross-correlation algorithm was strongly correlated to the angle of incidence for all stereo microphones, as expected. However, the degree of correlation, measured by Spearman's rank correlation coefficient, differed drastically (Nao: $\rho = 0.506$, Xtion: $\rho = -0.714$, Soundman: $\rho = -0.930$; $p << 0.0001$ for all microphones). We therefore chose the Soundman microphone for SSL.

### 7.2.3 Automatic Speech Recognition

For ASR, we used Google's cloud-based speech recognition with domain-dependent post-processing (Twiefel et al., 2014). In this approach, the post-processor translates each sentence in the list of candidate sentences returned by Google's service into a string of phonemes. To be able to exploit the quality of the well-trained acoustic models employed by Google's service, the ASR hypothesis is converted to a phonemic representation employing the SequiturG2P grapheme-to-phoneme converter. Then, the sentence from a list of in-domain sentences is selected as the most likely sentence, which has the least Levenshtein distance to any of the candidate phoneme strings. For our implementation, we used the 10 top results

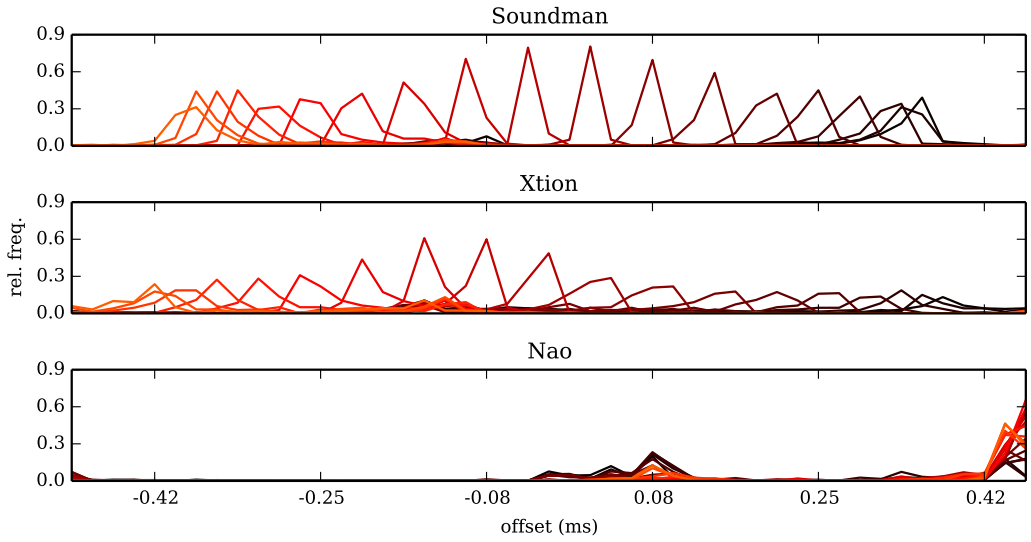



Figure 7.5: Results of SSL with cross-correlation using different stereo microphones – Histograms of maximizing time shifts for each angle for the Soundman, Xtion, and Nao microphones. Each shade represents a histogram for one angle.

and the target sentences.

An advantage of this approach is the hard constraints of the results, as each possible result can be mapped to an expected sentence. Experiments reported by Twiefel et al. (2014) showed that the sentence list approach obtained the best performance for in-domain recognition with respect to other approaches on the TIMIT speech corpus[9] with a *sentence-error-rate* of 0.521. The sentences that we use for our scenario are: "*Look at me*", "*Come to me*", "*Turn around*", "*Turn to me*", "*Help me*", "*Yes, please*", "*No, thank you*", and "*Stop*".

### 7.2.4 Multimodal Controller

The multimodal controller modulates the motor behavior of the humanoid and other operations of the system based on the information conveyed by the different sensors. This module is responsible for estimating the reliability of the modalities in terms of last arrived valid signal from the audio-visual modules.

When the vision-based position is not available or the last tracked position is older than 3 seconds, then SSL will be used. If the last valid SSL angle is older than 3 seconds, then the robot will ask *Where are you?* and wait for either audio or visual input. If audio-visual inputs are in conflict, i.e. the user's position estimated by the tracking framework and the SSL are widely discrepant, then more priority will be given to the visual estimation. This is due to the fact that the SSL module is more likely to return unreliable estimations, e.g., in situations

[9]TIMIT Acoustic-Phonetic Continuous Speech Corpus: `https://catalog.ldc.upenn.edu/LDC93S1`

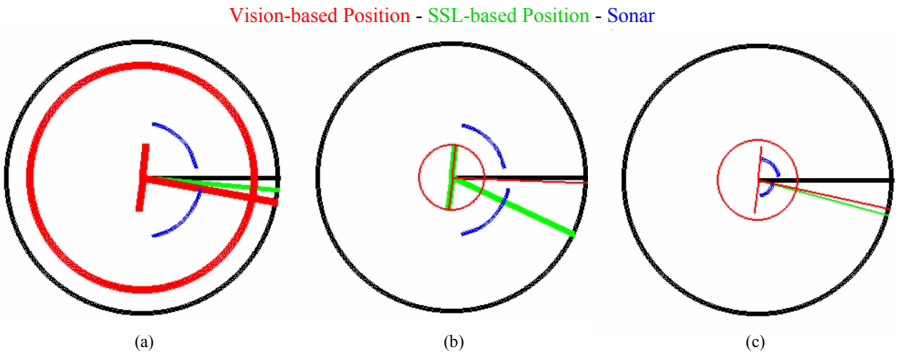



Figure 7.6: Visualization of multimodal robot perception in a ratio of 3 meters - The thickness of the lines represents the reliability of the information sources (the thicker the more reliable). (a) Visual information is used to estimate the position of the person (SSL is also computed but not used); (b) When visual information is not available (e.g. out of date), then SSL is used to estimate the position; (c) The person is too close to the robot (30 cm) so that the depth sensor cannot track the position (out of the operation range) and the sonar sensors detect a possible obstacle.

with strong background noise. When the robot is moving, the controller uses Nao's sonar sensors to stop before obstacles that can cause the damage to the robot. A visual example of the interplay of different modalities is shown in Fig. 7.6.

At any time, the robot can receive vocal commands that have priority over other modules. The person can use a set of short commands to interact with the robot that will result in the following behaviors. For *Look at me*, Nao will orient towards the person in the environment using vision and audio. If the position of the person is not known through vision (out of the FOV or occluded), the robot will use SSL. If still, the robot is not able to estimate the position of the person, it will ask *"Where are you?"* and wait for hints delivered vocally. For *Come to me*, the robot approaches the person to a fixed distance of 1 meter using the last estimated position. When the person is not in the FOV of the robot, the command *Turn to me* is used to rotate the robot (not only the head) towards the person and then establish visual contact. For *Turn around*, the robot will perform a 180° turn. The command *Stop* will terminate any operation that the robot is performing, e.g., interrupting a turn or stopping the approaching robot at a desired distance.

When the person says *Help me* or when a fall is detected through vision (see Section 7.2.5), Nao will approach the person and ask whether assistance is required (e.g., to stand up in the case of fall). If the answer is *Yes, please* or no vocal answer is detected, Nao can get in contact with the person's caregiver or relative for further assessment of the situation. In the case of a fall, the system will store the last 5 seconds of activity before the fall as an RGB video that can be used to evaluate the seriousness of the event. The fall detection scenario is illustrated in Fig. 7.7.


(a) (b)

Figure 7.7: Fall detection scenario: When a fall event is detected, the Nao will approach the person (a) and take a picture of the scene (b) (Parisi et al., 2016c).

### 7.2.5 Fall Detection

To detect fall events, we use a learning-based approach that reports novel behavioral patterns that were not presented during the training phase (Parisi and Wermter, 2013). We train a neural network architecture on a dataset of 3D body motion from depth map videos comprising normal behavior, i.e. domestic actions such as walking, sitting, and lying down, and then trigger an alarm when abnormal behavioral patterns are detected. To contrast sensor noise and tracking errors, the neural architecture is also responsible for automatically removing noisy samples from the extracted body features during the training and test stage. Experiments in a home-like environment (Fig. 7.8) showed that the system detects falls with 96% accuracy.

The combination of a depth sensor with the learning-based approach allows us to tailor the robust detection of fall events independently from the background surroundings and changing light conditions. This is especially advantageous in scenarios with a mobile sensor.

#### Learning Framework

Unsupervised neural network learning has shown to be a prominent approach for the detection of abnormal events (Hu et al., 2004a), also referred to as anomaly detection (Chandola et al., 2009). We propose a hybrid neural-statistical framework to approximate the normal behavior with trained self-organizing map (SOM) networks and subsequently detect behavioral patterns that do not conform to the expected learned behavior with an abnormality test.

The SOM is a competitive neural network introduced by Kohonen (1990) that has been shown to be a compelling approach for clustering motion expressed in terms of multi-dimensional flow vectors (Hu et al., 2004b; Nag et al., 2005). The

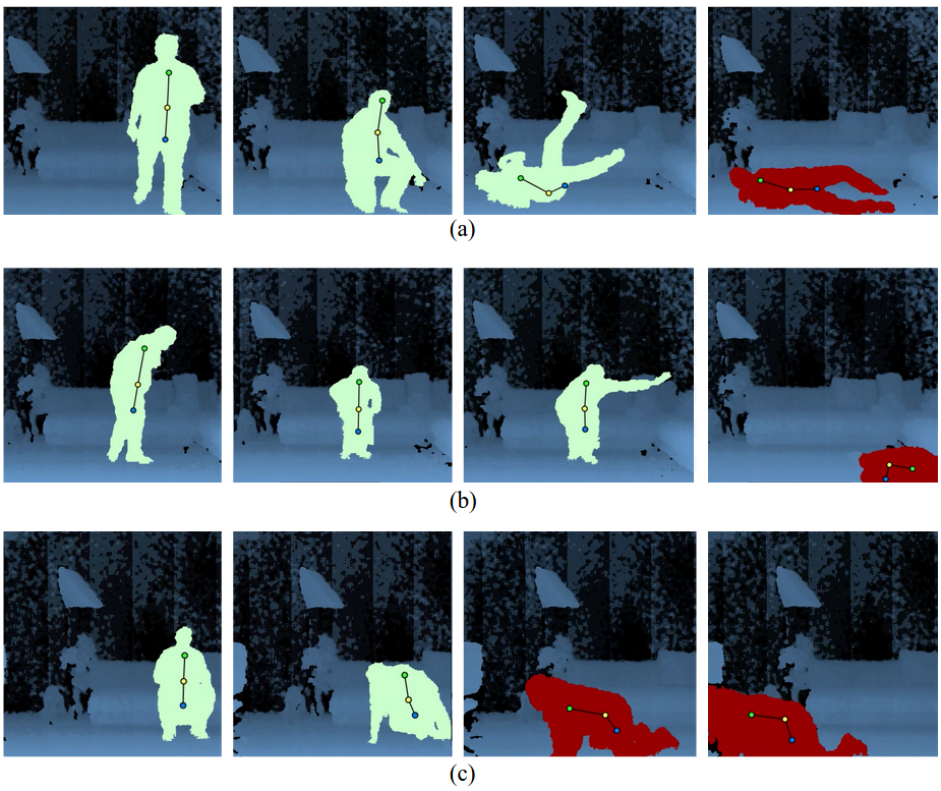



Figure 7.8: Abnormal event detection from video sequences. The system can successfully detect abnormal actions and report them (red body) (Parisi et al., 2016c).

proposed learning framework consists of three SOM networks. We consider two-dimensional networks with units arranged on a hexagonal lattice in the Euclidean space and a Gaussian neighborhood function trained with a batch variant of the SOM algorithm (see Section 3.2.1). A first network $\Phi_0$ is trained to detect outlier values from the extracted pose-motion vectors caused by tracking errors and sensor noise. During the second learning stage step, a hierarchical SOM-based approach is used to learn spatiotemporal properties of action sequences from denoised training samples. After this initial learning phase, the pose-motion vectors are processed again to perform a threshold-based test and remove outliers from the training set. The denoised training set is then fed to a hierarchical SOM-based architecture composed of two networks, $\Phi_1$ and $\Phi_2$, for clustering the subspace of normal actions taking into account spatiotemporal relationships of action sequences. A flow chart of this learning stage is illustrated by Fig. 7.9. At detection time, extracted vectors will be denoised and processed through the hierarchy of trained SOM networks. New observations that deviate from the learned behavior, i.e. below an abnormality threshold, will be reported as abnormal. The detection of noise and abnormal behavior is based on the same abnormality test using two different automatically computed thresholds.

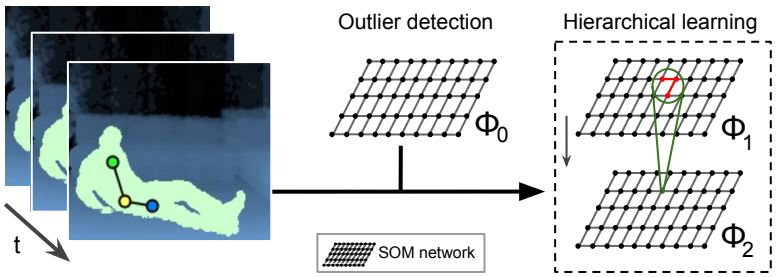
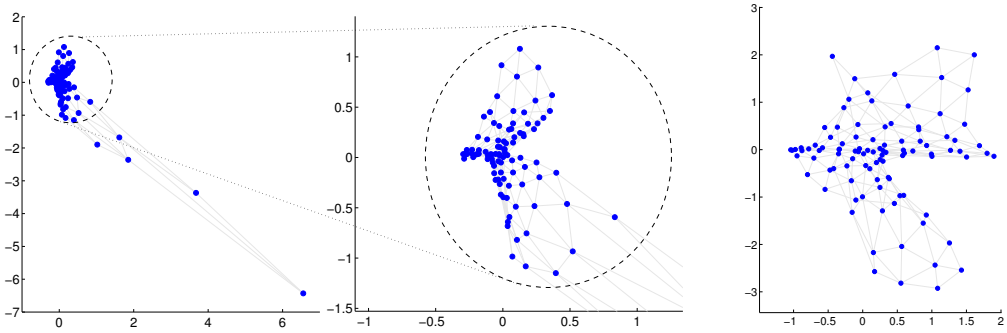



Figure 7.9: Flow chart of our SOM-based learning stage. A first network $\Phi_0$ is trained to detect and remove outliers from extracted pose-motion vectors. Preprocessed vectors are fed to a hierarchy of networks ($\Phi_1$ and $\Phi_2$) to cluster spatiotemporal relationships of action sequences (Parisi et al., 2016c).



(a) (b)

Figure 7.10: Effects of outliers in the clustering of training data (Parisi and Wermter, 2013) – (a) The first SOM was trained with the full set of extracted motion vectors. The presence of highly noisy observations in the training set decreased the unfolding of the projected feature map; (b) This second SOM was trained after removing outliers from the training set, gave a more representative clustering of the observations from tracked motion.

## Tracking Errors

An outlier can be seen as an observation that does not follow the pattern suggested by the majority of the observations belonging to the same data cloud (Nag et al., 2005). From a geometrical perspective, outliers are to be found detached from the dominating distribution of the subspace of normal actions.

In our approach, we differentiate between outliers introduced by tracking errors and outliers caused by tracked abnormal events. For this purpose, we assume that the behavior of a moving target must be consistent over time. Therefore, we consider highly inconsistent changes in body posture and speed to be caused by tracking errors rather than the actual tracked motion. As shown in our experiments, the presence of tracking errors in the training set may negatively affect the SOM-based clustering of pose-motion features. Fig. 7.10 illustrates these effects after the learning phase. A first SOM was trained with the full set of extracted mo-

tion vectors, for which outliers in the data decreased the unfolding of the projected feature map (Fig. 7.10.a). These noisy samples were detected by our algorithm and removed from the training set. As seen from the second SOM trained with the denoised training set (Fig. 7.10.b), the absence of outliers allowed a more representative clustering of the motion vectors for the subspace of normal actions.

While we use the same algorithm to detect outliers, two different abnormality thresholds are automatically computed that take into account the different characteristics of tracking noise and abnormal pose-motion vectors. Tracking errors in the test set are detected using this first trained SOM network as a reference and then removed.

## Abnormality Detection Algorithm

The goal of the detection algorithm is to test if the most recent observation is abnormal or not. For this purpose, the degree of abnormality for every test observation is expressed with the estimation of a P-value. If the P-value is smaller than a given threshold, then the observation is considered to be abnormal and reported as such.

