Fortschritt-Berichte VDI

VDI

Reihe 8

Mess-,
Steuerungs- und
Regelungstechnik

Nr. 1251

Dipl.-Ing. Thomas Guthier,
Frankfurt am Main

Visual Motion Processing

Berichte aus dem

Institut für
Automatisierungstechnik
und Mechatronik
der TU Darmstadt

Fortschritt-Berichte VDI

Reihe 8

Mess-, Steuerungs- und Regelungstechnik

Dipl.-Ing. Thomas Guthier, Frankfurt am Main

Nr. 1251

Visual Motion Processing

Berichte aus dem

Institut für Automatisierungstechnik und Mechatronik der TU Darmstadt

Guthier, Thomas
Visual Motion Processing
Fortschr.-Ber. VDI Reihe 8 Nr. 1251. Düsseldorf: VDI Verlag 2016.
166 Seiten, 61 Bilder, 12 Tabellen.
ISBN 978-3-18-525108-5, ISSN 0178-9546,
€ 62,00/VDI-Mitgliederpreis € 55,80.

Keywords: Human Action Recognition – Computational Neuro-science – Computer Vision – Machine Learning – Deep Learning

The capability to recognize biological motion, i.e. gestures, human actions or face movements is crucial for social interactions, for predators, prey or artificial systems interacting in a dynamic environment.
In this thesis an artificial feed-forward neural network for biological motion recognition is proposed. Like its natural counterpart, it consists of multiple layers organized in two streams, one for processing static and one for processing dynamic form information. The key component of the proposed system is a novel unsupervised learning algorithn, called VNMF, that is based on sparsity, non-negativity, inhibition and direction selectivity.
In the first layer of the dorsal stream, the VNMF is modified to solve the optical flow estimation problem. In the subsequent layer the VNMF algorithm extracts prototypical patterns, such as optical flow patterns shaped e.g. as moving heads or lim parts. For the ventral stream the VNMF algorithm learns distict gradient structures, resembling edges and corners. All these patterns represent simple cells of the feed-forward hierachy. The classification performance of the feed foward neural network is analyzed on three real world datasets for human action recognition and one face expression recognition dataset, achieving results comparable to current computer vision approaches.

Bibliographische Information der Deutschen Bibliothek
Die Deutsche Bibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliographie; detaillierte bibliographische Daten sind im Internet unter http://dnb.ddb.de abrufbar.

Bibliographic information published by the Deutsche Bibliothek
(German National Library)
The Deutsche Bibliothek lists this publication in the Deutsche Nationalbibliographie (German National Bibliography); detailed bibliographic data is available via Internet at http://dnb.ddb.de.

D 17

ISSN 0178-9546
ISBN 978-3-18-525108-5

Visual Motion Processing

Vom Fachbereich
Elektrotechnik und Informationstechnik
der Technischen Universität Darmstadt
zur Erlangung des akademischen Grades eines
Doktor-Ingenieurs (Dr.-Ing.)
genehmigte Dissertation

von

Dipl.-Ing. Thomas Guthier

geboren am 16. Juni 1983 in Heppenheim/Hessen

Referent:	Prof. Dr.-Ing. J. Adamy
Korreferent:	Prof. Dr.rer.nat. B. Sendhoff
Tag der Einreichung:	15. April 2015
Tag der mündlichen Prüfung:	14. October 2015

D17
Darmstadt 2015

Danksagung

Mein größter Dank gilt meiner Familie, welche bei allen meinen Entscheidungen stehts bedingungslos hinter mir steht. Diese Sicher- und Geborgenheit hat mir die angemessene Zeit gegeben, welche eine Dissertation von jedem einfordert.
Mein nächster Dank gilt Julian Eggert und Volker Willert, von denen ich alles wichtige gelernt habe was ich über Bildverarbeitung und Lernalgorithmen weiß. Ihr beider Anteil an dieser Arbeit ist in gewisser Weise größer als der Meinige.
Prof. Adamy, Prof. Sendhoff und Prof. Körner danke ich für ihr Vertrauen und das sie die vorliegende Arbeit überhaupt erst ermöglicht haben. Ich danke Prof. Adamy dafür, dass er meine Ausbildung, fachlich und darüber hinaus, gefördert hat.
Den Kaffeepausen und den daran teilnehmenden Kollegen danke ich für unbezahlbare Unterhaltungen. Letztendlich dafür, dass ich mich nahzu an jedem Morgen auf die Arbeit freuen durfte.
Dann wären da noch meine Studenten, welche mit ihren Ideen und ihrer investierten Zeit wesentlich zum gelingen dieser Dissertation beigetragen haben. Vielen dank dafür, dass ihr mir die Betreuung eurer Abschlussarbeit anvertraut habt.
Bei der Natur bedanke ich mich dafür, dass sie so etwas komplexes wie das menschliche Gehirn hat entstehen lassen. Ein spannenderes Themenfeld als dieser schwer definierbare Raum zwischen Biologie und vereinfachter, mathematisch-technischer Algorithmik hätte ich mir nicht erträumen können.

Es ist schon erstaunlich wieviele Menschen letztendlich solch eine Arbeit beeinflussen. Mein letzter Dank gilt all denen welche sich oben eventuell nicht widerfinden, aber die auf ihre eigene Art meine Dissertation beeinflusst haben. Im speziellen...
Isabell, Marlene, Dominik Haumann, Erich Lenhardt, Jochen Grießer, Andreas Lutz, Thorsten Graber, Valentina Ansel, Adrian Šošić, Nikola Aulig, Andrea Schnall, Matthias Platho, Moritz Schneider, Kim Listmann, Dieter Lens und viele andere ...