For a given training set $X$ and a new test observation $x_{n+1}$ presented to the network $\Phi$, the algorithm is summarized as follows (Hoglund et al., 2000):

0. Compute the set of quantization errors $Q = (q_1, q_2, ..q_n)$.

1. Compute $q_{(n+1)}$ with respect to $\Phi$.

2. Define $B$ as the number of quantization errors $(q_1, ..., q_n)$ greater than $q_{(n+1)}$.

3. Define the abnormality P-value as $P_{(n+1)} = B/n$.

As an extension to the algorithm proposed by Hoglund et al. (2000), abnormality thresholds are automatically computed for the trained networks $\Phi_0$ and $\Phi_2$. The choice of convenient threshold values that take into account the characteristics of the distributions can have a significant impact on the successful rates for abnormality detection. From a neural network perspective, the threshold values will consider the distribution of the quantization errors from each trained SOM. Based on previous research (Parisi and Wermter, 2013), we empirically define two different thresholds, $T_O$ for outlier detection and $T_A$ for abnormality detection:

$$T_O = \beta\sqrt{\overline{Q_O} + \sigma(Q_O) + \max(Q_O) + \min(Q_O)}, \tag{7.2}$$

$$T_A = \gamma\left[\frac{\overline{Q} + \sigma(Q)}{\max(Q) + \min(Q)}\right], \tag{7.3}$$

where $Q_O$ and $Q$ denote the quantization error sets for $\Phi_0$ and $\Phi_2$ respectively, $\overline{Q}$ denotes the mean value operator, $\sigma(Q)$ denotes the standard deviation, and

$\beta = 0.5$, $\gamma = 0.1$. In the case of $\Phi_0$, observations with P-values under the abnormality threshold $T_O$ are considered as outlier values and therefore removed from the training set. For $\Phi_2$, if $P_{(n+1)}$ is smaller than $T_A$, then the test observation $x_{n+1}$ is considered abnormal.

## Experimental Results

For the training and evaluation of the system, we used video sequences from the KT action dataset with full-body actions performed by 13 different participants with a normal physical condition (see Section 4.2). To avoid biased execution, the participants were not instructed on how to perform the actions. Training video sequences consisted of domestic actions such as *walking, sitting down and standing up*, and *bending to pick up objects*, whereas abnormal actions comprised *falling down* and *crawling*. We did not take into account those cases in which the user is already fallen on the ground since the tracking framework built on top of OpenNI would fail to provide a reliable recognition of the user and therefore the extraction of body features would be highly compromised.

At detection time, new extracted vectors were processed to remove outliers. For the last three denoised vectors, a new test trajectory $\tau_{i+1}$ was obtained from $\Phi_1$ and then fed to $\Phi_2$ to compute the abnormality test $\lambda(\tau_{i+1})$. We took the last 3 abnormality test results and returned as abnormality output the result of the statistical mode:

$$Mo(\lambda(\tau_{i+1}), \lambda(\tau_{i+2}), \lambda(\tau_{i+3})). \tag{7.4}$$

A new output was therefore returned every 9 samples, which corresponds to approximately less than 1 second of captured motion. As shown by our experiments, this approach led to increased detection accuracy.

We evaluated the detection algorithm on abnormal actions using standard measurements defined by Van Rijsbergen (1979):

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \tag{7.5}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \tag{7.6}$$

$$\text{F-score} = 2 \cdot \frac{\text{Recall} \cdot \text{Precision}}{\text{Recall} + \text{Precision}}, \tag{7.7}$$

$$\text{True negative rate} = \frac{\text{TN}}{\text{TN} + \text{FP}}, \tag{7.8}$$

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}. \tag{7.9}$$

A true positive (TP) was obtained when an abnormal event was detected between the first and the last frame where the abnormal action took place. True negatives (TN) refer to normal actions not detected as abnormal. False positives

Table 7.2: Performance of our abnormality detection algorithm on a data set of 13 participants.

| | Raw | Denoised | Improvement |
|---|---|---|---|
| Recall | 88% | **95%** | 7.02% |
| Precision | 90% | **97%** | 7.02% |
| F-score | 89% | **96%** | 7.02% |
| TN rate | 90% | **97%** | 6.90% |
| Accuracy | 89% | **96%** | 6.96% |

(FP) and false negatives (FN) refer respectively to normal actions reported as abnormal and abnormal behaviors not reported by the system.

The system evaluation is shown in Table 7.2. Our system detected abnormal fall events with 96% accuracy. The removal of noise from the training and test set was of significant importance for reducing detection errors in presence of partial occlusions and tracking errors introduced by the mobile sensor, with an improvement in accuracy of 6.96%. On the other hand, the accuracy of our system would be negatively influenced by: 1) highly-occluded users, leading to tracking errors and compromised feature extraction; and 2) the presence of actions sharing similar body features subject to classification ambiguity, i.e. detecting lying down as a fall, leading to a greater number of false positives.

The obtained results motivate future work in several directions. At the current state of the system, the depth sensor must be wired to an external, fixed processing unit to perform the tracking, thereby limiting the mobility of the humanoid. To achieve better mobility, the sensor could be wired to an onboard processing unit and then transmit the depth information via WiFi for further processing to be carried out in the cloud. Moreover, video files could be adopted instead of a single picture to better support telematic human evaluation, e.g. sending a video with the last 5 seconds of the user's activity before the fall event. In fact, the role of human assessment is of crucial importance to determine the seriousness of the detected event and to undertake effective intervention.

From a navigation perspective, the robot does not have any representation about the operational environment. A possible extension is to provide Nao with prior knowledge of the properties of the environment using a ceiling camera (Yan et al., 2013) or a mechanism for self-localization and mapping such as RatSLAM extended for humanoid robots (Müller et al., 2014). This would enhance Nao's navigational capabilities for, e.g., a scenario with multiple rooms in a residential context. Additionally, proxemic behaviors could be explored for socially-acceptable scenarios to navigate safely in a cluttered and dynamically changing domestic environment (Torta et al., 2011).

## 7.3 Integration of Dynamic Audiovisual Patterns

### 7.3.1 Introduction

Reinforcement Learning (RL) is an approach based on behavioral psychology in which an agent autonomously explores its environment in order to find an optimal policy to perform a given task (Sutton and Barto, 1998). At each state, the agent selects an action to perform in order to obtain a reward and reaching a new state. However, a known issue of this approach is the high number of training episodes required by the agent to learn a proper policy. In this regard, interactive reinforcement learning (IRL) has added an external parent-like trainer in order to speed up the learning process through either providing a reward (Thomaz and Breazeal, 2007) or policy shaping (Griffith et al., 2013). Therefore, robot learning can be sped up with the use of parent-like trainers who provide useful advice, allowing robots to learn a specific task in less time compared to a robot exploring autonomously (Cruz et al., 2015). In this regard, the parent-like trainer guides the apprentice robot with actions that allow to enhance its performance the same way as external caregivers may support infants in the accomplishment of a given task, with the provided support frequently decreasing over time. This teaching technique is known as parental scaffolding (Ugur et al., 2015).

When interacting with their caregivers, infants are subject to different environmental stimuli which can be present in various modalities. In general terms, it is possible to think about some of those stimuli as a guidance that the parent-like trainer delivers to the apprentice agent. However, when multiple modalities are considered, issues may emerge regarding the interpretation and integration of multimodal information, especially when the information from multiple sources is in conflict or ambiguous, e.g., yielding low confidence levels (Bauer et al., 2015). Consequently, instructions may not be clear and misunderstood, thereby leading to a decreased performance in the apprentice agent when solving a task (Cruz et al., 2016a). Although IRL approaches have been implemented in robotic scenarios (e.g., Suay and Chernova (2011); Knox et al. (2013)), an open issue is that the communication interface between the trainer and the robot in home-like environments is quite tedious and cumbersome for non-expert trainers. Therefore, there is a motivation to develop a more natural interactive scenario where external parent-like trainers can provide instructions using their natural communication skills such as speech and gestures.

In this section, we present a multimodal interactive reinforcement learning scenario which consists of a robot learning a domestic task. The robot can manipulate two objects with the goal of cleaning a table. During the learning process, advice can be provided by a parent-like trainer using vocal commands or performing hand gestures. Our proposed architecture is able to process information from multiple sources through the use of a neural associative memory that computes multimodal advice as a function of the recognition and confidence of unimodal modules. A general overview of the architecture is depicted in Fig. 7.11, where $\lambda$ and $\gamma$ are the label class and the confidence value respectively. First, auditory and visual patterns are

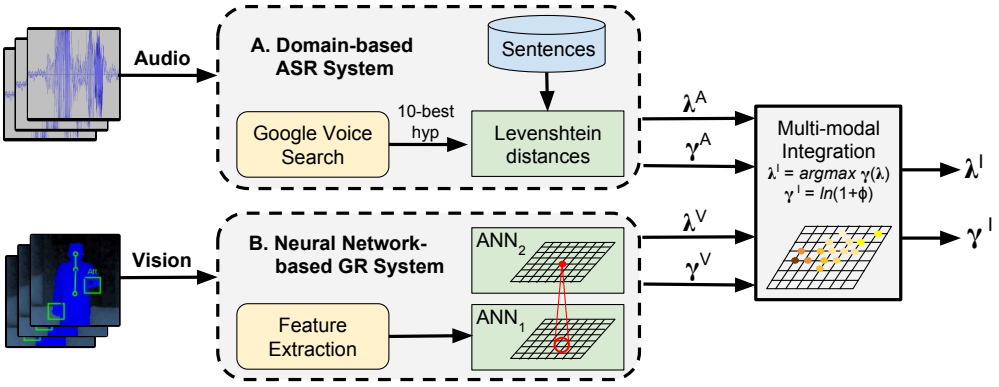



Figure 7.11: Overall view of our multimodal system architecture. The domain-based ASR system (Box A) processes auditory input to obtain an advice class label $\lambda^A$ and a confidence value $\gamma^A$, while the neural network-based gesture recognition system (Box B) processes visual input to predict an advice class label $\lambda^V$ and its confidence value $\gamma^V$. Subsequently, an associative neural network is used to compute an integrated advice label $\lambda^I$ and confidence value $\gamma^I$ (Cruz et al., 2016b).

processed individually as described in Sections 7.4.3 and 7.4.4 respectively. Then, predicted class labels for vocal commands and gestures along with their confidence values are used as input for the multimodal integration system (Section 7.4.5). We present a set of experiments using 7 possible advice classes from audiovisual inputs, showing that multimodal integration leads to a better performance of interactive reinforcement learning, with the robot being able to learn using a smaller number of training episodes compared to unimodal scenarios.

### 7.3.2 Robot Scenario

We extended the approach proposed in Cruz et al. (2015) to incorporate visual information and integrate it with audio as a more robust guidance during the learning process. The scenario consists of a humanoid robot in front of a table to clean it, comprising two objects that the robot can manipulate to achieve this task. The two objects are:

1. *cup*, which is initially placed at any location of the table and should be moved in order to finish the cleaning task;

2. *sponge*, which is used along with the robot's hand to clean different positions of the table.

For each object, we defined three locations: the *right* and *left* parts of the table, and an additional position defined as *home*, where the sponge should be placed when not in use. We define a set of 7 possible advice classes that can be given to the robot by a parent-like trainer. Each advice class has a spoken representation in a domain-based language and a visual representation with gestures from vision.

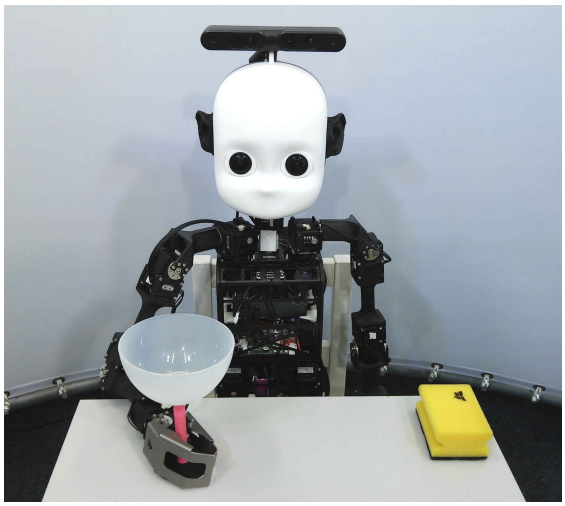


Figure 7.12: Cleaning scenario with the NICO robot. Our scheme is composed of 2 objects, 3 locations, and 7 action classes (Cruz et al., 2016b).

The advice can be delivered at any time using speech, gestures, or both with the following advice classes:

1. *get*, which allows the robot to pick up the nearest object to its gripper;
2. *drop*, which allows the robot to put down the object held in its hand;
3. *go* < *location* >, which moves the robot's gripper to one of the three defined locations: *go home*, *go left*, and *go right*;
4. *clean*, which allows the robot to clean the table surface at the current hand position;
5. *abort*, which cancels the execution of the cleaning task at any time.

Fig. 7.12 shows an example of the domestic scenario with our Neural Inspired COmpanion (NICO) robot. We use a microphone and a depth sensor to capture the advice from the parent-like trainer that is subsequently integrated and sent to the IRL algorithm as one single piece of consistent advice. The integrated advised action is then sent to the NICO robot to be performed using the *pypot* library (Lapeyre et al., 2014), allowing to control the robot actuators either in real or simulated environments.

### 7.3.3 Automatic Speech Recognition

The apprentice robot processes vocal advice by applying automatic speech recognition (ASR) based on *Google Voice Search* (GVS, Schalkwyk et al. 2010), a cloud-based ASR service to process audio data captured by a local microphone and generating hypotheses for the corresponding text representation. To overcome the issue of out-of-domain language models, we use DOCKS (Twiefel et al., 2014), a

post-processing technique to fit the ASR hypotheses provided by GVS to the given human-robot interaction (HRI) domain.

For our scenario, we define a set of robot commands represented by a list of sentences. To identify the best-matching hypothesis out of the list of sentences, the phonemic representation of the ASR hypothesis is compared to the phonemic representation of each sentence in the list. For this task, the Levenshtein distance (Levenshtein, 1966) is used to compute the difference of the obtained phoneme sequences. Then, the sentence having the shortest distance is chosen as the best-matching result. The Levenshtein distance is computed for the 10-best hypotheses provided by GVS. Given the set $H$ of the best-10 hypotheses and the set $S$ of the reference sentences, the predicted class label is computed as $\lambda^A = \text{argmin } \mathcal{L}(h_i, s_j)$, where $\mathcal{L}$ is the Levenshtein distance in our ASR system. The confidence value is computed as $\gamma^A = \max(0, 1 - \mathcal{L}(h_i, s_j)/|s_j|)$ with $h_i \in H$ and $s_j \in S$, both represented as phonemes. A diagram of the ASR system is illustrated in Box A of Fig. 7.11.

### 7.3.4 Gesture Recognition

People perform body gestures and co-speech gestures rather unconsciously in everyday life, for instance, when we explain the shape of an object (Dick et al., 2012). Similarly, gestures may be a convenient and complementary way to communicate in HRI scenarios. In Parisi et al. (2014a,b), we proposed two learning architectures for the recognition of a set of gesture classes in real time. Both the approaches were based on the idea of hierarchical learning with self-organizing networks as presented in Chapter 4. In Parisi et al. (2014a), we implemented a neural architecture for learning both static and dynamic hand gestures. Hand features were extracted from RGB-D video sequences (Fig. 7.13) and subsequently processed by a SOM-based architecture in terms of hand pose-motion features (Fig. 7.14). Reported results evidenced the importance of integrating both pose and motion features as suggested by experiments throughout Chapter 4. An evaluation of the architecture on a dataset of hand gestures is reported in Appendix D. In Parisi et al. (2014b), we considered a wider number of body features to perform gestures. The representation of gestures is hand-independent and gestures using both hands are also considered.

For recognizing gestures in our multimodal system, we used an extended version of Parisi et al. (2014a) for learning gestures from depth map videos using growing self-organizing networks. Furthermore, for each predicted label we also estimate a confidence value that expresses the degree of belief that the prediction is correct based on a set of predictions over a given time window.

#### Feature Extraction

Hand motion from depth images was extracted to represent gestures as hand-independent motion sequences. The set of gestures used in our robotic scenario are shown in Fig. 7.15. To encode motion patterns, only the motion information of

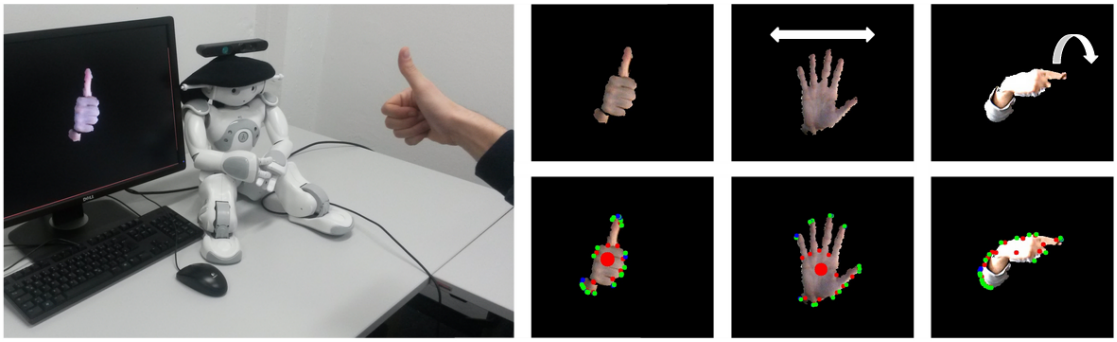


Figure 7.13: Example of hand segmentation and pose estimation with SHAPE for static and dynamic gestures (Parisi et al., 2014a).

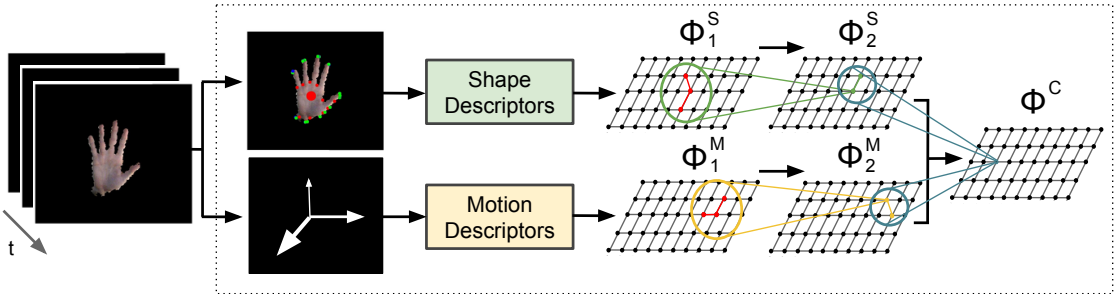



Figure 7.14: FINGeR pipeline for the hierarchical SOM-based clustering of encoded gestures with the SHAPE algorithm. Pose and motion properties are processed by two different streams, $\Phi^S$ and $\Phi^M$ respectively. Synchronized multi-cue representations are subsequently combined by $\Phi^C$ (Parisi et al., 2014a).

the most salient hand performing a gesture was taken into account (Parisi et al., 2014b). In case that both hands are used, the type of interaction between the hands is considered, i.e. *physical* if the two hands overlap, or *symmetric*, if they follow the same (mirrored) behavior. We consider a set of motion descriptors for a given set of tracked body joints, i.e. hands and head. For each frame $i$, the gesture feature vectors were of the form $\mathbf{m}_i = (s_i, \mathbf{v}_i, \varphi_i, h_i, \lambda_i)$, where $s_i$ is the hand interaction type, $\lambda_i$ is the annotated gesture label, $\mathbf{v}_i$ is the hand 3D motion intensity in terms of pixel difference from consecutive frames, $\varphi_i$ is the hand angle with respect to the $y$ axis in the image plane, and $h_i$ is the distance from the head.

Training videos were recorded with an ASUS Xtion depth sensor operating at 30 frames per second, from which we estimated the 3D skeleton model using the OpenNI/NITE framework. To attenuate noise, we computed the median value for each joint every 3 frames, resulting in a total of 10 feature vectors per second. These vector sequences are then clustered by a hierarchical learning architecture to obtain a representation of prototype gestures from a set of training samples.

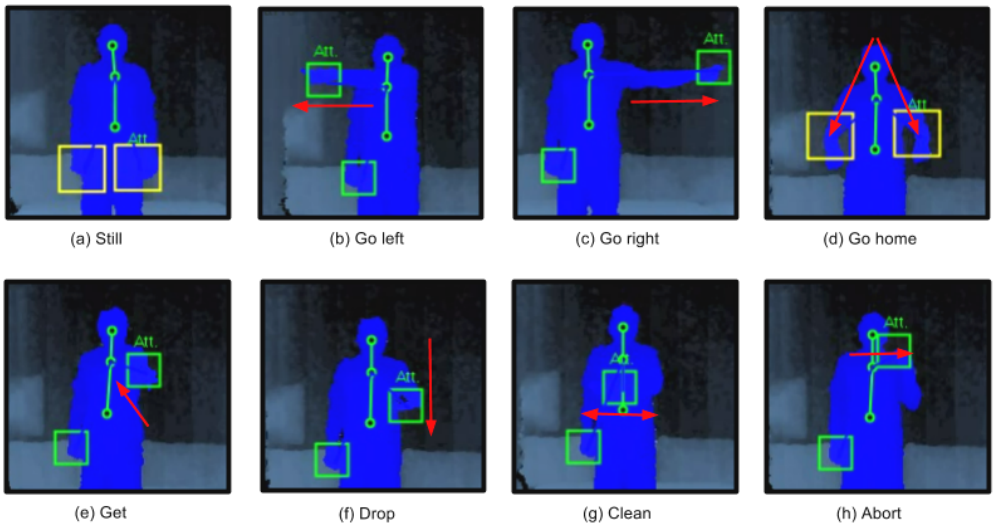



Figure 7.15: Gestures used as advice in the robotic scenario. Red arrows represent the hand movement performed to advice the robot. The motion from the most salient hand is used to estimate the motion vector. In case that both hands are used, the type of hand interaction is considered (details in the text). Since gesture labels are seamlessly predicted from depth map video sequences, we add the label *still* to indicate no advice at that moment (Cruz et al., 2016b).

## Learning Architecture

Our learning model consists of two hierarchically arranged GWR networks (Marsland et al., 2002) that incrementally obtain generalized representations of sensory inputs to learn latent spatiotemporal structure. Hierarchical learning is carried out by training the higher-level network with neuron activation trajectories from the lower-level network (see Chapter 4). The network in the first layer receives the sequence of vectors $\mathbf{m}_i$ as input. The network in the second layer is trained with neural activation trajectories from the first layer. These trajectories are obtained by computing the best-matching neurons of the input sequence $\mathbf{x}_i$ with respect to the trained network with $N$ neurons, so that a set of trajectories is given by

$$\Omega(\mathbf{x}_i) = \{\mathbf{w}_{b(\mathbf{x}_i)}, \mathbf{w}_{b(\mathbf{x}_{i-1})}, \mathbf{w}_{b(\mathbf{x}_{i-2})}\}, \tag{7.10}$$

with $b(\mathbf{x}_i) = \arg\min_{i \in N} \|\mathbf{x}_i - \mathbf{w}_j\|$. After the training of the higher level network is completed, each neuron will encode a sequence-selective gesture segment from 3 consecutive frames. This mechanism allows to obtain specialized neurons coding the spatiotemporal structure of the input. For classification purposes, neurons created in this second layer are attached to gesture labels obtained from the training set. During training of the GWR networks, we attach labels to neural activation trajectories as discussed in Section 4.3. The training parameters and number of neurons created after the training session are shown in Table 7.3.

In the hierarchical architecture, a predicted label is returned every 3 frames in a sliding window scheme. We considered the last 5 observations and computed the

Table 7.3: Training parameters for GWR hierarchical learning

| Parameters | Network Layer 1, 2 |
|---|---|
| Activation threshold | $a_T = \{0.85, 0.65\}$ |
| Firing threshold | $f_T = 0.01$ |
| Firing counter | $\tau_b = 0.3$, $\tau_n = 0.1$ |
| Learning rates | $\epsilon_b = 0.1$, $\epsilon_n = 0.01$ |
| Maximum edge age | 200 |
| Training epochs | 100 |
| N. of neurons after training | $\{337, 316\}$ |

statistical mode that returns the most frequent value in a set. Given the set of predictions $\Lambda^V$ and denoting $N$ as the number of occurrences of the mode within $\Lambda^V$, the confidence value is then defined as $\gamma^V = N/|\Lambda|$, yielding a maximum confidence value of 1 and a minimum of 0.2. Since we processed 10 feature vectors per second and we compute the mode of the last 5 predictions, our system returns a predicted label $\lambda^V$ and a confidence value $\gamma^V$ for a window of 7 frames (0.7 seconds).

### 7.3.5 Audiovisual Integration

Our integration function relates the predicted advice classes and confidence pairs from uni-sensory input, respectively denoted as $(\lambda^A, \gamma^A)$ for audio and $(\lambda^V, \gamma^V)$ for vision. The integrated predicted label $\lambda^I$ is calculated according to the highest confidence value:

$$\lambda^I = \underset{\lambda}{\operatorname{argmax}}\ \gamma(\lambda). \tag{7.11}$$

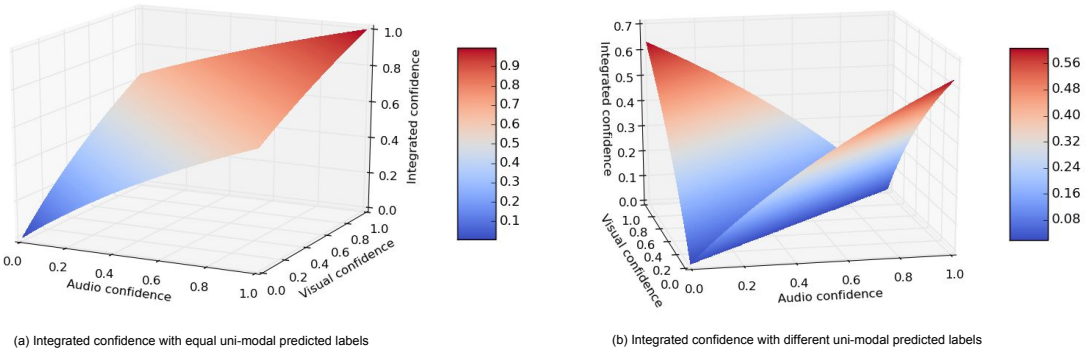
In other words, if the auditory and visual labels $\lambda^A$ and $\lambda^V$ are different, then the integrated label $\lambda^I$ takes the value from the modality which has the biggest confidence value. The integrated confidence value is computed by the function:

$$\gamma^I = \ln(1 + \phi), \tag{7.12}$$

where $\phi$ is a time-varying parameter which depends on each label $\lambda$ and confidence value $\gamma$. We refer to this parameter as the *likeliness* parameter, obtained according to the following equation:

$$\phi = \begin{cases} \gamma^A + \gamma^V & \text{if } \lambda^A = \lambda^V \\ |\gamma^A - \gamma^V| & \text{if } \lambda^A \neq \lambda^V \end{cases} \tag{7.13}$$

Therefore, if the labels $\lambda^A$ and $\lambda^V$ are the same, then the confidence value $\gamma^I$ is computed using $\phi = \gamma^A + \gamma^V$ in order to strengthen the integrated confidence level over the prediction made from both modalities. Instead, if the labels $\lambda^A$ and $\lambda^V$ are different, then the integrated confidence value $\gamma^I$ is computed using




(a) Integrated confidence with equal uni-modal predicted labels

(b) Integrated confidence with different uni-modal predicted labels

Figure 7.16: Confidence values used in the neural network-based associative architecture. While in (a) the corresponding output labels for audio and visual modalities are the same, in (b) they are different. During the training, we use a $20 \times 20$ grid for each modality, whereas for the test we use an equal distribution with a $100 \times 100$ grid (Cruz et al., 2016b).

$\phi = |\gamma^A - \gamma^V|$ in order to decrease the confidence level. This function yields an integrated confidence value $\gamma^I \in [\ln(1), \ln(3)] = [0, 1.0986]$. We use a unity-base normalization to rescale the range of confidence between 0 and 1:

$$\gamma^I = \frac{\gamma^I - min(\Gamma)}{max(\Gamma) - min(\Gamma)}. \tag{7.14}$$

where $\Gamma$ is the set of all possible confidence values $\gamma^I$. Fig. 7.16 shows the integrated confidence values when the predicted auditory and visual labels are the same (a) and different (b).

### Associative Learning

To implement the proposed integration model, we develop an associative neural architecture with a complex-valued quadratic neuron (Georgiou, 2006) that defines a two-dimensional grid on the output space as presented in (Georgiou and Voigt, 2013). For an input vector $X \in \mathbb{C}^n$, the scalar complex output is $y = X^* A X$, where $A \in \mathbb{C}^{n \times n}$ is the weight matrix and $X^*$ denotes the conjugate transpose. The output can be written as the summation of the individual terms that involve the components of $X$ and $A$:

$$y = \sum_{j=1}^{n} \sum_{k=1}^{n} \bar{x}_j x_k a_{jk}. \tag{7.15}$$

The gradient descent learning rule that minimizes the mean-square error is:

$$\triangle A = \alpha \varepsilon \bar{X} X^T, \tag{7.16}$$

where $\alpha$ is a small real-valued learning rate. For a given input vector $X$, the desired output $Y$ to be used in the learning algorithm is defined as the nearest intersection

point of the grid lines of the complex plane. In practice, a function $\Psi$ is defined which rounds to the nearest integer for grid lines spaced at a fixed distance $\delta$ in both directions:

$$\Psi(Y) = \frac{round(\delta Re(Y))}{\delta} + i\frac{round(\delta Im(Y))}{\delta}. \tag{7.17}$$

This function creates a virtual grid where the output snaps onto the nearest grid corner. The training algorithm is as follows:

0. Initialize the weights of the neuron with random values,
1. Compute $Y$,
2. Compute $d = \Psi(Y)$,
3. Update the weights of the neuron using Eq. 7.16.

At each iteration, the steps (1) to (3) are carried out for all the input vectors, so that a cluster in the input space will map to a similar region in the output space due to the continuity of the activation function. The stop criterion can be a fixed number of iterations, a decreasing learning rate, or a given minimum mean-square error over all inputs.

### 7.3.6 Experimental Results

For our experiments, we implemented the robotic domestic scenario described in Section 7.3.2. We recorded clips of advice from a parent-like trainer for all advice classes, including speech and gestures with four repetitions for each class. At recognition time, our goal was to predict the gesture label from novel audiovisual sequences $(\lambda^A, \lambda^V)$ and compute the confidence values $(\gamma^A, \gamma^V)$ that expressed how reliable these predictions are. After the independent processing each modality, audiovisual inputs were integrated using our neural architecture to compute the integrated gesture class $\lambda^I$ with confidence $\gamma^I$. We used a grid of $20 \times 20$ points for training and a subsequent validation grid of $100 \times 100$ points obtaining an average quantization error $e_q(n) = 0.05984$ computed as $e_q(n) = x_q(n) - x(n)$, where $x_q(n)$ and $x(n)$ are the sample sequences of the validation set and the training set respectively.

When working autonomously in the domestic scenario, the robot selects the actions using $\epsilon$-greedy action selection policy with $\epsilon = 0.1$. We used interactive advice probability of 0.3 since it has been shown to be effective and small enough (Cruz et al., 2016b). After the integration, we used different confidence levels to verify whether small confidence values benefit the learning scenario. Therefore, we considered $\gamma^I > \theta_{min}$ with $\theta_{min}$ being the minimum confidence threshold to be considered as a valid advice. In the case that the advice did not accomplish this minimal condition, the next action was selected through the aforementioned $\epsilon$-greedy policy. We tested different thresholds $\theta_{min} \in \{0.0, 0.25, 0.5, 0.75\}$, observing that in

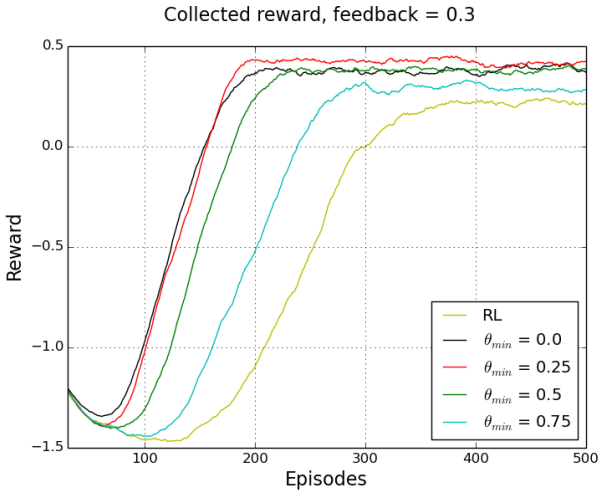



Figure 7.17: Integrated rewards with different thresholds of minimal confidence level. The best performance is observed with $\theta_{min} = 0.25$ depicted in red. Autonomous RL is shown as a baseline in yellow (Cruz et al., 2016b).

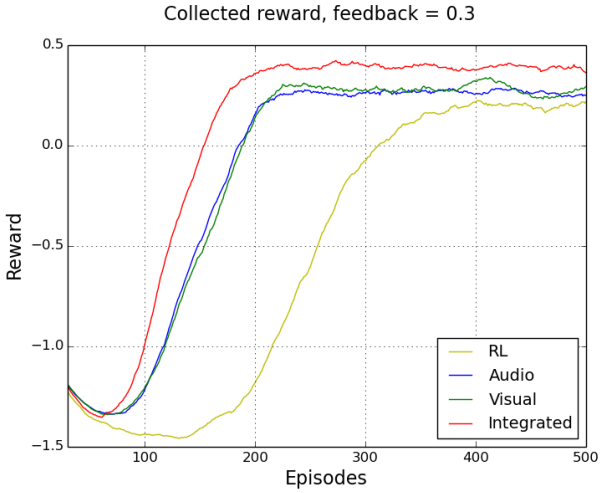



Figure 7.18: Collected rewards with advice from audio and visual modalities are shown in blue and green respectively. Autonomous RL is shown as a baseline in yellow. Working with advice from the multimodal integration approach, the IRL agent is able to collect faster and greater reward in comparison to individual advice approaches (Cruz et al., 2016b).

general IRL works better with $\theta_{min} = 0.25$. Fig. 7.17 shows the average convoluted rewards for those $\theta_{min}$ values using 100 agents over 500 training episodes.

Finally, we used a fixed threshold $\theta_{min} = 0.25$ during the learning process to compare unimodal and multimodal advice in the IRL scenario. In the unimodal IRL approach, collected rewards were close to each other in terms of the time required for convergence (more than 200 episodes) as well as the maximal reward value (approximately 0.3). On the other hand, the multimodal IRL approach

using both sensory inputs obtained the same level with respect to the unimodal approaches in fewer episodes (in this case, less than 200 episodes) and converged to a greater reward (approximately 0.4). Therefore, the IRL performance benefits from the integrated information, where greater rewards are accumulated faster in comparison to the use of unimodal modules. Fig. 7.18 shows the average collected reward over 500 training episodes for the uni- and multimodal learning procedure.