Contents

Abbreviations

BCA	Brightness Consistancy Assumption
BCE	Brightness Consistancy Equation
BOW	Bag of Words
EBA	Extratiate Body Area
FFNN	Feed Forward Neural Network
FER	Facial Expression Recognition
HAR	Human Action Recognition
HOG	Histogram of Oriented Gradients
HOF	Histogram of Optical Flow
HS	Horn and Schunk
ICA	Independent Component Analysis
ISA	Independent Subspace Analysis
IT	Inferior Temporal
LK	Lukas and Kanade
MT	Middle Temporal
MST	Medial Superior Temporal
NMF	Non-negative Matrix Factorization
OF	Optical Flow
OFE	Optical Flow Estimation
PCA	Principal Component Analysis
SC	Sparse Coding
sNMF	Sparse Non-negative Matrix Factorization
sNN-LDS	Sparse Non-negative Linear Dynamic Systems
STSp	Posterior Superior Temporal Sulcus
SVM	Support Vector Machine
V1	Primary Visual Cortex
VNMF	Vector Non-negative Matrix Factorization

Abstract

The capability to recognize biological motion, *i.e.* gestures, human actions or face movements is crucial for social interactions, for predators, prey or artificial systems interacting in a dynamic environment. The famous point-light-walker experiments [58] reveal that humans have a highly skilled mechanism dedicated to the analysis of motion information, however the exact details of this mechanism remain largely unclear. A popular theory is, that visual recognition is performed in a hierarchical feed-forward process, consisting of multiple learned simple cell/complex cell layers [53]. In the case of biological motion recognition these layers are spread throughout the ventral and dorsal stream of the visual cortex, the ventral stream being dedicated to static visual information, such as spatial gradient structures and the dorsal stream is related to dynamic visual information, such as the motion for each pixel in the input, also known as the optical flow.

In this thesis an artificial feed-forward neural network for biological motion recognition is proposed. Like its natural counterpart, it consists of multiple layers organized in two streams, one for processing static and one for processing dynamic form information. The key component of the proposed system is a novel unsupervised learning algorithm, called VNMF, that is based on sparsity, non-negativity, inhibition and direction selectivity.

In the first layer of the dorsal stream, the VNMF is modified to solve the optical flow estimation problem. In the subsequent layer the VNMF algorithm extracts prototypical patterns, such as optical flow patterns shaped *e.g.* as moving heads or limb parts. For the ventral stream the VNMF algorithm learns distinct gradient structures, resembling edges and corners. All these patterns represent the simple cells of the feed-forward hierarchy, while the complex cells are modeled by a non-linear maximum pooling operation.

The classification performance of the feed forward neural network is analyzed on three real world datasets for human action recognition and one face expression recognition dataset, outperforming other biological inspired models while being competitive with current computer vision approaches.

Kurzfassung

Gesten, Mimiken und andere natürliche Bewegungen sind ein wesentlicher Bestandteil zwischenmenschlicher Kommunikation. Darüber hinaus ist die visuelle Wahrnehmung von Bewegungen notwendig um sich in einer sich stetig verändernden Umgebung zurechtzufinden. Die berühmten *Point-Light-Walker* Experimente von Johansson [58] zeigen, dass Menschen Bewegungen auch ohne klar definierte Formen wahrnehmen können. Allerdings ist es nach wie vor unklar wie die Bewegungsinformationen im menschlichen Gehirn verarbeitet werden. Eine populäre Theorie [53] besagt, dass visuelle Informationen in aufeinander folgenden, gelernten Neuronen-schichten verarbeitet werden. Im Fall der visuellen Bewegungsanalyse sind die Schichten im ventralen und dorsalen Pfad des visuellen Kortex verteilt. Der ventrale Pfad verarbeitet statische, z.B. Kanten, Informationen, während der dorsale Pfad eher dynamische Informationen, z.B. Punktbewegungen, auch optischer Fluss genannt, verarbeitet.

In der vorliegenden Dissertation wird ein künstliches neuronales Netzwerk zur Erkennung von natürlichen Bewegungen vorgestellt, welches, dem biologischen Vorbild gleich, aus zwei parallelen Pfaden besteht. Die Schlüsselkomponente des vorgestellten Systems ist ein neuer Lernalgorithmus, welcher die neuronalen Verbindungen der verschiedenen Schichten ausschließlich anhand von Beobachtungen lernt. Die Kodierung der Bewegungsinformation erfolgt richtungsspezifisch anhand von spärlichen, nichtnegativen Aktivitäten, welche mit anderen Aktivitäten in ihrer lokalen Nachbarschaft konkurrieren. In der ersten Schicht des dorsalen Pfades wird das optische Flussfeld mit Hilfe des neuen Lernalgorithmus geschätzt. In der darauf folgenden Schicht werden prototypische Muster gelernt, deren Formen bewegliche Körperteile beschreiben. Im ventralen Pfad wird der VNMF Algorithmus verwendet um Kantenstrukturen zu lernen.

Die Klassifikationseigenschaften des neuronalen Netzes werden anhand von drei Datensätzen für Körper- und Gesichts-bewegungen evaluiert. Die Klassifikationsergebnisse des vorgestellten Systems sind genauer als die anderer biologisch inspirierter Modelle und vergleichbar mit aktuellen Modellen der Bildverarbeitung.

1 Introduction

The human visual cortex, *i.e.* the part of the human brain that processes visual information, solves multiple tasks, such as transforming the incoming light-waves to a set of spatio-temporally arranged objects, like trees, books, clouds, as well as recognizing and understanding complex movements, such as gestures or facial expressions, a task known as *biological motion recognition*. The ability to *see* is mainly learned after birth, newborn children have a very blurry vision without a clear focus. Thus, the neural network of the visual cortex has to be adapted to its surroundings and stays adaptable throughout our life. The core idea of this thesis is to analyse the principles underlying the learning mechanisms in the visual cortex and to apply them to an artificial counterpart: a feed-forward neural network for biological motion recognition.

How does the visual cortex process the visual information and how is learning performed in the brain? While the exact methods of the brain are far from being understood, a common theory is that visual information is processed throughout a hierarchy of multiple layers. For each layer the input is decomposed into a set of patterns, just like a puzzle, where each piece, *i.e.* pattern, represents a specific part of the input. Throughout the hierarchy the patterns get larger in size and increasingly object specific. The patterns in the first layer might represent small generic structures, like corners or edges, the mid-level structures group these patterns and thus represent increasingly complex structures, like object parts. In the final layer, there are object specific patterns which are only activated if the specific object is presented in the input.

How are these patterns learned? In principle there exist three types of learning concepts, *supervised learning*, *reinforcement learning* and *unsupervised learning*. For supervised learning a "teacher" is required, *e.g.* label information that tells the algorithm which class an input belongs to. Reinforcement learning does not require such a label, but a penalty (negative reinforcement) or a bonus (positive reinforcement) signal. This information has to be provided from outside and that is why both supervised and reinforcement learning are not fully self-reliant processes.