Together, these results show that multimodal integration leads to a better performance of interactive reinforcement learning with the robot being able to learn faster with greater rewards compared to unimodal scenarios. Experiments in our multimodal IRL scenario were conducted in an off-line scheme. Therefore, future work directions should consider experiments accounting for on-line interactions. Furthermore, experiments should also consider a wider number of parent-like trainers with different teaching characteristics.

## 7.4 Summary

In this chapter, we proposed the use of multimodal systems for enhancing human-robot interaction and triggering sensory-driven robot behavior in dynamic environments. In particular, we presented two robotic scenarios. The first scenario consisted of a humanoid robot for fall detection in a home-like environment, for which we used audiovisual cues to track a user performing daily activities, while a neural network architecture was responsible for detecting abnormal user behavior from action sequences captured with a depth sensor. Experiments have shown that multimodal processing is crucial to complement occasionally unavailable modalities, thereby providing a more robust perceptual experience to the robot. In the second scenario, we proposed the integration of speech and body gestures for providing trainer-like feedback to a learning agent in an interactive reinforcement learning task, thereby improving the overall learning performance. Our architecture integrates dynamic audiovisual patterns for a more natural trainer-like learning procedure, also accounting for possible conflicts in terms of contradictory predicted feedback classes or classes predicted with low confidence. Experiments have shown that multimodal feedback significantly enhances the agent's performance while learning the task of cleaning a table.

In both scenarios, robot motor control was triggered by the interplay of auditory and visual cues, showing that multisensory integration can be advantageous for a variety of reasons. One is that certain types of information can only be gleaned from some modalities and not from others. This is the case in our fall detection scenario in which verbal information is only available in the auditory modality. A second reason why multisensory integration can be useful is that it provides redundancy which can help improve accuracy and disambiguate. For instance, we exploit this aspect of multimodal learning in our first scenario when we integrate segmentation from depth perception and sound cues to estimate the position of a person in the environment. This would be much harder from either modality alone. Finally, it can be useful to employ another type of sensor even if the information

gleaned through it could be provided by a different sensor in principle: sometimes one modality just provides information simply in a more appropriate form, as exemplified by our use of the Nao's sonar sensors for obstacle detection which would be possible, at greater computational cost, using just color vision or depth perception. As future work, we aim to design a usability study to evaluate the systems in a real-world setting, for instance by studying the users' acceptance of the agents in terms of overall performance, human-robot communication, timing, and task sequence (Torta et al., 2014).

# Chapter 8

# Conclusion

Understanding others' actions plays a crucial role in our everyday lives. Human beings are able to reliably discriminate a series of socially relevant cues from body motion, with this ability being supported by highly skilled visual perception and other modalities. The main goal of this thesis is the modeling of artificial learning architectures for action perception with focus on the development of multimodal action representations. As a modeling foundation to address our research question, we focus on hierarchies of self-organizing neural networks motivated by experience-driven cortical organization.

## 8.1 Thesis Summary

We presented a number of neural network architectures that develop robust spatiotemporal representations for the task of multimodal action perception. As a starting point, we proposed a set of neurobiologically motivated architectures consisting of hierarchically-arranged network layers for processing action cues in the visual domain in terms of body posture and motion features. On this basis, we investigated the use of hierarchical self-organizing learning for the development of congruent multimodal action representations. In particular, we proposed a model where multimodal representations emerge from the co-occurrence of auditory and visual stimuli via the learning of associative connections between unimodal representations, yielding the bidirectional retrieval of audiovisual patterns.

In the spirit of hierarchical spatiotemporal processing, we proposed an extension of a growing self-organizing network equipped with recurrent connectivity, showing that this novel model accounts for the learning of robust action-label mappings also in the face of occasionally absent or even contradictory action class labels during the training phase. We demonstrated how the same recurrent neural network mechanism can deal with both action recognition and body motion assessment in real time.

Finally, we reported on two robot experiments of multimodal perception, one focused on a fall detection scenario and the other was set in a multimodal interactive reinforcement learning task. Our experiments showed that the integration of

multiple modalities significantly improves performance with respect to unimodal approaches for sensory-driven robot behavior.

## 8.2 Discussion

The research presented in this thesis considers interdisciplinary aspects of action perception and its underlying neural mechanisms with the aim to develop learning architectures for multimodal action processing. In the following sections, we discuss important modeling aspects of our neural network architectures and the results obtained, as well as analogies and limitations with respect to biological findings.

### Neurocognitive Architectures for Multimodal Integration

A variety of studies has shown the ability of the brain to integrate multimodal information for providing a coherent perceptual experience (Stein and Meredith, 1993; Ernst and Bülthoff, 2004; Stein et al., 2009). Specifically for the integration of audiovisual stimuli, neurophysiological studies have evidenced strong links between the areas in the brain governing visual and language processing for the formation of multimodal perceptual representations (Foxe et al., 2000; Raij et al., 2000; Belin et al., 2000, 2002; Pulvermüller, 2005). However, the question of how to develop artificial models that efficiently process and bind multimodal information has remained an issue to be investigated (Ursino et al., 2014).

The development of associations between co-occurring stimuli for multimodal binding has been strongly supported by neurophysiological evidence (Fiebelkorn et al., 2009; Ursino et al., 2014). Similar to Vavrečka and Farkaš (2014) and Morse et al. (2015), we argue that the co-occurrence of sensory inputs is a sufficient source of information to create robust multimodal representations with the use of associative links between unimodal representations that can be incrementally learned in an unsupervised fashion. However, in contrast to previous models focused on the development of object–word mappings, we focus on the development of associative links between action labels and visual actions, which have high spatial and temporal variance, thereby requiring a processing architecture that accounts for the generalization of inputs at different spatiotemporal scales.

From a neurobiological perspective, neurons selective to actions in terms of complex biological motion have been found in a wide number of brain structures (Giese and Rizzolatti, 2015). An example is the STS, which is thought to be an associative learning device for linking different unimodal perceptual representations, and consequently crucial for social cognition (Allison et al., 2000; Adolphs, 2003; Beauchamp, 2005; Beauchamp et al., 2008). It has been shown that different regions in the STS are activated by naturally occurring, highly correlated action features, such as pose, motion, the characteristic sound of an action (Beauchamp et al., 2004; Barraclough et al., 2005) and linguistic stimuli (Belin et al., 2002; Wright et al., 2003; Stevenson and James, 2009).

In Chapter 5, we proposed a simplified computational model that learns to integrate audiovisual patterns of action sequences. Our model incrementally learns a set of associative connections in a self-organized manner to bind unimodal representations from co-occurring multisensory inputs. Therefore, neurons in the associative layer are tuned to multimodal action snapshots in terms of action-word mappings. The focus of our study was the self-organizing development of associative connections between visual and auditory action representations. For audiovisual stimulation, neurons in the posterior STS showed greater response to multimodal stimuli than to unimodal ones, with these multimodal responses being greater than the sum of the single unimodal responses. This principle, referred to as superadditivity, has not been observed for auditory-tactile stimulation (Beauchamp et al., 2008), suggesting that multimodal patterns are integrated in a principled way according to modality-specific properties. The modeling of neurobiologically observed principles underlying audiovisual integration in the STS for speech and non-speech stimuli, such as superadditivity (Calvert et al., 2000), spatial and temporal congruence (Bushara et al., 2001; Macaluso et al., 2004), and inverse effectiveness (Stevenson and James, 2009), was out of the scope of this thesis and will be subject to future research.

Based on the principle of learning associative connections from co-occurring inputs, it is possible to extend the development of associative patterns beyond the audiovisual domain. For instance, several neurophysiological studies have evidenced strong interaction between the visual and motor representations, more specifically including the STS, parietal cortex, and premotor cortex (see Giese and Rizzolatti (2015) for a recent survey), with higher activation of neurons in the motor system for biomechanically-plausible, perceived motion sequences (Miller and Saygin, 2013). From the perspective of our model, we could think of emerging associative connections between auditory, visual, and motor representations in terms of the self-organizing binding of temporally correlated activations. However, while our architecture scales up to a larger number of modalities, it does not account for crossmodal learning aspects, e.g. in an embodied robot perception scenario where motor contingencies influence audiovisual mappings (Morse et al., 2015). Consequently, the extension of our model in such a direction would require additional mechanisms for the crossmodal learning of spatiotemporal contingencies built on the basis of modality-specific properties.

## Self-Organizing Hierarchies of Networks

Hierarchies may provide a convenient trade-off in terms of invariance-selectivity by decomposing a complex task in a hierarchy of simpler ones (Poggio and Smale, 2003). From a computational perspective, a hierarchical structure has the advantage of increased computational efficiency by sharing functionalities across multiple levels, e.g., low-level networks represent a dictionary of features that can be shared across multiple tasks. The proposed hierarchical learning architectures yield progressively specialized neurons encoding latent spatiotemporal dynamics of the input. Neurons in higher-level layers will encode prototype sequence-selective

snapshots of visual input, following the assumption that the recognition of actions must be selective for temporal order (Giese and Poggio, 2003; Hasson et al., 2008). In Chapter 4 and 5, the temporal processing of features was explicitly modeled in terms of neurons in higher-level layers computing the concatenation of neural activation trajectories from lower-level layers, which increases the dimensionality of neural weights along the hierarchy. This issue was addressed in Chapter 6, where we proposed a novel temporal extension of the GWR with context learning (Strickert and Hammer, 2005) and a Gamma Memory model (de Vries and Príncipe, 1992; Estévez and Vergara, 2012), showing that hierarchically-arranged GWR networks with recurrent connections can account for the learning of action features with increasingly larger spatiotemporal receptive fields.

A hierarchical organization is consistent with neurophysiological evidence for increasingly large spatiotemporal receptive windows in the human cortex (Taylor et al., 2015; Hasson et al., 2008; Lerner et al., 2011), where simple features manifest in low-level layers closest to sensory inputs, while increasingly complex representations emerge in deeper layers. Specifically for the visual cortex, Hasson et al. (2008) showed that while early visual areas such as the primary visual cortex (V1) and the motion-sensitive area (MT+) yield higher responses to instantaneous sensory input, high-level areas such as the STS were more affected by information accumulated over longer timescales ($\sim$ 12 seconds). This kind of hierarchical aggregation is a fundamental organizational principle of cortical networks for dealing with perceptual and cognitive processes that unfold over time (Fonlupt, 2003).

Motivated by the process of input-driven self-organization exhibited by topographic maps in the cortex (Nelson, 2000; Willshaw and von der Malsburg, 1976; Miikkulainen et al., 2005), we proposed a series of learning architectures encompassing a hierarchy of self-organizing networks. Growing neural networks the ability to dynamically change their topological structure through competitive Hebbian learning (Martinetz, 1993) and incrementally match the distribution of the data in input space (see Chapter 3). Different from other incremental models of self-organization that create new neurons at a fixed growth rate (e.g. Fritzke 1995, 1997), GWR networks (Marsland et al., 2002) create new neurons whenever the activity of well-trained neurons is smaller than a given threshold. This mechanism creates a larger number of neurons at early stages of training and then tunes the weights through subsequent training epochs. While the process of neural growth of the GWR algorithm does not resemble biologically plausible mechanisms of neurogenesis (e.g., Eriksson et al. 1998; Gould 2007; Ming and Song 2011), it is an efficient learning model exhibiting a computationally convenient trade-off between adaptation to dynamic input and learning convergence.

The two parameters modulating the growth rate of the network are the activation threshold and the firing counter threshold. The activation threshold establishes the maximum discrepancy (distance) between the input and its best-matching neuron in the network, with larger values of the threshold yielding a smaller discrepancy. The firing counter threshold is used to favour the training of recently created neurons before creating new ones. Intuitively, the average discrepancy between the input and the network representation should decrease for

a larger number of neurons. On the other hand, there is no such straightforward relation between the number of neurons and the classification performance. This is because the classification process consists of predicting the label of novel samples by retrieving attached labels to the inputs' best-matching neurons, irrespective of the actual distance between the novel inputs and the selected neurons (see Section 5.3). Therefore, convenient threshold values should be chosen by taking into account the distribution of the input and, in the case of a classification task, the classification performance.

## Action Features and Representations

For the processing of action features in Chapters 4, 5, 6, and 7, we rely on the extraction of a simplified 3D skeleton model from which we compute relevant cues describing body pose and motion while maintaining a low-dimensional feature space. The skeleton model estimated by OpenNI, although not anatomically faithful, provides a convenient representation from which it is possible to extrapolate actor-independent action dynamics. The use of such models is in line with biological evidence demonstrating that human observers are very proficient in recognizing and learning complex motion underlying a skeleton structure (Jastorff et al., 2006; Hiris, 2007). These studies show that the presence of a holistic structure improves the learning speed and accuracy of action patterns, also for non-biologically relevant motion such as artificial complex motion patterns. On the other hand, skeleton models may be susceptible to sensor noise and situations of partial occlusion and self-occlusion (e.g. caused by body rotation) for which body joint values may be noisy or missing. In Chapter 6, we proposed a neural architecture able to learn spatiotemporal action features from depth images with segmented body silhouettes, thereby addressing the issue of noisy skeletons. On the one hand, in this case we rely on the correct segmentation of body shape from depth-map image sequences. On the other hand, approaches for extracting action features from cluttered environments have been shown to be either computationally expensive or they require large amounts of training data (Guo et al., 2016), thus they are not ideal for real-world scenarios (see Chapter 7).

Our proposed neural models for action perception create prototype action representations based on statistically significant features presented during the training process. This process allows to generalize spatiotemporal properties of the training set to classify novel samples and yields invariance to scale and position of the visual stimuli. Our recognition scheme for action sequences is in line with a number of studies demonstrating that action discrimination is selective to temporal order (Bertenthal and Pinto, 1993; Giese and Poggio, 2003; Jastorff et al., 2006). These action representations are view dependent, i.e., if the perspective of the sensor or the orientation of the person with respect to the sensor change, actions may not be reliably recognized. This is not in contradiction with biological studies showing that biological motion recognition is strongly dependent on stimulus view and orientation. Sumi (1984b) as well as Pavlova and Sokolov (2008) demonstrated that action recognition is impaired by biological motion stimuli being upside-down

or rotated with respect to the image plane. Similarly, Jastorff et al. (2006) found that learned visual representations seem to be highly orientation-dependent, i.e., discrimination performance increased only when the test patterns presented the same orientation as in the training. Therefore, view-dependent action recognition is consistent with the idea that biological motion perception is based on the matching of learned two-dimensional patterns. On the other hand, there is a strong motivation to develop artificial systems that account for view-independence responses, e.g., achieved by means of 3D internal models (Sumi, 1984a). In our implementation of the GNG and the GWR algorithms, we used the Euclidean distance as a metric to compute the distance of prototype neurons and neuron trajectories from the current input. Giese et al. (2008) investigated perceptual representations of full-body motion finding motion patterns that reside in perceptual spaces with well-defined metric properties. They conducted experiments with 2D and 3D joints of prototype trajectories with results implying that perceptual representations of complex motion patterns closely reflect the metric of movements in the physical world. Although more precise neural mechanisms that implement distance computation remain to be explored, we can assume that the Euclidean distance is an adequate metric to compare articulated movement patterns.

In our models for processing actions in terms of pose-motion features, we have assumed that the pose and the motion pathways do not interact before the stage of integration. This is a strong simplification with respect to biological mechanisms, where the two streams comprise interactions at multiple levels (Felleman and Van Essen, 1991). From a computational perspective, it would be interesting to investigate the interplay of pose-motion cues and recognition strategies when one of the two cues is suppressed. Our neural architectures require that both the pose and motion samples are available for parallel processing and integration. However, Tyler and Grossman (2011) demonstrated that observers can shift between pose- and motion-based strategies, depending on the available cue. In other words, suppressing one of the cues does not fully impair action perception. In line with this assumption, we could extend our models with inter-lateral connections so that neurons from distinct pathways can co-activate in the presence of single-cue input. This mechanism would require network layers to be equipped with symmetric, inter-network references that link prototype neurons in different stream populations, and that enable the computing of activation trajectories in both pathways when only neurons from one pathway are activated. In this setting, the dynamics of learning and neural mechanisms of integration can be investigated.

## 8.3 Future Work

Our proposed neural network architectures for action perception are based on a set of strong simplifications. In this section, we discuss two main research directions aimed to address some important shortcomings.

## Attention as a Modulator of Action Perception

In this thesis, we focused on feedforward hierarchical learning mechanisms of action recognition and assessment. In Chapter 6, we introduced recurrent connectivity in network layers to process sequential visual input with increasingly larger spatiotemporal receptive fields as strongly supported by biological findings (Taylor et al., 2015; Hasson et al., 2008; Lerner et al., 2011). However, anatomical and neurophysiological studies have shown that the visual cortex exhibits significant feedback connectivity between different cortical areas (Felleman and Van Essen, 1991; Salin and Bullier, 1995). In particular, action perception demonstrates strong top-down modulatory influences from attentional mechanisms (Thornton et al., 2002) and higher-level cognitive representations such as biomechanically plausible motion (Shiffrar and Freyd, 1990). More specifically, audiovisual spatial attention allows animals and humans to process relevant environmental stimuli while suppressing irrelevant information. Therefore, attention as a modulator in action perception is also desirable from a computational perspective, thereby allowing the suppression of uninteresting parts of the visual scene and thus simplifying the detection of human motion in cluttered environments (e.g., in the robot-human assistance scenario presented in Chapter 7).

Several brain areas and neural mechanisms have been identified to be involved in the processing of spatial attention during perception (Driver, 2001). For instance, the midbrain area superior colliculus (SC) plays a crucial role in spatial attention in terms of target selection and estimating motor consequences such as eye and head saccades (Krauzlis et al., 2013). The integration of audiovisual stimuli in the SC has been extensively investigated from a neurophysiological perspective (Ursino et al., 2014), with different computational approaches modeling the integration of multiple perceptual cues for triggering spatial attention in line with neurobehavioral evidence, e.g. with the use of a self-organizing neural architecture (Bauer et al., 2015). The SC is connected to higher cortical areas such as the visual and the auditory cortex, both able to process information events that unfold over larger temporal time scales such as the visual recognition of body actions and speech. Top-down connectivity from cortical areas is used by the SC to modulate attentional shifts.

Consequently, future work directions may include the development of a cortico-collicular architecture aimed at modeling crossmodal attention and accounting for the interplay between the SC and cortical processing. This architecture could extend the computational model of multimodal integration performed by the SC proposed by Bauer et al. (2015) by adding cortical feedback and recurrent self-organizing networks for the integration of inputs in the spatiotemporal domain as proposed in Chapter 6. For a biologically plausible model of crossmodal learning, neural network models should account for the modeling of multimodal integration principles of co-occurring stimuli such as superadditivity (Calvert et al., 2000), spatial and temporal congruence (Bushara et al., 2001; Macaluso et al., 2004), and inverse effectiveness (Stevenson and James, 2009). Multimodal representations in the SC may serve as input for a cortical visual-auditory integration model, using

recurrent self-organizing networks to learn inherent spatiotemporal structure, e.g. recognition of actions from visual and auditory cues. The output from cortical areas can be used as feedback for the SC model, thereby modulating attentional shifts as an interplay between bottom-up and top-down processing mechanisms.

This architecture would aim to model the underlying neural mechanisms of crossmodal attention in terms of cortico-collicular interaction with the aim of reproducing behavioral responses supported by psychological studies on attentional shifts from audiovisual stimuli. Furthermore, this model could be embedded in a robot to test whether crossmodal attention effectively improves action perception.

## Life-Long Learning of Action Representations

The neural network architectures proposed in Chapters 4, 5 and 6 as well as other similar hierarchical models are designed for learning a batch of training actions, thus implicitly assuming that a training set is available (e.g. Giese and Poggio 2003; Guo et al. 2016). Ideally, this training set contains all necessary knowledge that can be readily used to predict novel samples in a given domain. However, this training scheme is not suitable for more natural scenarios where an artificial agent should incrementally process a set of perceptual cues as these become available over time. Therefore, life-long learning is considered to be essential for cognitive development and plays a key role in autonomous robotics for the progressive acquisition of knowledge through experience and the development of meaningful internal representations during training sessions (Zhou, 1990; Lee, 2012).

It has been argued that hierarchical predictive models with interactions between top-down predictions and bottom-up regression may provide a computational mechanism to account for the learning of dynamic input distributions in an unsupervised fashion (Jung et al., 2015). Predictive coding (Rao and Ballard, 1999; Huang and Rao, 2011) has been widely studied for understanding many aspects of brain organization and, in particular, it has been proposed that the visual cortex can be modeled as a hierarchical network with reciprocal connections where top-down feedback connections from higher-order cortical areas convey predictions of lower-order neural activity and bottom-up connections carry the residual prediction errors. Tani and Nolfi (1999) and Tani (2003) proposed that the generation and recognition of sensory-motor patterns for on-line planning in a robot learning scenario can be obtained by using recurrent neural network models extended with prediction error minimization. However, neural network models that implement a predictive learning scheme to achieve life-long learning have not been yet fully investigated.

With the use of recurrent self-organizing as proposed in Chapter 6, life-long learning can be developed in terms of prediction-driven neural dynamics with action representations emerging from the interplay of feedforward–feedback connectivity in a self-organizing hierarchy. In our proposed architecture, the growth of the networks is modulated by their capability to predict neural activation sequences from the previous network layer. The ability of the architecture to correctly predict action labels from incoming sequence may be then used to modulate neural activ-

ity along the hierarchy. More specifically, feedback connectivity from the symbolic layer containing action labels could have modulatory effects on the growth rate of lower-level networks so that a sufficient number of prototype neurons are created as a dictionary of primitives subsequently used to learn spatiotemporal statistics of the input. This mechanism may be employed to modulate the amount of learning necessary to adapt to the dynamic input distribution and develop robust action representations. Convenient threshold values should be chosen so that the layers adapt to dynamic input (yielding smaller prediction errors) while showing convergence with stationary input. Additionally, other important principles that play a role in life-long learning such as the influence of reward-driven motivational and attentional functions (Ivanov et al., 2012) can be taken into account and will be subject to future research.

## 8.4 Conclusion

In conclusion, this thesis contributes to the knowledge about multimodal action representations which can emerge from deep neural network self-organization. Studies on biological motion perception have evidenced the hierarchical processing of stimuli with increasing complexity of representation, together with the development of topographic maps driven by the distribution of the input as a common feature of cortical networks. In the light of these findings, rudimentary models of experience-driven self-organization can be extended to the design of neural network architectures for body motion processing.

We proposed a set of neural network architectures for the learning of action representations from videos. Our approach consists of hierarchically-arranged self-organizing networks processing action cues in terms of body posture and motion features. We investigated the use of self-organizing learning for the development of congruent multimodal action representations from auditory and visual stimuli. Furthermore, we proposed a novel temporal extension of a self-organizing network equipped with recurrent connectivity for dealing with time-varying patterns. Reported experiments showed that deep self-organizing architectures yield robust action representations, exhibiting comparable performance to state-of-the-art results also in the case of sensory uncertainty and conflicts. Additionally, we showed how the same recurrent neural network mechanism can deal with both action recognition and body motion assessment in real time.

Although a full understanding of the biological mechanisms for multimodal action perception remains to be determined, we proposed a set of neurally inspired computational approaches as a basis for modeling the development of higher levels of cognition in artificial systems.

# Appendix A

# List of Abbreviations

**ANN** artificial neural network.

**ASR** automatic speech recognition.

**BMU** best-matching unit.

**BPTT** backpropagation through time.

**DoG** difference of Gaussians.

**DTW** dynamic time warping.

**F5** ventral premotor cortex.

**FOV** field of view.

**GNG** growing neural gas.

**GWR** growing when required.

**HMM** hidden Markov model.

**HOG** histogram of oriented gradient.

**HRI** human-robot interaction.

**IID** interaural intensity difference.

**IT** inferior temporal cortex.

**IRL** interactive reinforcement learning.

**kNN** k-nearest neighbor.

**KO** kinetic occipital cortex.

**LGN** lateral geniculate nucleus..

**MST** medial superior temporal cortex.

**MT** middle temporal cortex.

**MT+** motion-sensitive area.

**MTG** middle temporal gyrus.

**NG** neural gas.

**NN** neural network.

**OF** optic flow.

**OSS-GWR** online semi-supervised growing when required.

**PCA** principal component analysis.

**pSTS** posterior superior temporal sulcus.

**RL** reinforcement learning.

**ROS** robot operating system.

**S-GWR** supervised growing when required.

**SC** superior colliculus.

**SOM** self-organizing map.

**SSL** sound source localization.

**STG** superior temporal gyrus.

**STS** superior temporal sulcus.

**SVM** support vector machine.

**TDOA** time difference of arrival.

**TKM** temporal Kohonen map.

**V1** primary visual cortex.

**V2** secondary visual cortex (prestriate cortex).

**V4** visual area in the extrastriate visual cortex.

**VQ** vector quantization.

# Appendix B

# Supplementary Algorithms

**Algorithm 4** Growing Neural Gas (Fritzke, 1995)

1: Start with a set $N$ of two nodes at random positions $w_a$ and $w_b$ in the input space.
2: Apply an input signal $\xi$ according to the input distribution $P(\xi)$.
3: Find the closest unit $s_1$ and the second closest unit $s_2$ to $\xi$ in $N$.
4: Create a connection between $s_1$ and $s_2$ if it does not exist and set the age of the connection $(s_1, s_2)$ to 0.
5: Increment the age of all edges connected to $s_1$.
6: Update the local error of $s_1$ by $\Delta E_{s_1} = \|\xi - w_{s_1}\|^2$.
7: Move $s_1$ towards $\xi$ by fraction $\epsilon_b$: $\Delta w_{s_1} = \epsilon_b(\xi - w_{s_1})$.
8: Move the neighbors of $s_1$ towards $\xi$ by fraction $\epsilon_n$: $\Delta w_n = \epsilon_n(\xi - w_n)$.
9: Remove all edges with their ages larger than $a_{max}$ and remove nodes without edges.
10: **if** the number of inputs signals is an integer multiple of a parameter $\lambda$ **then**
11: Determine the node $q$ with the maximum accumulated error.
12: Insert a new node $r$ halfway between $q$ and its neighbor $f$ with the largest error: $N = N \cup \{r\}$ with $w_r = 0.5(w_q + w_f)$.
13: Insert the edge connecting $r$ with $q$ and $f$ and remove $(q, f)$.
14: Decrease the error of $q$ and $f$ by $\alpha$: $\Delta E_q = -\alpha E_q$, $\Delta E_f = -\alpha E_f$.
15: Initialize the error of $r$ with the interpolated error: $E_r = 0.5 \cdot (E_q + E_f)$.
16: Decrease all node error variables by $\beta$: $\Delta E_c = -\beta E_c$ $(\forall c \in N)$.
17: **end if**
18: If the stop criterion is not met (minimum error, max network size, max number of neurons), go to step 2.

**Algorithm 5** Growing When Required (Marsland et al., 2002)

1: Start with a set $A$ consisting of two random neurons $n_1$ and $n_2$ in the input space.
2: Initialize an empty set of connections $C = \emptyset$.
3: At each iteration, generate an input sample $\xi$ according to the input distribution $P(\xi)$.
4: For each neuron $i$ calculate the distance from the input $\|\xi - w_i\|$.
5: Select the best matching neuron and the second-best matching neuron such that:
$s = \arg\min_{n \in A} \|\xi - w_n\|$,
$t = \arg\min_{n \in A/\{s\}} \|\xi - w_n\|$.
6: Create a connection $C = C \cup \{(s,t)\}$ if it does not exist and set its age to 0.
7: Calculate the activity of the best matching neuron: $a = exp(-\|\xi - w_s\|)$.
8: If ($a <$ activity threshold $a_T$) and (firing counter $<$ firing threshold $f_T$) then:
Add a new neuron between $s$ and $t$: $A = A \cup \{(r)\}$
Create the weight vector: $w_r = 0.5 \cdot (w_s + \xi)$
Create edges and remove old edge: $C = C \cup \{(r,s),(r,t)\}$ and $C = C/\{(s,t)\}$
9: Else, i.e. no new node is added, adapt the positions of the winning neuron and its neighbors $i$:
$\Delta w_s = \epsilon_b \cdot h_s \cdot (\xi - w_s)$,
$\Delta w_i = \epsilon_n \cdot h_i \cdot (\xi - w_i)$,
where $0 < \epsilon_n < \epsilon_b < 1$ and $h_s$ is the value of the firing counter for neuron $s$.
10: Increment the age of all edges connected to $s$: $age_{(s,i)} = age_{(s,i)} + 1$.
11: Reduce the firing counters:
$h_s(t) = h_0 - \frac{S(t)}{\alpha_b} \cdot (1 - exp(-\alpha_b t/\tau_b))$,
$h_i(t) = h_0 - \frac{S(t)}{\alpha_n} \cdot (1 - exp(-\alpha_n t/\tau_n))$.
12: Remove all edges with ages larger than $a_{max}$ and remove neurons without edges.
13: If the stop criterion is not met, go to step 3.

# Appendix C

# Action Sequences

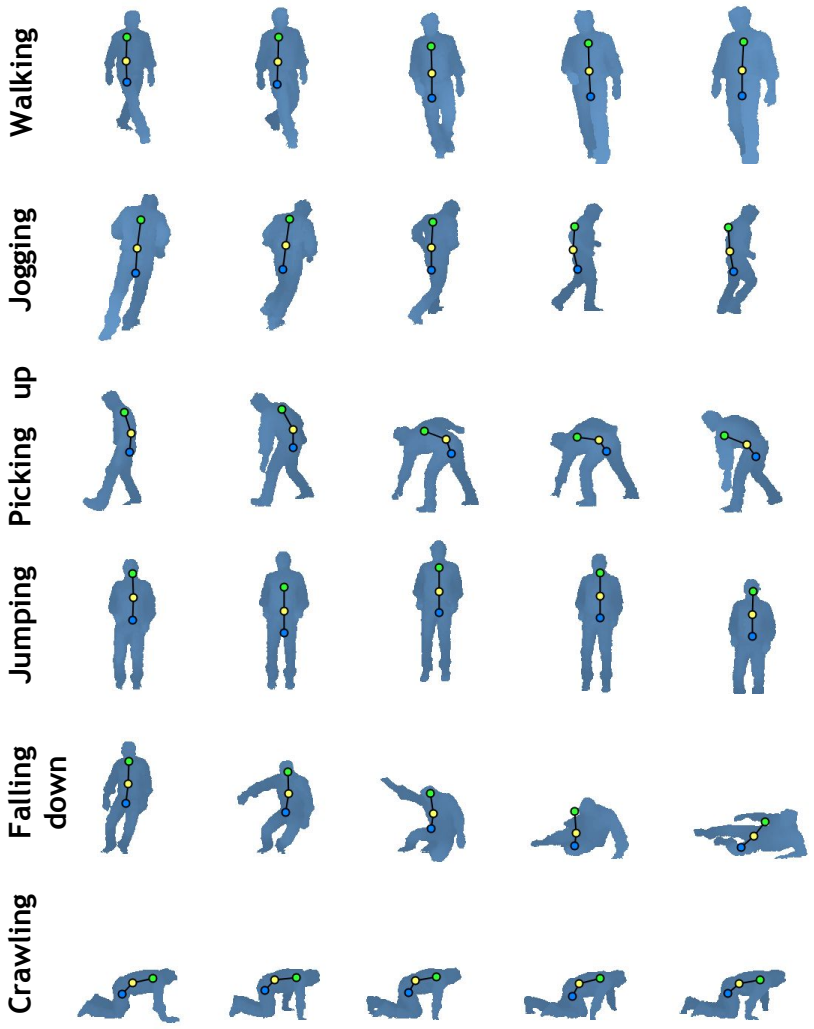



Figure C.1: Example sequences from the KT action dataset (5 frames per second).

# Appendix D

# Additional Results

## CAD-60

Results on S-GWR learning on the CAD-60 dataset. See next page.

## Hand Gesture Recognition

Reported results in (Parisi et al., 2014a) were obtained training the neural network architecture with 10 gesture classes. We captured RGB-D videos with an ASUS Xtion sensor at a constant frame rate of 30 Hz. Each gesture was performed 10 times by three different subjects for a total of 300 training gestures. For testing, each gesture class was performed 30 times by varying sensor distances within the operation range. We run experiments on the 300 testing gestures with single-cue information, i.e. motion and shape, and their combination. The recognition result is based on the statistical mode of the last 3 output labels obtained from the network. For our test set, the average accuracy increase on using combined cues over motion and shape inputs individually is 17% and 13% respectively. Multi-cue combination showed better results also compared to choosing the best result between single-cue approaches, with an average improvement of 10%.

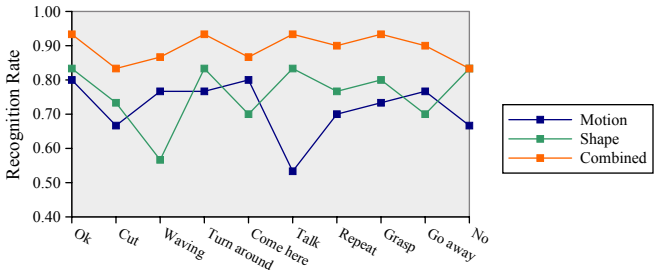



Figure D.1: Evaluation of the system on a set of 10 hand gestures (Parisi et al., 2014a).