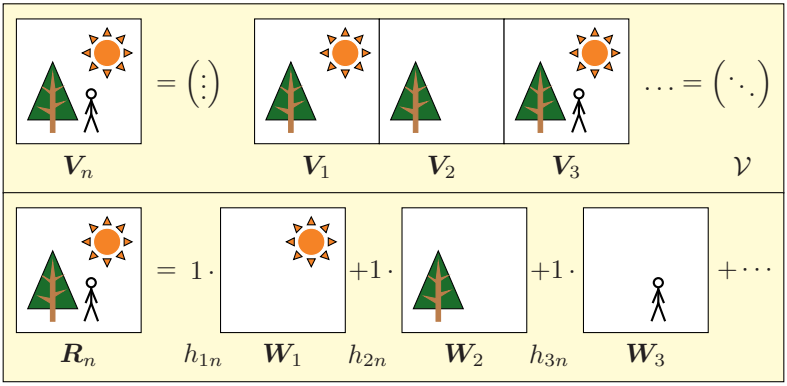

Figure 1.1: Illustrative example of the pattern learning task. Upper row: input images showing a tree, the sun and a stick figure person. The pixel values of an input image can be stored in a single vector $\boldsymbol{V}_n$. Multiple images are stored in the input matrix $\mathcal{V}$. The lower row shows a learned model $\boldsymbol{R}_n$. The model is a linear superposition of learned basis patterns $\boldsymbol{W}_1$, $\boldsymbol{W}_2$, ..., $\boldsymbol{W}_j$ each with an activation h_{jn}. The example shows a desirable decomposition into prototypical parts which can occur in different combinations to form the given input images. Each basis pattern represents a specific input part, *i.e.* a tree, sun, person. This kind of parts-based decomposition is desirable, because the corresponding activations indicate whether the specific pattern is present in the input. For a parts-based decomposition the activations give a meaningful, interpretable description of the input.

Contrariwise unsupervised learning relies only on self-organization and is often termed *learning without a teacher*. In this thesis, unsupervised learning is modeled as a generative process. An arbitrary initialized model creates a reconstruction that is compared to input data and the difference between the generated reconstruction and the input is minimized by updating, *i.e.* learning the model parameters. This concept is illustrated in fig. 1.1 and discussed in the following.

The basic idea is that each of the inputs is reconstructed by a superposition of basic patterns, where each basic pattern is weighted by a corresponding activation. If the basic patterns represent prototypical input parts they are able to reconstruct the input and the corresponding activations indicate whether this specific part is active in the input. A parts-based decomposition is thus on the one hand generic, *i.e.* the parts

can be used to reconstruct a large variety of input data. On the other hand, the patterns are discriminative, because the corresponding activations indicate the presence of specific part in the input. They are therefore useful features for a classification hierarchy.

In short, the goal for these learning algorithms in this thesis is to learn prototypical, parts-based patterns in an unsupervised fashion. However, there exist multiple solutions to this particular unsupervised learning problem, most of which do not include prototypical parts, but focus on the generic ability. It is thus important to find learning principles that favor parts-based decompositions. Fig. 1.2 illustrates the principles for the learning algorithms that are presented in this thesis:

- non-negativity,
- sparse activations,
- inhibition.

All three can be motivated by the properties of neural processing as observed in the visual cortex and enforce a parts-based encoding of the input. Neural activations are always strictly non-negative. A non-negative encoding for unsupervised learning was first proposed as *Positive Matrix Factorization* in 1994 by Paatero and Tapper [78] and became famous in 1999 when introduced as *Non-negative Matrix Factorization* (NMF) by Lee and Seung [66]. In 1997 Olshausen and Fleet showed in [77] that using a sparse decomposition on natural image patches results in patterns similar to those found in the primary visual cortex (V1). While neural activations cannot have negative values, they can inhibit other activations, making inhibition and not subtraction a central property of neural processes. Another neural coding principle exploited in this thesis is having

- direction selective

representations. In case of vectorial data, describing *e.g.* movement directions, a non-negative representation can be enforced by using a direction selective encoding. *I.e.* each movement direction has its own non-negative representation. Following this idea allows for a fully non-negative model even when the input contains positive and negative values.

In this thesis novel unsupervised learning algorithms based on the idea of having a *generic* and *prototypical, parts-based* decomposition are proposed. It is shown that these algorithms, combined in a multi-layer classification hierarchy, are well suited to solve visual biological motion recognition tasks.

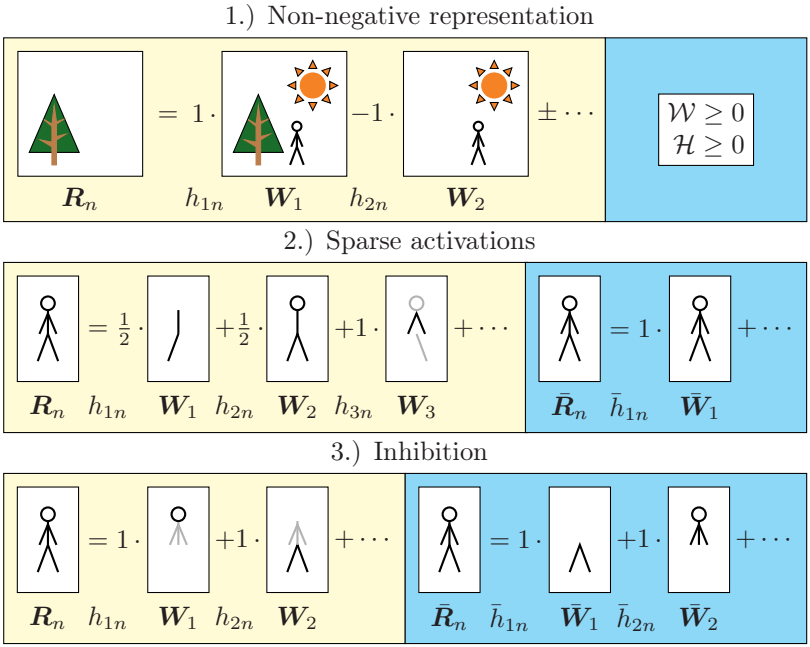

Figure 1.2: Visualization of the desired properties for the learning algorithm. 1.) Non-negativity: a model that allows subtractions (yellow box) can achieve a reconstruction using combination of holistic patterns and will not result in parts-based decomposition. 2.) Sparse activations: examples for a non-sparse (yellow box) and a sparse (blue box) decomposition. Sparsity favors large patterns that group input parts that occur together. 3.) Inhibition: the activation of overlapping patterns (yellow box) is penalized, which results in prototypical patterns (blue box).

1.1 Biological Motion Recognition

The capability to recognize complex motions, such as gestures, human actions or face movements is crucial for social interactions, for predators, prey or artificial systems interacting in a dynamic environment. The famous point-light-walker experiments of Johansson in 1973 [58] revealed that humans have a highly skilled mechanism dedicated to the analysis of motion information. He introduced the term *biological motion*, because the first test scenarios for the point-light-walkers included the distinction

Figure 1.3: Example applications. Upper row, from left to right: Detection and recognition of traffic participants, detection of fights in public places, touch-free human-machine interactions in a sterile operation setup. Lower row, from left to right: Gesture and action recognition for entertainment systems and two examples for non-verbal communications, *i.e.* facial expressions and gestures.

between human, thus biological, and artificial movements. Motivated by a recent discussion about the brain's capability to recognize a large variety of different articulated movements [18], in this thesis, the term biological motion is used to refer to any form of articulated movement coming from humans, animals or artificial systems, in contrast to ego-motion induced global optical flow fields [79].

The recognition of biological motion is a critical task of the human visual cortex, because it is very important in human-human interactions. This makes motion recognition very interesting for artificial systems as well, since it could improve human-machine interaction in several areas. The applications of motion recognition can be roughly split into three categories: First, *communication*, such as the recognition of *gestures* or *facial expressions*. Second, *detection*, *e.g.* people or bicyclist detection, which both have very specific movement patterns, like the opposite movement of the legs during walking. The third category is *scene understanding*, for example *human action recognition*. The idea of human action recognition is to not only detect people or identify a specific person, but to get an understanding of what the person is doing: Is he walking, running, fighting, etc. The term *actions* corresponds to small, possibly repetitive movements, such as running, walking, punching, kicking, jumping, *a.s.o.* More complex move-

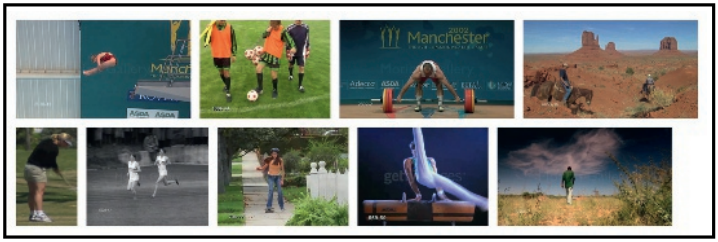

Figure 1.4: Examples for the nine action classes in the UCF Sports dataset [88].

ments, such as a combination of the simple actions, thus action sequences, are not discussed in this thesis, but can be thought of as concatonations of prototypical movement patterns. The recognition of motion is important to get an understanding of what is happening in a scene. This is useful, for example to detect a panic in a crowd, a fight in an elevator or at public places, such as train stations. Some application examples are illustrated in fig. 1.3.

Due to the temporal component, biological motion recognition encounters a large set of variations that are now discussed for the challenging UCF Sports human action recognition dataset [88]. Fig. 1.4 depicts examples of the nine action classes represented in the dataset: *diving*, *kicking*, *weight lifting*, *horse riding*, *golfing*, *running*, *skateboarding*, *gymnastics* and *walking*.

Figure 1.5: Two example sequences of the *kicking* action. The action consists of a sequence of poses and can have strong variations in its execution.

Figure 1.6: Upper row: multiple examples of the same pose of the *golfing* action. Lower row: examples of the same pose of the *running* action. Each pose can have strong view-point variations as well as strong variations in the appearance of the person performing the action.

1.1.1 Temporal and View-Point Variations

Fig. 1.5 shows two example sequences of the *kicking* action. For the same action the involved poses can be very different and the pose sequence can vary as well. This makes the problem difficult, because videos with a very low degree of similiarity have to be grouped together, while videos that share *e.g.* identical poses or even identical sub-actions (see fig. 1.7) must stay in different class categories.

Even for classes with very distinct poses, like *golfing*, there are view-point variations as depicted in fig. 1.6. The classification has to cope with variations in scale, the viewing angle, texture variations, varying backgrounds and lighting conditions.

Figure 1.7: Left: nearly identical poses of different actions. Right: almost identical pose sequence for two different actions.

Figure 1.8: Left: class specific key poses. Right: class specific sequences.

1.1.2 Discriminative Features

A major difficulty lies in finding the discriminative features for the different actions. While some actions are very easy to differentiate, because they have almost nothing in common, like *e.g. diving* and *weight lifting*, other actions are very hard to differentiate. They share common poses and in some cases even common sub-actions, like *e.g. running* and *kicking* (see fig. 1.7). However, the same actions can have specific *key poses* or very class specific sub-actions at a differnt point in time (see fig. 1.8).

Since it is not possible to say whether a class is specific because of small local features, or a full body pose or even a pose sequence it is best to use unsupervised and *not supervised* learning for the feature extraction. The supervised learning of the classification should be as late in the hierarchy as possible as to maximize the systems level of self-organization.

1.2 Computational Models for Biological Motion Recognition

Biological motion recognition includes multiple and diverse fields, such as biologically inspired computational models and application driven computer vision algorithms. In computer vision biological motion recognition is often divided into *human action*, *gesture* and *facial expression* recognition. A review of proposed approaches in computer vision can be found *e.g.* in [1]. One of the currently most successful approaches, the *Bag of Words* (BOW) is discussed in appendix A.

Classification systems in computer vision typically consist of two steps: First the feature extraction and second the classifier. The classifier is

a supervised learning algorithm, while the feature extraction is mainly done via hand-designed features like the SIFT [70] or HOG/HOF [21] descriptors. One important contribution of this thesis is the comparison of novel, learned, pattern-based descriptors to the HOG/HOF descriptors.

1.2.1 Computational Neuroscience

Biologically inspired models are motivated by the capability of the brain to solve complex tasks like biological motion recognition. In addition these models can help to understand how the brain actually solves specific problems. In computational neuroscience the goal is to find computational models that can help explain the observations of neurophysiological experiments. A popular example is the primary visual cortex (V1). Neurophysiological experiments show that V1 consists of multiple cell populations which work similar to Gabor filters of varying size and orientation. Models from computational neuroscience try to find learning algorithms that learn similar filters when presented with natural images [51, 77].