Table D.1: Precision, recall, and F-score of S-GWR learning on the five environments of the CAD-60 dataset.

| **Location** | **Activity** | **Precision** | **Recall** | **F-score** |
|---|---|---|---|---|
| Office | talking on the phone | 94.1 | 92.8 | 93.4 |
| | drinking water | 92.9 | 91.5 | 92.2 |
| | working on computer | 94.3 | 93.9 | 94.1 |
| | writing on whiteboard | 95.7 | 94.0 | 94.8 |
| | **Average** | **94.3** | **93.1** | **93.7** |
| Kitchen | drinking water | 93.2 | 91.4 | 92.3 |
| | cooking (chopping) | 86.4 | 86.7 | 86.5 |
| | cooking (stirring) | 88.2 | 86.2 | 87.2 |
| | opening pill container | 90.8 | 84.6 | 87.6 |
| | **Average** | **89.7** | **87.2** | **88.4** |
| Bedroom | talking on the phone | 93.7 | 91.9 | 92.8 |
| | drinking water | 90.9 | 90.3 | 90.6 |
| | opening pill container | 90.8 | 90.1 | 90.4 |
| | **Average** | **91.8** | **91.7** | **91.7** |
| Bathroom | wearing contact lens | 91.2 | 87.0 | 89.1 |
| | brushing teeth | 90.6 | 88.0 | 89.3 |
| | rinsing mouth | 87.9 | 85.8 | 86.8 |
| | **Average** | **89.9** | **86.9** | **88.4** |
| Living room | talking on the phone | 94.8 | 92.1 | 93.4 |
| | drinking water | 91.7 | 90.8 | 91.2 |
| | relaxing on couch | 93.9 | 91.7 | 92.8 |
| | talking on couch | 94.7 | 93.2 | 93.9 |
| | **Average** | **93.8** | **92.0** | **92.9** |

# Appendix E

# Publications Originating from this Thesis

## Journal Articles

- Barros, P., **Parisi, G. I.**, Weber, C., Wermter, S. (2016) Emotional Attention Modulation Applied to Expression Perception with Deep Neural Models. Neurocomputing, in press.

- **Parisi, G. I.**, Tani, J. Weber, C., Wermter, S. (2016) Emergence of Multimodal Action Representations from Neural Network Self-Organization. Cognitive Systems Research, doi:10.1016/j.cogsys.2016.08.002.

- **Parisi, G. I.**, Weber, C., Wermter, S. (2015) Self-Organizing Neural Integration of Pose-Motion Features for Human Action Recognition. Frontiers in Neurorobotics 9(3), doi:10.3389/fnbot.2015.00003.

## Book Chapters

- **Parisi, G. I.**, Wermter, S. (2016) A Neurocognitive Robot Assistant for Robust Event Detection. Trends in Ambient Intelligent Systems: Role of Computational Intelligence, Series "Studies in Computational Intelligence", pp. 1-28, Springer.

## Conference Papers

- Cruz, F., **Parisi, G. I.**, Twiefel, J., Wermter, S. (2016) Multi-Modal Integration of Dynamic Audiovisual Patterns for an Interactive Reinforcement Learning Scenario. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 759–766, Daejeon, Korea.

- Mici, L., **Parisi, G. I.**, Wermter, S. (2016) Recognition of Transitive Actions with Hierarchical Neural Network Learning. In Proceedings of the International Conference on Artificial Neural Networks (ICANN), pages 472–479, Barcelona, Spain.

- **Parisi, G. I.**, Magg, S., Wermter, S. (2016) Human Motion Assessment in Real Time Using Recurrent Self-Organization. In Proceedings of the IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN), pages 71–76, New York, US.

- Cruz, F., **Parisi, G. I.**, Wermter, S. (2016) Learning Contextual Affordances with an Associative Neural Architecture. In Proceedings of the European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN), pages 665-670, Bruges, Belgium.

- **Parisi, G. I.**, v. Stosch, F., Magg, S., Wermter, S. (2015) Learning Human Motion Feedback with Neural Self-Organization. In Proceedings of International Joint Conference on Neural Networks (IJCNN), pp. 2973-2978, Killarney, Ireland.

- Borghetti Soares, M., Barros, P., **Parisi, G. I.**, Wermter, S. (2015) Learning objects from RGB-D sensors using point cloud-based neural networks. In Proceedings of European Symposium on Artifical Neural Networks, Computational Intelligence and Machine Learning (ESANN), pages 439-444, Bruges, Belgium.

- Barros, P., **Parisi, G. I.**, Jirak D. and Wermter, S. (2014) Real-time Gesture Recognition Using a Humanoid Robot with a Deep Neural Architecture. In Proceedings of the IEEE-RAS International Conference on Humanoid Robots (Humanoids), pages 83-88, Madrid, Spain.

- **Parisi, G. I.**, Weber, C., Wermter, S. (2014) Human Action Recognition with Hierarchical Growing Neural Gas Learning. In Proceedings of the International Conference on Artificial Neural Networks (ICANN), pages 89-96, Springer Heidelberg. Hamburg, Germany.

- **Parisi, G. I.**, Jirak, D., Wermter, S. (2014) HandSOM - Neural Clustering of Hand Motion for Gesture Recognition in Real Time. In Proceedings of the IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN), pages 981-986, Springer Heidelberg. Edinburgh, Scotland, UK.

- **Parisi, G. I.**., Barros, P., Wermter, S. (2014) FINGeR: Framework for Interactive Neural-based Gesture Recognition. In Proceedings of the European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN), pages 443-447, Bruges, Belgium.

- **Parisi, G. I.**, Wermter, S. (2013) Hierarchical SOM-Based Detection of Novel Behavior for 3D Human Tracking. In Proceedings of IEEE International Joint Conference on Neural Networks (IJCNN), pages 1380-1387, IEEE. Dallas, US.

## Workshop Papers and Extended Abstracts

- **Parisi, G. I.**, Wermter, S. (2016) Towards Open-Ended Learning of Action Sequences with Hierarchical Predictive Self-Organization. IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN), Workshop on Behaviors Adaptation, Interaction and Learning for Assistive Robotics, New York, US.

- **Parisi, G. I.**, Weber, C., Wermter, S. (2015) Towards Emerging Multimodal Cognitive Representations from Neural Self-Organization. IEEE-RAS International Conference on Humanoid Robots (Humanoids), Workshop on Towards Intelligent Social Robots: Current Advances in Cognitive Robotics, Seoul, South Korea.

- **Parisi, G. I.**, Bauer, J., Strahl, E., Wermter, S. (2015) A Multi-modal Approach for Assistive Humanoid Robots. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Proceedings of the Workshop on Multimodal and Semantics for Robotics Systems (MuSRobS), pp. 10-15, Hamburg, Germany.

- von Stosch, F., **Parisi, G. I.**, Strahl, E., Wermter, S. (2015) Learning Human Motion Feedback with Neural Self-Organization. Video Session @ 24th International Joint Conference on Artificial Intelligence (IJCAI), Buenos Aires, Argentina.

- Cruz, F., **Parisi, G. I.**, Wermter, S. (2015) Contextual Affordances for Action-Effect Prediction in a Robotic-Cleaning Task. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS),Workshop Learning Object Affordances: A Fundamental Step to Allow Prediction, Planning and Tool Use?, Hamburg, Germany.

- **Parisi, G. I.**, Strahl, E., Wermter, S. (2014) Robust Fall Detection with an Assistive Humanoid Robot. 14th IEEE-RAS International Conference on Humanoid Robots (Humanoids), Madrid, Spain.

# Appendix F

# Acknowledgements

I would like to acknowledge the support and effort of those who have extensively contributed to conducting stimulating research and completing my PhD studies at the University of Hamburg during the last three years.

First of all, I would like to deeply thank my supervisor Stefan Wermter for constant guidance, indispensable support, and insightful advice.

I wish to thank my current and former colleagues of the Knowledge Technology research group. In particular, I thank Jorge Dávila-Chacón, Johannes Bauer, Doreen Jirak, Pablo Barros, Sven Magg, Nils Meins, Francisco Cruz, Stefan Heinrich, Alex Yang, Nicolás Navarro-Guerrero, Junpei Zhong, Johannes Twiefel, and Sasha Griffiths for vivid discussions both at a professional and personal level. Furthermore, I thank Cornelius Weber, Erik Strahl and Katja Kösters for invaluable analytical, technical, and administrative support respectively.

In these three years, I had the chance to travel and meet outstanding researchers. I would like to thank Jun Tani, Angelo Cangelosi, Martin Giese, Francesca Odone, George M. Georgiou, Thomas Martinetz, and Stephen Marsland for inspiring discussions and feedback.

I gratefully acknowledge the support of the University of Hamburg, the DAAD German Academic Exchange Service for the Cognitive Assistive Systems project (Kz:A/13/94748), the Transregio TRR169 on Crossmodal Learning, and the Hamburg Landesforschungsförderung.

On a more personal note, I am greatly thankful to my dearest friends Frank Lococo, Jordan Besta, Leontina Di Cecco, Giulia D'Angelo, Mauro Meloni, Rosa Crespo, and Jorman Gerge for countless moments of laughter and insight. Additionally, I thank Thomas Haake, Fredrik Thordendal, Jens Kidman, Guillermo Francella, and Hugo Sofovich for enduring sparks of creativity and multilayered inspiration.

Finally, I would like to wholeheartedly thank my parents E.J. and Laura, and my brother Matías for unconditional love and never-ending support of all kinds.

# Bibliography


Adolphs, R. (2003). Cognitive neuroscience of human social behaviour. *Nature Reviews Neuroscience*, 4(3):165–178.

Aerts, M., Esselink, R., Post, B., van de Warrenburg, B., and Bloem, B. (2012). Improving the diagnostic accuracy in parkinsonism: a three-pronged approach. *Practical Neurology*, 12(1):77–87.

Allison, T., Puce, A., and McCarthy, G. (2000). Social perception from visual cues: role of the STS region. *Trends in cognitive sciences*, 4(7):267–278.

Andreakis, A., von Hoyningen-Huene, N., and Beetz, M. (2009). Incremental unsupervised time series analysis using merge growing neural gas. In *Workshop on Self-Organizing Maps (WSOM)*, volume 5629 of *Lecture Notes in Computer Science*, pages 10–18. Springer.

Baccouche, M., Mamalet, F., Wolf, C., Garcia, C., and Baskurt, A. (2011). Sequential deep learning for human action recognition. In *Human Behavior Understanding (HBU): Second International Workshop*, pages 29–39. Springer Berlin Heidelberg.

Barraclough, N. E., Xiao, D., Baker, C. I., Oram, M. W., and Perrett, D. I. (2005). Integration of visual and auditory information by superior temporal sulcus neurons responsive to the sight of actions. *Journal of Cognitive Neuroscience*, 17(3):377–391.

Bauer, J., Magg, S., and Wermter, S. (2015). Attention modeled as information in learning multisensory integration. *Neural Networks*, 65:44–52.

Beauchamp, M. S. (2005). See me, hear me, touch me: multisensory integration in lateral occipital-temporal cortex. *Current Opinion in Neurobiology*, 15(2):145 – 153.

Beauchamp, M. S., Lee, K. E., Argall, B. D., and Martin, A. (2004). Integration of auditory and visual information about objects in superior temporal sulcus. *Neuron*, 41(5):809–823.

Beauchamp, M. S., Lee, K. E., Haxby, J. V., , and Martin, A. (2003). fmri responses to video and point-light displays of moving humans and manipulable objects. *Journal of Cognitive Neuroscience*, 15(7):991–1001.

Beauchamp, M. S., Yasar, N. E., Frye, R. E., and Ro, T. (2008). Touch, sound and vision in human superior temporal sulcus. *NeuroImage*, 41(3):1011 – 1020.

Belin, P., Zatorre, R. J., and Ahad, P. (2002). Human temporal-lobe response to vocal sounds. *Cognitive Brain Research*, 13(1):17 – 26.

Belin, P., Zatorre, R. J., Lafaille, P., Ahad, P., and Pike, B. (2000). Voice-selective areas in human auditory cortex. *Nature*, 403(6767):309–312.

Bertenthal, B. I. and Pinto, J. (1993). Complementary processes in the perception and production of human movement. *Dynamic Approaches to Development 2 (MIT Press, Cambridge)*, pages 209–239.

Beyer, O. and Cimiano, P. (2011). Online labelling strategies for growing neural gas. In Yin, H., Wang, W., and Rayward-Smith, V., editors, *Proceedings of the International Conference on Intelligent Data Engineering and Automated Learning (IDEAL)*, volume 6936 of *LNCS*, pages 76–83. Springer Berlin Heidelberg.

Blake, R. and Shiffrar, M. (2007). Perception of human motion. *Annual Review of Psychology*, 58(1):47–73.

Blakemore, C. and Cooper, G. F. (1970). Development of the brain depends on the visual environment. *Nature*, 228(5270):477–478.

Blakemore, C. and Van Sluyters, R. C. (1975). Innate and environmental factors in the development of the kitten's visual cortex. *Journal of Physiology*, 248(3):663–716.

Blank, M., Gorelick, L., Shechtman, E., Irani, M., and Basri, R. (2007). Actions as space-time shapes. *Transactions on Pattern Analysis and Machine Intelligence*, 29(12):2247–2253.

Bloom, L. (1993). *The transition from infancy to language: Acquiring the power of expression*. New York: Cambridge University Press.

Bushara, K., Grafman, J., and Hallett, M. (2001). Neural correlates of auditory-visual stimulus onset asynchrony detection. *Journal of Neuroscience*, 21(1):300 – 304.

Calvert, G. A. (2001). Crossmodal processing in the human brain: insights from functional neuroimaging studies. *Cerebral Cortex*, 11(8):1110–1123.

Calvert, G. A., Campbell, R., and Brammer, M. J. (2000). Evidence from functional magnetic resonance imaging of crossmodal binding in the human heteromodal cortex. *Current Biology*, 10(11):649 – 657.

Chandola, V., Banerjee, A., and Kumar, V. (2009). Anomaly detection: A survey. *ACM Computer Surveillance*, 15.

Chang, Y.-J., Chen, S.-F., and Huang, J.-D. (2011). A Kinect-based system for physical rehabilitation: A pilot study for young adults with motor disabilities. *Research in Developmental Disabilities*, 32(6):2566–2570.

Chappell, G. J. and Taylor, J. G. (1993). The temporal Kohonen map. *Neural Networks*, 6(3):441–445.

Cruz, F., Magg, S., Weber, C., and Wermter, S. (2016a). Training agents with interactive reinforcement learning and contextual affordances. *IEEE Transactions on Autonomous Mental Development*.

Cruz, F., Parisi, G., Twiefel, J., and Wermter, S. (2016b). Multi-modal integration of dynamic audiovisual patterns for an interactive reinforcement learning scenario. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 759–766.

Cruz, F., Twiefel, J., Magg, S., Weber, C., and Wermter, S. (2015). Interactive reinforcement learning through speech guidance in a domestic scenario. In *Proceedings of International Joint Conference on Neural Networks (IJCNN)*, pages 1341–1348.

Cucchiara, R., Prati, A., and Vezzani, R. (2007). A multi-camera vision system for fall detection and alarm generation. *Expert Systems*, 24:334–345.

Dautenhahn, K. (1999). Robots as social actors: Aurora and the case of autism. In *Third Cognitive Technology Conference*.

de Vries, B. and Príncipe, J. C. (1992). The gamma model–a new neural model for temporal processing. *Neural Networks*, 5(4):565–576.

Dick, A., Goldin-Meadow, S., Solodkin, A., and Small., S. (2012). Gesture in the developing brain. *Developmental Science*, 15(2):165–180.

Diraco, G., Leone, A., and Siciliano, P. (2010). An active vision system for fall detection and posture recognition in elderly healthcare. *Conference & Exhibition: Design, Automation & Test in Europe. Dresden: European Design and Automation Association*, pages 1536–1541.

Dirichlet, G. L. (1850). Über die reduktion der positiven quadratischen formen mit drei unbestimmten ganzen zahlen. *Journal für die Reine und Angewandte Mathematik*, 40:209–227.

Driver, J. (2001). A selective review of selective attention research from the past century. *British Journal of Psychology*, 92(1):53–78.

Eriksson, P. S., Perfilieva, E., Bjork-Eriksson, T., Alborn, A.-M., Nordborg, C., Peterson, D. A., and Gage, F. H. (1998). Neurogenesis in the adult human hippocampus. *Nature Medicine*, 4(11):1313–1317.

Ernst, M. O. and Bülthoff, H. (2004). Merging the senses into a robust percept. *Trends in Cognitive Sciences*, 8(4):162–169.

Estévez, P. A. and Hernández, R. (2011). Gamma-filter self-organizing neural networks for time series analysis. In *Workshop on Self-Organizing Maps (WSOM)*, pages 151–159.

Estévez, P. A. and Vergara, J. R. (2012). Nonlinear time series analysis by using gamma growing neural gas. In *Workshop on Self-Organizing Maps (WSOM)*, pages 205–214.

Faria, D. R., Premebida, C., and Nunes, U. (2014). A probabilistic approach for human everyday activities recognition using body motion from RGB-D images. In *Proceedings of the IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, pages 842–849.

Fathi, A. and Mori, G. (2008). Action recognition by learning mid-level motion features. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*.

Felleman, D. and Van Essen, D. (1991). Distributed hierarchical processing in the primate cerebral cortex. *Cerebral Cortex*, 1(1):1–47.

Fiebelkorn, I. C., Foxe, J. J., and Molholm, S. (2009). Dual mechanisms for the cross-sensory spread of attention: How much do learned associations matter? *Cerebral Cortex*, 20:109–120.