The computational model presented in this thesis follows this idea and applies it to the field of biological motion recognition. The point-light-walker experiments[1)] show that humans can recognize biological motion even without explicit form information. These observations started an ongoing discussion on how form and motion contribute to the recognition process. While neurophysiological experiments, *e.g.* discussed in [18, 37–40, 100], indicate the importance of both, form and motion information, there are several open questions, *e.g.* the role of explict low-level motion information, such as optical flow [48, 98, 112, 114]. The optical flow explicitly describes the movement of each pixel and is itself not selective to form. However, by grouping parts with consistent movements, like an upper-arm or a torso, the spatial configuration and the movement direction of these parts can be used to identify characteristic motion patterns. Early computational models propose the use of optical flow patterns, *e.g.* in a hierarchical manner [38, 56]. To the contrary, motivated by lesion experiments of a patient whose early motion processing areas are impaired but who could nevertheless recognize biological motion, [62, 100] suggest that low-level motion plays no major role in the recognition of biological motion. In their related proposed model, motion is only incorporated on a higher level, as the transition between full body poses. Their model is in good accordance with neurophysiological experiments which indicate that

[1)] Discussed in more detail in section 2.1.

early motion processing areas of the brain may not be involved in biological motion perception [4, 93]. However, low-level motion information improves the recognition in the presence of noise [4], which hints to an involvement of low-level motion. As suggested in [62], optical flow could be used to segment the moving person from the background or, as discussed in [37], the spatial configuration of mid-scale optical flow patterns could be used as a way to describe the human body form alongside with static shape or gradient information. Thus, body postures can be defined by the spatial configuration of two, possibly redundant, types of information: static and dynamic, *e.g.* gradient and optical flow patterns.

Similar to [32, 38, 56, 62, 100], this thesis contributes to the ongoing discussion on a *functional level* by presenting a computational model that consists of two complementary streams for motion information processing, one for dynamic and one for static form information. Except for the final layer, all layers of the proposed model consist of features gained by unsupervised learning algorithms, based on the idea of non-negativity, sparsity, inhibition and direction selectivity. The classification performance of the individual and combined streams is examined in complex real world scenarios to analyze how low-level motion, *i.e.* optical flow fields, can contribute to the recognition of human actions. The presented model is further compared to related models based on computer vision algorithms.

1.3 Summary & Thesis Structure

The contributions in this thesis can be summarized as follows:

1. A novel feed forward hierarchical system for biological motion recognition is proposed.

2. The central components of this system are novel unsupervised learning algorithms for optical flow estimation (VNMF-OFE) and feature extraction (VNMF), based on the idea of non-negativity, sparsity, inhibition and a direction-selective encoding.

3. The contribution of static and dynamic form descriptors and thus the influence of low-level motion descriptors for biological motion recognition is analysed and discussed.

4. The analysis further includes an evaluation of the proposed system on computer vision datasets, including a comparison of the proposed learned feature descriptors to state-of-the-art HOG/HOF descriptors,

other pattern learning methods and a comparison of the overall system to related biologically inspired models and state-of-the-art computer vision models.

The thesis is structured as follows: First, the proposed biological motion recognition system is introduced along with its neural counterpart. Next, the mathematical foundations of the related and novel unsupervised learning algorithms are proposed. This includes the VNMF algorithm that plays a central role for the feature extraction in the recognition system. In the following chapter optical flow estimation is reviewed and a modified version of the VNMF for optical flow estimation, the VNMF-OFE algorithm, is introduced and analyzed. The analysis focuses on how robustly the novel algorithm can preserve small but important moving structures. After that, the VNMF is applied as a feature extraction method on the estimated optical flow fields of human actions. In addition, the VNMF is applied on gradient amplitudes of the same data and the learned descriptors are compared to state-of-the-art HOG/HOF [21] descriptors. In the subsequent chapter the classification performance of the learned optical flow and gradient descriptors and their individual contribution is analyzed. The descriptors are further compared to the HOG/HOF descriptors and related work on the Weizmann human action [8], UCF-Sports [88] and a facial expression recognition benchmark.

2 Computational Model

The proposed biological motion recognition system is highly inspired by the processing of biological motion in the human visual cortex. In this chapter neurophysiological experiments along with a functional analysis of motion recognition in general are discussed. The discussion includes questions like how motion is represented in the brain, *i.e.* whether small movements are explicitly represented or whether biological motion is only analyzed as a sequence of distinct body poses. In addition, the contributions of static and dynamic information for the recognition is analyzed.

Based on the review of biological motion recognition in the brain, the proposed biological motion recognition system is introduced as a *Feed Forward Neural Network* (FFNN) that consists of two processing streams, one for static and one for dynamic, *i.e.* explicit motion information. The networked architecture is inspired by the idea that the visual cortex contains two streams a *what* and a *where* stream. The ventral stream that processes static information is considered to awnser what is in the image and the dorsal stream that processes motion information locates the object. This thesis differs concerning this classical two stream, because motion information is used in combination with static information to determine *what* is in the image.

2.1 Biological Motion Recognition in the Brain

It is without question that the human brain has developed a remarcable capability of recognizing complex biological movements. However, the underlying neural processes that guide visual motion recognition are far from being understood. There are approximately 4-6 billion neurons in the human visual cortex, which cover $\sim 20\%$ of the cortical surface [106]. This highly complex system solves multiple vision tasks that are far too complex to be analyzed in one dissertation. To give an idea of the brain functionality a very rough and simplified summary of different tasks in visual recognition is given in the appendix B. The focus in this thesis is

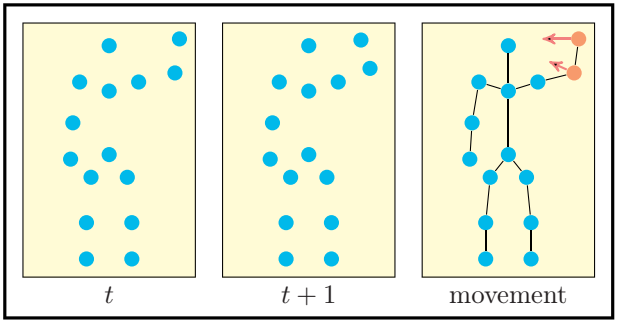

Figure 2.1: Point-light-walker experiments as introduced by Johansson in 1973 [58].

on gradient and motion information and the contribution of both channels to biological motion recognition, including the subtasks of optical flow estimation, feature extraction for optical flow and gradient information. In the following a short review of neurobiological research on biological motion recognition is given.

Research on motion processing in the brain dates back to experiments by Exner in 1887 [30], who first described a motion specific perceptual effect. In his experiments, two points are set so close to each other that their spatial location cannot be distinguished. If they change their illumination over time, human observers can recognize a movement. Because the movement cannot be tracked back to the exact spatial dot location, the *change over time*, *i.e.* motion, must be directly measured. A brief review over this experiment and the history of motion recognition is given in [95].

The current development is highly influenced by the famous point-light-walker experiments which were first introduced by Johansson [58] in 1973. The experiments show that humans can extract various information about a person, such as the gender, the physical fitness, and even emotional states, all while whatching only lighted dots, without an explicitly represented shape or texture, as depicted in fig. 2.1. The recognition fails if the points are shown as a static image. Even though point-light stimuli are hardly found in a natural environment, the observed recognition performance highlights the importance of time-varying visual stimuli for the recognition of biological motion.

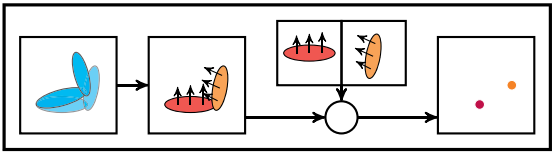

Figure 2.2: From left to right: First, an image of a moving arm for two consecutive time steps. Second, the corresponding optical flow field of the articulated arm movement. Third, two specific optical flow patterns, each for an area with coherent motion. Last, the position where each pattern has to be placed to reconstruct the incoming optical flow field. The forearm and the upper arm each have a coherent, but different translational movement and can thus be differentiated based on the related optical flow pattern.

2.1.1 Different Motion Representations

It is not clearly defined on which layer of abstraction motion should be explicitly described. On a small scale every pixel has its own movement, which is called the *optical flow*. On a large spatial scale there are *pose sequences*, which itself are already abstract descriptions and not known beforehand. Pose sequences are way more specific than pixel movements and therefore should be handled in a different manner.

The two motion types are different on a functional level as well. On the larger spatial scale, motion is defined by pose sequences. Poses consist of body part configurations, that form groups, or again parts, whose occurrence and relative spatial relation define the pose. Actions can be defined by either the temporal sequence of the *full body pose* or by temporal sequences of *pose parts*, *i.e.* body part configurations. The first approach has the advantage that each pose sequence is explicitly represented, which makes the sequences easy to classify. This approach is quite suited when the number of distinct pose sequences is limited. For a more general machinery that is designed to represent and recognize a large variety of human actions, gestures or facial expressions, a parts-based approach is more feasible, because the common parts can be used in different combinations to describe complex motion. A model and a learning algorithm for a non-negative pose sequence description are discussed in appendix D and [45]. The focus of this thesis is on temporally short actions, thus the small scale motion in form of optical flow fields is more important than a specific pose sequence description.

The small scale optical flow is very useful in abstracting relations between pixels and thus relate motion with forms. *E.g.* the movement of rigid objects on a small spatial and time scale can be described by affine or, as a simplification, translational motion. Fig. 2.2 shows an articulated arm movement where the forearm has a different movement than the upper arm. The related optical flow field can be used to differentiate between these two forms, by grouping optical flow vectors which exhibit identical translational motion. Thus, regions with coherent motions, or *parts* of the optical flow field are related to moving rigid body parts, *i.e.* the form of body parts. In addition, the direction of the movement might be class-specific, since *e.g.* walking mainly consists of horizontal motion, while jumping has more vertical components. Finally, the velocity can be class-specific as well, *e.g.* to distinguish walking from running or hand-shaking from boxing. In conclusion, small scale motion, *i.e.* optical flow, is extremely powerful in providing features for biological motion recognition.

The small scale motion description therefore has two parts. First, the estimation of the optical flow and second, finding optical flow features, *e.g.* parts-based optical flow patterns. The optical flow as well as the patterns have to be generic, because they must be able to represent all kinds of observed movements.

2.1.2 Static and Dynamic Form Description

For a biological motion recognition system it is important to find a robust description for body or face part configurations, *e.g.* via form descriptors. Optical flow describes motion and by grouping parts with coherent movement, thus extracted optical flow patterns can be used to describe form information. Since the optical flow patterns simultaneously describe form and motion, they are termed *dynamic form patterns*. Form can also be defined by spatial gradient patterns, which in the following are termed *static form patterns*.

As discussed in [37], the spatial configuration of optical flow patterns can be used as a way to describe the human body form, alongside with static shape or gradient information. Thus, body postures can be defined by the spatial configuration of two, possibly redundant, types of information: static and dynamic, *e.g.* gradient and optical flow patterns.

2.1.3 Neurophysiological Experiments

While the point-light-walker experiments shed a light on the amazing motion recognition performance of the human brain, neurophysiological experiments try to identify the brain areas that are involved in the recognition process. There exist a vast amount of experiments, discussed *e.g.* in [7, 39, 59, 82, 83, 101], where it is shown that multiple areas in the visual cortex are involved in the recognition of biological motion. They are located in the two main processing streams in the visual cortex: In the *ventral* stream that is related to static object recognition and the *dorsal* stream dealing with position and motion specific information. The extent to which each area contributes in which form to the recognition of biological motion is a vividly discussed topic. One result that is commonly agreed upon, is that the *posterior superior temporal sulcus* (STSp) plays an important role and is to some extent specialized to human actions [7, 55, 59, 82, 83].

A question that is much less understood and that is related to the functional discussion about motion representation in the previous section is: How do form and motion contribute to biological motion recognition?

One theory is that motion is only relevant on a global scale, *i.e.* that global shapes over time are sufficient to represent biological motion [62, 100]. This idea is contradictory to other experiments [35] in which the authors state that explicit motion is required for biological recognition. However, they assume that form and motion are at some level integrated or that there exist some sort of higher-order motion cues. Another idea is that form and motion have individual contributions to biological motion recognition [101]. This is in good accordance with the functional discussion in section 2.1.1.

Another interesting relation is the effect of *implied motion* [55] observed in biological motion recognition. The authors in [55] show that static stimuli alone can trigger the motion selective areas. The hypothesis is that there may exist a top-down mechanism that subsequently activates the motion selective areas after the static stimuli activated a *combined representation*. The benefit of such a combined representation and the effect of the implied motion is used in computational models based on *flow-object* analysis [43, 68].

2.1.4 Brain Areas based on Neurophysiological Experiments

Motion processing in humans starts as early as on the retina. An overview of recent experimental results and computational models for early motion

processing is given in [9]. A common feature for all early motion processing is *direction selectivity*, thus the ability to differentiate between the movement in distinct spatial directions [9, 49, 59].