Fleischer, F., Caggiano, V., Thier, P., and Giese, M. (2013). Physiologically inspired model for the visual recognition of transitive hand actions. *The Journal of Neurosciences*, 33(15):6563–6580.

Földiák, P. (1991). Learning invariance from transformation sequences. *Neural Computation*, 3(2):194–200.

Fonlupt, P. (2003). Perception and judgement of physical causality involve different brain structures. *Cognitive Brain Research*, 17(2):248 – 254.

Foxe, J. J., Morocz, I. A., Murray, M. M., Higgins, B. A., Javitt, D. C., and Schroeder, C. E. (2000). Multisensory auditorysomatosensory interactions in early cortical processing revealed by high-density electrical mapping. *Cognitive Brain Research*, 10(12):77 – 83.

Fritzke, B. (1995). A growing neural gas network learns topologies. In *Advances in Neural Information Processing Systems 7*, pages 625–632. MIT Press.

Fritzke, B. (1997). A self-organizing network that can follow non-stationary distributions. In *Proceedings of the International Conference on Artificial Neural Networks (ICANN)*, pages 613–618. Springer.

Gaglio, S., Lo Re, M., and Morana, M. (2014). Human activity recognition process using 3-D posture data. *IEEE Trans. on Human-Machine Systems*, 99:1–12.

Gallese, V., Fadiga, L., Fogassi, L., and Rizzolatti, G. (1996). Action recognition in premotor cortex. *Brain*, 2:593–609.

Garcia, J. O. and Grossman, E. D. (2008). Necessary but not sufficient: Motion perception is required for perceiving biological motion. *Vision Research*, 48(9):1144–1149.

Gentner, D. (1982). Why nouns are learned before verbs: Linguistic relativity versus natural partitioning. *Language development: Language, thought,and culture*, 2:301–334.

Georgiou, G. (2006). Exact interpolation and learning in quadratic neural networks. In *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*, pages 230–234.

Georgiou, G. and Voigt, K. (2013). Self-organizing maps with a single neuron. In *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*, pages 1–6.

Giese, M., Thornton, I., and Edelman, S. (2008). Metrics of the perception of body movement. *Journal of Vision*, 8(9):1–18.

Giese, M. A. (2015). Biological and body motion perception. *Johan Wagemans (ed.):The Oxford Handbook of Perceptual Organization, Oxford University Press, Oxford*.

Giese, M. A. and Poggio, T. (2003). Neural mechanisms for the recognition of biological movements. *Nature Reviews Neuroscience*, 4(3):179–192.

Giese, M. A. and Rizzolatti, G. (2015). Neural and computational mechanisms of action processing: Interaction between visual and motor representations. *Neuron*, 88(1):167–180.

Goodhill, G. and Sejnowski, T. (1997). A unifying objective function for topographic mappings. *Neural Computation*, 9:1291–1303.

Gorelick, L., Blank, M., Shechtman, E., Irani, M., and Basri, R. (2005). Actions as space-time shapes. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 1395–1402.

Gould, E. (2007). How widespread is adult neurogenesis in mammals? *Nature Reviews Neuroscience*, 8(6):481–488.

Griffith, S., Subramanian, K., Scholz, J., Isbell, C., and Thomaz, A. (2013). Policy shaping: Integrating human feedback with reinforcement learning. In *Advances in Neural Information Processing Systems*, pages 2625–2633.

Guo, Y., Liu, Y., Oerlemans, A., Lao, S., Wu, S., and Lew, M. S. (2016). Deep learning for visual understanding: A review. *Neurocomputing*, 187:27 – 48.

Gupta, R., Chia, A. Y.-S., and Rajan, D. (2013). Human activities recognition using depth images. In *ACM International Conference on Multimedia*, pages 283–292.

Hagenbuchner, M., Sperduti, A., , and Tsoi, A. C. (2003). A self-organizing map for adaptive processing of structured data. *IEEE Transactions on Neural Networks*, 14(3):491–505.

Han, J., Shao, L., Xu, D., and Shotton, J. (2013). Enhanced computer vision with Microsoft Kinect sensor. *IEEE Transactions on cybernetics*, 43(5):1318–1334.

Hasson, U., Yang, E., Vallines, I., Heeger, D. J., and Rubin, N. (2008). A hierarchy of temporal receptive windows in human cortex. *The Journal of Neuroscience*, 28(10):2539–2550.

Hazelhoff, L., Han, J., and de With, P. (2008). Video-based fall detection in the home using principal component analysis. *Proceedings of the 10th International Conference on Advanced Concepts for Intelligent Vision Systems*, pages 298–309.

Hebb, D. O. (1949). The organization of behavior: a neuropsychological theory. *Wiley, New York*.

Hiris, E. (2007). Detection of biological and nonbiological motion. *Journal of Vision*, 7(12):1–16.

Hirsch, H. (1985). The role of visual experience in the development of cat striate cortex. *Cellular and Molecular Neurobiology*, 5:103–121.

Hirsch, H. and Spinelli, D. (1970). Visual experience modifies distribution of horizontally and vertically oriented receptive fields in cats. *Science*, 168(3933):869–871.

Hirsch-Pasek, K., Golinkoff, R., and Hollich, G. (2000). An emergentist coalition model for word learning: mapping words to objects is a product of the interaction of multiple cues. In *Becoming a Word Learner: a debate on lexical acquisition*, pages 136–165. Oxford University Press.

Hoglund, A. J., Hatonen, K., and Sorvari, A. S. (2000). A computer host-based user anomaly detection system using self-organizing maps. In *Proceedings of International Joint Conference on Neural Networks (IJCNN)*, pages 411–416.

Hosoya, H. and Hyvärinen, A. (2016). Learning visual spatial pooling by strong PCA dimension reduction. *Neural Computation*, 28(7):1249–1264.

Hu, W., Tan, T., Wang, L., and Maybank, S. (2004a). A survey on visual surveillance of object motion and behaviors. *IEEE Transactions on Systems, Man and Cybernetics*, 34(3):334–352.

Hu, W., Xie, D., and Tan, T. (2004b). A hierarchical self-organizing approach for learning the patterns of motion trajectories. *IEEE Transactions on Neural Networks*, 15(1):135–144.

Huang, Y. and Rao, R. P. N. (2011). Predictive coding. *Wiley Interdisciplinary Reviews: Cognitive Science*, 2(5):580–593.

Hubel, D. H. and Wiesel, T. H. (1962). Receptive fields, binocular and functional architecture in the cat's visual cortex. *Journal of Physiology*, 160:106–154.

Hubel, D. H. and Wiesel, T. H. (1967). Cortical and callosal connections concerned with the vertical meridian of visual fields in the cat. *Journal of Neurophysiology*, 30:1561–1573.

Hubel, D. H. and Wiesel, T. H. (1970). The period of susceptibility to the psychological effects of unilateral eye closure in kittens. *Journal of Physiology*, 206:419–436.

Hubel, D. H., Wiesel, T. N., and LeVay, S. (1977). Plasticity of ocular dominance columns in monkey striate cortex. *Philosophical Transactions of the Royal Society of London, Series B: Biological Sciences*, 278:377–409.

Ivanov, I., Liu, X., Clerkin, S., Schulz, K., Friston, K., Newcorn, J. H., and Fan, J. (2012). Effects of motivation on reward and attentional networks: an fMRI study. *Brain and Behavior*, 2(6):741–753.

Jain, A., Tompson, J., LeCun, Y., and Bregler, C. (2015). Modeep: A deep learning framework using motion features for human pose estimation. In *Proceedings of the Asian Conference on Computer Vision (ACCV)*, pages 302–315, Cham. Springer International Publishing.

Jastorff, J., Kourtzi, Z., , and Giese, M. A. (2006). Learning to discriminate complex movements: biological versus artificial trajectories. *Journal of Vision*, 6(8):791–804.

Jégou, H., Douze, M., Schmid, C., and Pérez, P. (2010). Aggregating local descriptors into a compact image representation. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3304–3311.

Jhuang, H., Serre, T., Wolf, L., and Poggio, T. (2007). A biologically inspired system for action recognition. In *Proceedings of the International Conference on Computer Vision (ICCV)*.

Jiang, Z., Lin, Z., and Davis, L. (2012). Recognizing human actions by learning and matching shape-motion prototype trees. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 34(3):533–547.

Johansson, G. (1973). Visual perception of biological motion and a model for its analysis. *Percept Psychophys*, 14:195–204.

Johansson, G. (1976). Spatio-temporal differentiation and integration in visual motion perception. an experimental and theoretical analysis of calculus-like functions in visual data processing. *Psychiatric Research*, pages 379–393.

Jolliffe, I. (2002). Principal component analysis. *Springer, New York*, pages 2805–2845.

Jung, M., Hwang, J., and Tani, J. (2015). Self-organization of spatio-temporal hierarchy via learning of dynamic visual image patterns on action sequences. *PLoS ONE*, 10(7):e0131214.

Kachouie, R., Sedighadeli, S., Khosla, R., and Chu, M. (2014). Socially assistive robots in elderly care: A mixed-method systematic literature review. *Int. J. Hum. Comput. Interaction*, 30(5):369–393.

Kaluza, B., Cvetkovic, B., Dovgan, E., Gjoreski, H., M., G., and Lustrek, M. (2013). A multi-agent care system to support independent living. *International Journal of Artificial Intelligence Tools*, 23(1):1–20.

Kidd, C. D. and Breazeal, C. (2007). A robotic weight loss coach. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 1985–1986.

Knapp, C. and Carter, G. C. (1976). The generalized correlation method for estimation of time delay. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 24(4):320–327.

Knox, W., Stone, P., and Breazeal, C. (2013). Training a robot via human feedback: A case study. In *Proceedings of the International Conference on Social Robotics (ICSR)*, pages 460–470.

Kohonen, T. (1990). The self-organizing map. *Proceedings of the IEEE*, 78:1464–1480.

Kohonen, T. (1991). Self-organizing maps: optimization approaches. In *Artificial neural networks, II*, pages 981–990.

Kohonen, T. (2013). Essentials of the self-organizing map. *Neural Networks*, 37:52–65.

Koppula, H. S., Gupta, R., and Saxena, A. (2013). Learning human activities and object affordances from RGB-D videos. *International Journal of Robotics Research*, 32(8):951–970.

Krauzlis, R. J., Lovejoy, L. P., , and Zénon, A. (2013). Superior colliculus and visual spatial attention. *Annual review of neuroscience*, 36(1):165–182.

Lacheze, L., Yan, G., Benosman, R., Gas, B., and Couverture, C. (2009). Audio/video fusion for objects recognition. *IEEE/RSJ Intl. Conference on Intelligent Robots and Systems (IROS-09), St. Louis, MO*, pages 652–657.

Lange, J. and Lappe, M. (2006). A model of biological motion perception from configural form cues. *Journal of Neuroscience*, 26(11):2894–2906.

Lapeyre, M., Rouanet, P., Grizou, J., N'Guyen, S., Falher, A. L., Depraetre, F., and Oudeyer, P.-Y. (2014). Poppy: Open source 3D printed robot for experiments in developmental robotics. In *Proceedings of the Joint IEEE International Conferences on Development and Learning and Epigenetic Robotics (ICDL-EpiRob)*, pages 173–174.

Layher, G., Giese, M. A., and Neumann, H. (2013). Learning representations of animated motion sequences – a neural model. In *35th annual meeting of the Cognitive Science Society*.

Layher, G., Giese, M. A., and Neumann, H. (2014). Learning representations of animated motion sequencesA neural model. *Topics in Cognitive Science*, 6(1):170–182.

Lee, J. (2012). Encyclopedia of the sciences of learning. pages 887–893, Boston, MA. Springer US.

Lee, T. and Mihailidis, A. (2005). An intelligent emergency response system: preliminary development and testing of automated fall detection. *J Telemed Telecare*, 11:194–198.

Lerner, Y., Honey, C. J., Silbert, L. J., and Hasson, U. (2011). Topographic mapping of a hierarchy of temporal receptive windows using a narrated story. *The Journal of neuroscience*, 31(8):2906–2915.

Levenshtein, V. (1966). Binary codes capable of correcting deletions, insertions and reversals. *Soviet Physics Doklady*, 19:707.

Liu, C., Lee, C., and Lin, P. (2010). A fall detection system using k-nearest neighbor classifier. *Expert Systems with Applications*, 37:7174–7181.

Lowe, D. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110.

Lücke, J. (2009). Receptive field self-organization in a model of the fine structure in v1 cortical columns. *Neural Computation*, 21(10):2805–2845.

Macaluso, E., George, N., Dolan, R., Spence, C., and Driver, J. (2004). Spatial and temporal factors during processing of audiovisual speech: a {PET} study. *NeuroImage*, 21(2):725 – 732.

Marsland, S., Nehmzow, U., and Shapiro, J. (2005). On-line novelty detection for autonomous mobile robots. *Robotics and Autonomous Systems*, 51(2-3):191–206.

Marsland, S., Shapiro, J., and Nehmzow, U. (2002). A self-organising network that grows when required. *Neural Networks*, 15(8-9):1041–1058.

Martinetz, T., Berkovich, S., and Schulten, K. (1993). Neural-gas network for vector quantization and its application to time-series prediction. *IEEE Transactions on Neural Networks*, 4(4):558–569.

Martinetz, T. M. (1993). Competitive hebbian learning rule forms perfectly topology preserving maps. In *Proceedings of the International Conference on Artificial Neural Networks (ICANN)*, pages 427–434, Amsterdam. Springer.

Martinson, E. (2014). Detecting occluded people for robotic guidance. In *Proceedings of the IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, pages 744–749.

Meltzoff, A., , and Moore, M. (1977). Imitation of facial and manual gestures by human neonates. *Science*, 198:74–78.

Miaou, S., Sung, P., and Huang, C. (2006). A customized human fall detection system using omni-camera images and personal information. *Proceedings of the 1st Distributed Diagnosis and Home Healthcare Conference. Arlington: Institute of Electrical and Electronics Engineers*, pages 39–42.

Mici, L., Parisi, G., and Wermter, S. (2016). Recognition of transitive actions with hierarchical neural network learning. In *Proceedings of the International Conference on Artificial Neural Networks (ICANN)*, pages 472–479.

Miikkulainen, R., Bednar, J. A., Choe, Y., and Sirosh, J. (2005). *Computational Maps in the Visual Cortex*. Springer.

Miller, L. E. and Saygin, A. P. (2013). Individual differences in the perception of biological motion: Links to social cognition and motor imagery. *Cognition*, 128(2):140 – 148.

Mineiro, P. and Zipser, D. (1998). Analysis of direction selectivity arising from recurrent cortical interactions. *Neural Computation*, 10(2):353–371.

Ming, G.-l. and Song, H. (2011). Adult neurogenesis in the mammalian brain: Significant answers and significant questions. *Neuron*, 70(4):687–702.

Morse, A. F., Benitez, V. L., Belpaeme, T., Cangelosi, A., and Smith, L. B. (2015). Posture affects how robots and infants map words to objects. *PLoS ONE*, 10(3):e0116012.

Mountcastle, V. B. (1957). Modality and topographic properties of single neurons of cat's somatic sensory cortex. *Journal of Neurophysiology*, 20:408–434.

Mozer, M. (1995). A focused backpropagation algorithm for temporal pattern recognition. In *Hillsdale, NJ: Lawrence Erlbaum Associates*, page 137169.

Müller, S., Weber, C., and Wermter., S. (2014). RatSLAM on humanoids - a bio-inspired SLAM model adapted to a humanoid robot. In *Proceedings of the International Conference on Artificial Neural Networks (ICANN)*, pages 789–796.

Mundher, Z. A. and Zhong, J. (2014). A real-time fall detection system in elderly care using mobile robot and Kinect sensor. *Int. Journal of Materials, Mechanics and Manufacturing*, 2(2):133–138.

Nag, A. K., A., M., and Mitra, S. (2005). Multiple outlier detection in multivariate data using self-organizing maps title. *Computational Statistics*, 2(2):245–264.

Nalin, M., Baroni, I., Sanna, A., and Pozzi, C. (2012). Robotic companion for diabetic children: emotional and educational support to diabetic children, through an interactive robot. In *ACM SIGCHI*, pages 260–263.

Nelissen, K., Luppino, G., Vanduffel, W., Rizzolatti, G., and Orban, G. (2005). Observing others: Multiple action representation in frontal lobe. *Science*, 310:332–336.

Nelson, C. A. (2000). Neural plasticity and human development: the role of early experience in sculpting memory systems. *Developmental Science*, 3(2):115–136.

Neri, P., Morrone, M., and Burr, D. (1998). Seeing biological motion. *Nature*, 395:894–896.

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

Niebles, J. C., Wang, H., and Fei-Fei, L. (2008). Unsupervised learning of human action categories using spatial-temporal words. *International Journal of Computer Vision*, 79(3):299–318.

Noda, K., Arie, H., Suga, Y., and Ogata, T. (2014). Multimodal integration learning of robot behavior using deep neural networks. *Robotics and Autonomous Systems*, 62(6):721 – 736.

Norman, J., Payton, S., Long, J., and Hawkes, L. (2004). Aging and perception of biological motion. *Psychology and Aging*, 19:19–25.

O'Donovan, M. J. (1999). The origin of spontaneous activity in developing networks of the vertebrate nervous system. *Current Opinion in Neurobiology*, 9:94–104.