The signals from the retina are sent to the *primary visual cortex* (V1), where, amongst other cells, there are direction selective cell populations as well as neurons sensitive to motion boundaries [59, 86]. Most of the cell populations in V1 are similar to Gabor filters of different orientations and frequencies, thus they perform operations similar to spatial gradient calculation.

V1 is connected to multiple other visual processing areas, including the motion sensitive areas of the dorsal stream and the static ventral stream. The dorsal stream inculdes the hMT+ complex (area *middle temporal* (MT) and *medial superior temporal* (MST)) and the *posterior superior temporal sulcus* (STSp). A theory is that the first levels contain a generic motion processing, which gets more class specific throughout the hierarchy. In between the generic early motion processing areas V1 and MT and the already class specific neurons in STSp, there might exist mid-level motion patterns related to limb forms and limb movement. The ventral stream processing for biological motion recognition includes the *extrastriate body area* (EBA) which to some extent overlaps with the dorsal stream.

The simplified architecture is illustrated in fig. 2.3. In the following the different neurophysiological areas that contribute to biological motion recognition are discussed in more detail.

2.1.5 Dorsal Stream Areas In Biological Motion Recognition

V1, which performs basic processing for both, the ventral and dorsal stream, is connected to the motion specific hMT+ complex. Here, cells in area MT perform what is often termed early motion processing. In the subsequent area MST the receptive fields are larger and related to global movement patterns, such as rotations, and expansions. These are affine movement patterns related to ego-motion [10].

However, these global patterns are not well suited to contribute to biological motion recognition, because articulated human actions are spatially restricted by the underlying body form and do not include globally expanding movements. In fact, a lot of research on the dorsal stream is related to the global aspects of movements, like navigating and detecting moving objects [10, 103]. That is why the dorsal stream is often termed the *where* path in distinction to the ventral *what* path. These global movement

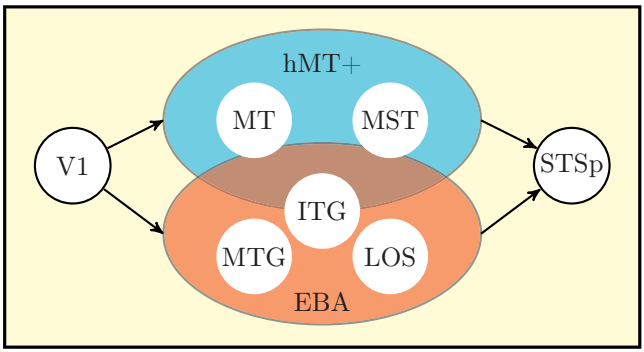

Figure 2.3: Neurophysiological areas involved in the recognition of biological motion. The hMT+ complex is related to explicit motion representations and the EBA complex is sensitive to human body postures and limbs. The two areas are partially overlapping. The illustration does not reflect the exact or relative position of the distinct brain areas. Details on the localization can be found in [110].

patterns are ego-motion induced and related to the depth structure of the observed environment, but not to complex articulated movements.

Contrary to this focus on the global aspects of motion, locally restrained *mid-level motion patterns* are more likely candidates to contribute to the recognition of biological motion.

2.1.6 Mid-Level Motion Patterns

The presence of mid-level motion patterns has been proposed in multiple computational models [13, 38], to bridge the gap between early form invariant motion processing and STSp reaction to full body movements. However, compared to the early motion processing stages there are rather few experiments on this topic.

One publication of particular interest is [83], where it is investigated whether STSp responses are limited to human articulated motion or if they include so called 'creature' motion. Their creatures were artificially created random concatenations of limb-like constructs. In their experiments, Pyles et al. measured the fMRI responses of humans observing creature and human movements. The main result of their study is that STSp responded more strongly to human action than to creature action, which is an indicator

for a STSp specialization to human movements. This was concluded to be in good accordance with its connection to the motor system, because the observing humans are not capable of performing the actions of the creatures. In addition they localized two brain areas in the ventral pathway, area ITS and area pITG, which responded to the creature motion as well as to the human biological motion and which thus play a role in a more general processing of articulated motion, not restricted to human movements. ITS and pITG respond both to creatures as well as humans, who do not share a common global form, but rather common limb forms. In addition they found another area, area IOS, which seemed to react exclusively to novel object movements. In [110], area ITG is found to be a limb-selective part of the EBA, that is overlapping with the hMT+ complex.

ITGs specialization on limbs and its activations by human and 'creature' movement, makeing it a strong candidate to be part of a kind of mid-level motion pattern processing region that relates motion with distinct shapes. However, this hypothesis needs additional confirmation, because it is not clear if the neural activations of the ITG cells are mainly driven by the static limb forms or by the specific articulated limb movements.

2.1.7 Ventral Stream Areas In Biological Motion Recognition

The ventral stream of humans is suggested to perform core object recognition [23]. In macaque monkeys one of the key areas involved is the inferior temporal (IT) complex, whose human counterpart could be the *lateral occipital cortex* (LOC) [23]. Core object recognition is the ability to discriminate between different visual objects in a scale and view-point invariant manner, while being fast and thus mainly feed-forward driven. The experimental results inspired many computational models, *e.g.* [33, 65, 87, 111].

In the case of biological motion recognition, body poses can be seen as objects, whose recognition helps in classifying the displayed human action. Areas related to the recognition of human bodies and limbs are the *fusiform body area* (FBA) and the *extrastriate body area* (EBA) that are both located on the *inferior temporal sulcus* (ITS) [40, 101].

Those areas are also located next to the dorsal hMT+ complex [110] and EBA can be further separated into three limb selective areas, including *infero temporal gyrus* (ITG) that overlaps with the hMT+ complex [110], the *lateral occipital sulcus/middle occipital gyrus* (LOS/MOG) and the *middle temporal gyrus* (MTG).