Oram, M. W. and Perrett, D. I. (1996). Integration of form and motion in the anterior superior temporal polysensory area (STPa) of the macaque monkey. *Journal of Neurophysiology*, 76(1):109–129.

Orban, G., Lagae, L., Verri, A., Raiguel, S., D., X., Maes, H., and Torre, V. (1982). First-order analysis of optical flow in monkey brain. *Proceedings of the National Academy of Sciences*, 89(7):2595–2599.

Paiement, A., Tao, L., Camplani, M., Hannuna, S., Damen, D., and Mirmehdi, M. (2014). Online quality assessment of human motion from skeleton data. In *Proceedings of the British Machine Vision Conference (BMVC)*. BMVA Press.

Papadopoulos, G. T., Axenopoulos, A., and Daras, P. (2014). Real-time skeleton-tracking-based human action recognition using Kinect data. In *MultiMedia Modeling*, volume 8325 of *Lecture Notes in Computer Science*, pages 473–483. Springer.

Parisi, G. I., Barros, P., and Wermter, S. (2014a). FINGeR: Framework for interactive neural-based gesture recognition. In *Proceedings of the European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN), Bruges, Belgium*, pages 443–447.

Parisi, G. I., Jirak, D., and Wermter, S. (2014b). HandSOM - Neural clustering of hand motion for gesture recognition in real time. In *Proceedings of the IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN), Edinburgh, Scotland, UK*, pages 981–986.

Parisi, G. I., Magg, S., and Wermter, S. (2016a). Human motion assessment in real time using recurrent self-organization. In *Proceedings of the IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, pages 71–76.

Parisi, G. I., Tani, J., Weber, C., and Wermter, S. (2016b). Emergence of multimodal action representations from neural network self-organization. *Cognitive Systems Research*.

Parisi, G. I., von Stosch, F., Magg, S., and Wermter, S. (2015a). Learning human motion feedback with neural self-organization. In *Proceedings of International Joint Conference on Neural Networks (IJCNN)*, pages 2973–2978.

Parisi, G. I., Weber, C., and Wermter, S. (2014c). Human action recognition with hierarchical growing neural gas learning. In *Proceedings of the International Conference on Artificial Neural Networks (ICANN)*, pages 89–96.

Parisi, G. I., Weber, C., and Wermter, S. (2015b). Self-organizing neural integration of pose-motion features for human action recognition. *Frontiers in Neurorobotics*, 9(3).

Parisi, G. I., Weber, C., and Wermter, S. (2016c). A neurocognitive robot assistant for robust event detection. *Trends in Ambient Intelligent Systems: Role of Computational Intelligence, Series "Studies in Computational Intelligence", Springer*, pages 1–28.

Parisi, G. I. and Wermter, S. (2013). Hierarchical SOM-based detection of novel behavior for 3D human tracking. In *Proceedings of International Joint Conference on Neural Networks (IJCNN)*, pages 1380–1387.

Pavlova, M., Krageloh-Mann, I., Sokolov, A., and Birbaumer, N. (2001). Recognition of point-light biological motion displays by young children. *Perception*, 30:925–933.

Pavlova, M. and Sokolov, S. (2008). Orientation specificity in biological motion perception. *Percept Psychophys*, 62:889–899.

Perrett, D., Rolls, E., and Caan, W. (1982). Visual neurons responsive to faces in the monkey temporal cortex. *Experimental Brain Research*, 47:329–342.

Pirsiavash, H., Vondrick, C., and Torralba, A. (2014). Assessing the quality of actions. In *Proceedings of the European Conference Computer Vision (ECCV)*, pages 556–571.

Poggio, T. and Smale, S. (2003). The mathematics of learning: Dealing with data. *Notices of the American Mathematical Society*, 50:2003.

Poom, L. and Olsson, H. (2002). Are mechanisms for perception of biological motion different from mechanisms for perception of nonbiological motion? *Perceptual and Motor Skills*, 95:1301–1310.

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

Pulvermüller, F. (2005). Brain mechanisms linking language and action. *Nature Reviews Neuroscience*, 6(7):576–582.

Raij, T., Uutela, K., and Hari, R. (2000). Audiovisual integration of letters in the human brain. *Neuron*, 28(2):617–625.

Rao, R. and Ballard, D. (1999). Predictive coding in the visual cortex: a functional interpretation of some extra-classical receptive-field effects. *Nature Neuroscience*, 2:79–87.

Rodriguez, M. D., Ahmed, J., and Shah, M. (2008). Action MACH: a spatio-temporal maximum average correlation height filter for action recognition. In *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–18.

Rougier, C., Meunier, J., St-Arnaud, A., and Rousseau, J. (2011). Robust video surveillance for fall detection based on human shape deformation. *IEEE Transactions on Circuits and Systems for Video Technology*, 21:611–622.

Salin, P. and Bullier, J. (1995). Corticocortical connections in the visual system: structure and function. *Physiological Reviews*, 75(1):107–154.

Sanchez-Riera, J., Alameda-Pineda, X., Wienke, J., and Deleforge, A. (2009). Online multimodal speaker detection for humanoid robots. *IEEE-RAS International Conference on Humanoid Robots (Humanoids-12), Osaka, Japan*, pages 126–133.

Saxe, R., Carey, S., and Kanwisher, N. (2004). Understanding other minds: Linking developmental psychology and functional neuroimaging. *Annual Review of Psychology*, 55:87–124.

Schalkwyk, J., Beeferman, D., Beaufays, F., Byrne, B., Chelba, C., Cohen, M., Kamvar, M., and Strope, B. (2010). Your word is my command: Google search by voice: A case study. In *Advances in Speech Recognition, Springer US*, pages 61–90.

Scheffer, A., Schuurmans, M., van Dijk, N., van der Hooft, T., and de Rooij, S. (2008). Fear of falling: measurement strategy, prevalence, risk factors and consequences among older persons. *Age Ageing*, 37(1):19–24.

Scherer, D., Müller, A., and Behnke, S. (2010). Evaluation of pooling operations in convolutional architectures for object recognition. In *Proceedings of the International Conference on Artificial Neural Networks (ICANN)*, pages 92–101, Berlin, Heidelberg. Springer-Verlag.

Schindler, K. and Van Gool, L. J. (2008). Action snippets: How many frames does human action recognition require? In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE Computer Society.

Schnupp, J., Nelken, I., and King, A. (2010). Auditory neuroscience: Making sense of sound. *1st ed. Cambridge, MA: MIT Press*.

Schuldt, C., Laptev, I., and Caputo, B. (2004). Recognizing human actions: A local SVM approach. In *Proceedings of the International Conference on the Pattern Recognition (ICPR)*, pages 32–36, Washington, DC, USA. IEEE Computer Society.

Sejnowski, T. J. (1977). Statistical constraints on synaptic plasticity. *Journal of Theoretical Biology*, 69:385–389.

Sengpiel, F., Stawinski, P., and T., B. (1999). Influence of experience on orientation maps in cat visual cortex. *Nature Neurosciensce*, 2(8):727–732.

Senkowski, D., Saint-Amour, D., Höfle, M., and Foxe, J. J. (2011). Multisensory interactions in early evoked brain activity follow the principle of inverse effectiveness. *NeuroImage*, 56(4):2200–2208.

Shan, J. and Akella, S. (2014). 3D human action segmentation and recognition using pose kinetic energy. In *Workshop on Advanced Robotics and its Social Impacts (IEEE)*, pages 69–75.

Shatz, C. J. (1990). Impulse activity and the patterning of connections during CNS development. *Neuron*, 5:745–756.

Shatz, C. J. (1992). The developing brain. *Scientific American*, 267(3):60–67.

Shatz, C. J. (1996). Emergence of order in visual system development. *Proceedings of the National Academy of Sciences*, 93:602–608.

Shiffrar, M. and Freyd, J. J. (1990). Apparent motion of the human body. *Psychological Science*, 1:257–264.

Smith, L. and Yu, C. (2008). Infants rapidly learn word-referent mappings via cross-situational statistics. *Cognition*, 106(3):1558–1568.

Somers, D., Nelson, S., and Sur, M. (1995). An emergent model of orientation selectivity in cat visual cortical simple cells. *Journal of Neuroscience*, 15(8):5448–5465.

Stanley, J. C. (1976). Computer simulation of a model of habituation. *Nature*, 261:146–148.

Stein, B. E. and Meredith, M. A. (1993). *The merging of the senses*. The MIT Press, Cambridge, MA, US.

Stein, B. E., Stanford, T. R., and Rowland, B. A. (2009). The neural basis of multisensory integration in the midbrain: Its organization and maturation. *Hearing Research*, 258(12):4–15.

Stevenson, R. A. and James, T. W. (2009). Audiovisual integration in human superior temporal sulcus: Inverse effectiveness and the neural processing of speech and object recognition. *NeuroImage*, 44(3):1210 – 1223.

Stiles, J. (2000). Neural plasticity and cognitive development. *Developmental Neuropsychology*, 18(2):237–272.

Strickert, M. and Hammer, B. (2005). Merge SOM for temporal data. *Neurocomputing*, 64.

Su, C.-J. (2013). Personal rehabilitation exercise assistant with Kinect and dynamic time warping. *International Journal of Information and Education Technology*, 3(4):448–454.

Suarez, J. and Murphy, R. (2012). Hand gesture recognition with depth images: A review. In *Proceedings of the IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, pages 411–417.

Suay, H. and Chernova, S. (2011). Effect of human guidance and state space size on interactive reinforcement learning. In *Proceedings of IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, pages 1–6.

Sumi, S. (1984a). Model-based vision: a program to see a walking person. *Image and Vision Computing*, 1:5–19.

Sumi, S. (1984b). Upside-down presentation of the johansson moving light-spot pattern. *Perception*, 13:283–302.

Sung, J., Ponce, C., Selman, B., and Saxena, A. (2012). Unstructured human activity detection from RGBD images. In *Proceedings of the International Conference on Robotics and Automation (ICRA)*, pages 842–849.

Sur, M. and Leamey, C. A. (2001). Development and plasticity of cortical areas and networks. *Nature Reviews Neuroscience*, 2:251–262.

Sutton, R. and Barto, A. (1998). Reinforcement learning: An introduction. In *A Bradford Book*.

Tani, J. (2003). Learning to generate articulated behavior through the bottom-up and the top-down interaction processes. *Neural networks*, 16(1):11–23.

Tani, J. and Nolfi, S. (1999). Learning to perceive the world as articulated: an approach for hierarchical learning in sensory-motor systems. *Neural Networks*, 12(7-8):1131–1141.

Taylor, P., Hobbs, J. N., Burroni, J., and Siegelmann, H. T. (2015). The global landscape of cognition: hierarchical aggregation as an organizational principle of human cortical networks and functions. *Scientific Reports*, 5(18112).

Thirkettle, M., Benton, C. P., and Scott-Samuel, N. E. (2009). Contributions of form, motion and task to biological motion perception. *Journal of Vision*, 9(3):28.

Thomaz, A. and Breazeal, C. (2007). Asymmetric interpretations of positive and negative human feedback for a social learning agent. In *Proceedings of the IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, pages 720–725.

Thornton, I., Pinto, J., and Shiffrar, M. (1998). The visual perception of human locomotion. *Cognitive Neuropsychology*, 15:535–552.

Thornton, I. M., Rensink, R. A., and Shiffrar, M. (2002). Active versus passive processing of biological motion. *Perception*, 31:837–853.

Thurau, C. and Hlavác, V. (2008). Pose primitive based human action recognition in videos or still images. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE Computer Society.

Tinetti, M., Liu, W., and Claus, E. (1993). Predictors and prognosis of inability to get up after falls among elderly persons. *Journal of American Medical Association*, 269(1):65–70.

Torta, E., Cuijpers, R., Juola, J., and van der Pol, D. (2011). Design of robust robotic proxemic behaviour. *Social Robotics*, 7072:21–30.

Torta, E., Werner, F., Johnson, D., Juola, J., and Cuijpers, R. (2014). Evaluation of a small socially-assistive humanoid robot in intelligent homes for the care of the elderly. *J Intell Robot Syst*, 76:57–71.

Troje, N. F. (2002). Decomposing biological motion: a framework for analysis and synthesis of human gait patterns. *Journal of Vision*, 2(5):371–387.

Twiefel, J., Baumann, T., Heinrich, S., and Wermter, S. (2014). Improving domain-independent cloud-based speech recognition with domain-dependent phonetic post-processing. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 1529–1536.

Tyler, S. and Grossman, E. (2011). Feature-based attention promotes biological motion recognition. *Journal of Vision*, 11(10):1–6.

Ugur, E., Nagai, Y., Celikkanat, H., and Oztop, E. (2015). Parental scaffolding as a bootstrapping mechanism for learning grasp affordances and imitation skills. *Robotica*, 33:1163–1180.

Ungerleider, L. and Mishkin, M. (1982). Two cortical visual systems. *Analysis of Visual Behavior. Cambridge: MIT press*, pages 549–586.

Ursino, M., Cuppini, C., and Magosso, E. (2014). Neurocomputational approaches to modelling multisensory integration in the brain: A review. *Neural Networks*, 60:141–165.

Van Rijsbergen, C. J. (1979). Information retrieval. *Butterworth-Heinemann, 2nd edition, London.*

Vangeneugden, J., Pollick, F., and Vogels, R. (2009). Functional differentiation of macaque visual temporal cortical neurons using a parametric action space. *Cerebral Cortex*, 19(3):593–611.

Vavrečka, M. and Farkaš, I. (2014). A multimodal connectionist architecture for unsupervised grounding of spatial language. *Cognitive Computation*, 6(1):101–112.

Velloso, E., Bulling, A., Gellersen, G., Ugulino, W., and Fuks, G. (2013a). Qualitative activity recognition of weight lifting exercises. In *Augmented Human International Conference (ACM)*, pages 116–123.

Velloso, E., Bulling, A., and Gellersen, H. (2013b). MotionMA: Motion modelling and analysis by demonstration. In *Proceedings of the SIG-CHI Conference on Human Factors in Computing Systems*, pages 1309–1318, New York, NY, USA. ACM.

Vettier, B. and Garbay, C. (2014). Abductive agents for human activity monitoring. *International Journal on Artificial Intelligence Tools*, 23.

Voegtlin, T. (2002). Recursive self-organizing maps. *Neural Networks*, 14(3):39–71.

Volkhardt, M. and Gross, H.-M. (2013). Finding people in home environments with a mobile robot. *European Conference on Mobile Robots (ECMR), Barcelona, Spain*, pages 282–287.

Volkhardt, M., Schneemann, F., and Gross, H.-M. (2013). Fallen person detection for mobile robots using 3D depth data. *Proc. IEEE Int. Conf. on Systems, Man, and Cybernetics (IEEE-SMC)*, pages 3573–3578.

Voronoi, G. (1907). Nouvelles applications des paramétres continus á la théorie des formes quadratiques. *Journal für die Reine und Angewandte Mathematik*, 133:97–178.

Willshaw, D. J. and von der Malsburg, C. (1976). How patterned neural connections can be set up by self-organization. *Proceedings of the Royal Society of London B: Biological Sciences*, 194(1117):431–445.

Wong, R. O. L. (1999). Retinal waves and visual system development. *Annual Review of Neuroscience*, 22:29–47.

Wright, T. M., Pelphrey, K. A., Allison, T., Mckeown, M. J., and Mccarthy, G. (2003). Polysensory interactions along lateral temporal regions evoked by audiovisual speech. *Cerebral Cortex*, 13(10):1034–1043.

Xu, R., Agarwal, P., Kumar, S., Krovi, V., and Corso, J. (2012). Combining skeletal pose with local motion for human activity recognition. *Articulated Motion and Deformable Objects, Mallorca, Spain: Springer Berlin Heidelberg*, pages 114–123.

Yan, W., Torta, E., van der Pol, D., Meins, N., Weber, C., Cuipers, R., and Wermter, S. (2013). Learning robot vision for assisted living. In *Robotic Vision: Technologies for Machine Leaning and Vision Applications, ch. 15, IGI Global*, pages 257–280.

Zhang, C. and Tian, Y. (2012). RGB-D camera-based daily living activity recognition. *Journal of Computer Vision and Image Processing*, 2(4):1–7.

Zhou, H. H. (1990). Csm: A computational model of cumulative learning. *Machine Learning*, 5(4):383–406.

Zhu, Y., Chen, W., and Guo, G. (2014). Evaluating spatiotemporal interest point features for depth-based action recognition. *Image and Vision Computing*, 32:453–464.

# Declaration on Oath

## *Eidesstattliche Versicherung*

I hereby declare on oath, that I have written the present dissertation by my own and have not used other than the acknowledged resources and aids.

*Hiermit erkläre ich an Eides statt, dass ich die vorliegende Dissertationsschrift selbst verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe.*

Hamburg, November 10, 2016
City and Date *(Ort und Datum)*

German I. Parisi
Signature *(Unterschrift)*

# Erklärung zur Veröffentlichung

Ich erkläre mein Einverstndnis mit der Einstellung dieser Dissertation in den Bestand der Bibliothek.

Hamburg, November 10, 2016
Ort, Datum

German I. Parisi
Unterschrift