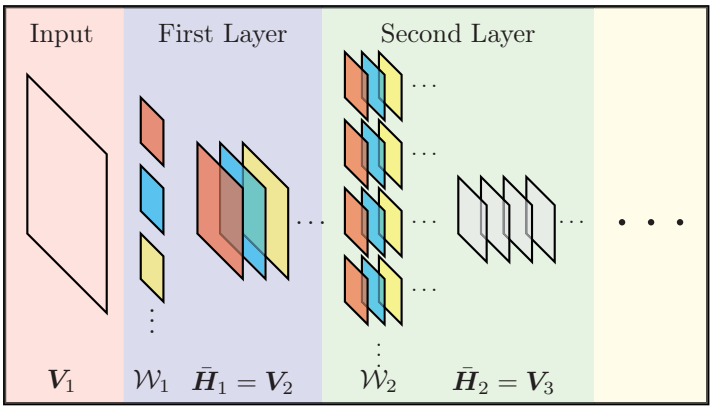

Figure 2.4: The overall architecture of a Feed Forward Neural Network. The input of the first layer $\boldsymbol{V}_1$ is filted by a set of basic patterns $\mathcal{W}_1$ and the resulting, post-processed activations $\bar{\boldsymbol{H}}_1$ are the input $\boldsymbol{V}_2$ for the subsequent layer.

2.1.8 Posterior Superior Temporal Sulcus (STSp)

Area STSp is located a few centimeters anterior to the motion responsive *middle temporal* (MT) and *medial superior temporal* (MST) areas [83] in the dorsal stream, that are also known as the hMT+ complex. It is connected to several other brain areas, *e.g.* the motor system, the hMT+ complex [59] as well as to areas of the ventral vision pathway. Due to its multiple connections it is suggested that the STSp either integrates form and motion or defines form from motion [82]. Pyles et al. [83] show that area STSp reacts to articulated human stimuli but not to artificial 'creature' (random concatenations of limbs) stimuli. They conclude that STSp is specialized to human movement, *i.e.* unlike the hMT+ complex it is not a part of a generic motion processing, but is already *class-specific*. The experiments reported in [40] further suggest that STSp has a *person-centric* representation and is view-point invariant. In addition to being selective to human full body movements and gestures, area STSp also reacts to face movements [34].

2.2 Proposed Computational Model

The biological motion recognition system proposed in this thesis is similar to the biological archetype illustrated in fig. 2.3. It is a hierarchical sytem, whoes layers are learend with novel unsupervised learning algorithms. During recognition the hierarchical architecture is used as a *Feed Forward Neural Network* (FFNN). For visual recognition, FFNN are typically applied for static object recognition, which consist of *one information processing stream*. Following the previous discussion and to cope with the temporal variations, the biological motion recognition system consists of *two streams*,

- one stream for *static* features, in our case based on gradients and
- one stream for *motion* features, in our case based on optical flow.

Before the system is introduced, FFNN along with their invariance properties are shortly discussed.

2.2.1 Feed-Forward Neural Networks

A popular approach towards invariant object recognition are so called *Feed Forward Neural Networks* (FFNN), which also comporise *Convolutional Neural Networks* or *Deep Learning* architectures. These methods are highly motivated by the human visual cortex, especially the primary visual cortex (V1) and the areas V2-V4 as well as the IT-complex [23]. They date back to the early works of Rosenblatt [89], Minsky [73] and the Neocognitron proposed by Fukushima [33]. More recent approaches are *e.g.* proposed in [87, 111] and a discussion is provided in [65]. The main idea is that primary visual object recognition is performed in a purely feed-forward manner throughout a cascade of so called *simple cell/complex cell* layers. The receptive field size as well as the specialization of the layers increases along the hierarchy. As a consequence, the representation in the first layers is spatially bound and not class specific, while the representations in the middle layers show some invariance towards spatial shifts, and the final layer contains so called *grandmother cells* that are object specific and invariant to various 3D transformations. The architecture is depicted in figure 2.4.

Each layer consists of four stages (see figure 2.5):

1. data preprocessing,
2. matching of the input to prelearned patterns (simple cell response),

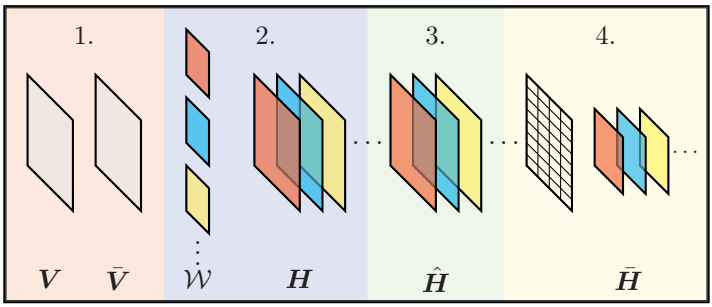

Figure 2.5: The four stages of one simple cell/complex cell layer of a Feed Forward Neural Network. The input $\boldsymbol{V}$ is preprocessed and the resulting $\bar{\boldsymbol{V}}$ is filtered with the simple cell patterns $\mathcal{W}$. The corresponding activations $\boldsymbol{H}$ undergo a post-processing and in a subsequent step post-processed activations $\hat{\boldsymbol{H}}$ are spatially pooled to form the final output $\bar{\boldsymbol{H}}$ of the layer.

3. non-linear post-processing of the resulting activations,
4. and spatial pooling (complex cell response).

The second step makes the responses specific and roughly corresponds to an *AND*-, while the fourth step does a spatial grouping, similar to an *OR*-operator.

FFNN have many variables, such as the number of layers, the pooling size and operation, *e.g.* max, sum, winner-take-most [111], the preprocessing, *e.g.* whitening or normalization, the kind of non-linearity, but most important the simple cell patterns. They are typically learned via back-propagation, which requires a large amount of training data, because the gradients converge to zero during the back-propagation through multiple layers which is known as the *vanishing gradient problem*. A common way to overcome this problem is to use bottom up unsupervised pretraining. Here, beginning with the earliest layer, the patterns are learned via unsupervised learning algorithms. The inputs either consist of randomly extracted small sample patches or, to be in better accordance with the detection, in a convolutional manner on the entire input images. The choice of the learning algorithm and the related properties of the extracted patterns is crucial. Commonly, sparse coding algorithms are applied, but there are other methods that are more extensively discussed in chapter 3.

After the patterns of the first layer are learned, the input is projected onto the second layer and the second layer patches are learned, *a.s.o.* In an optional step, the prelearned layers can be refined by supervised learning, *e.g.* back-propagation. However, as discussed in section 1.1.1, since there are many overlaps between different classes on different levels, supervised learning on early layers can be problematic.

2.2.2 Invariance Properties of Feed-Forward Neural Networks

The main invariance that is implemented into a FFNN is an invariance towards small shifts, due to the pooling layers, *i.e.* complex cells, that loosen the information about the absolute local position of each feature. Furthermore, the simple cell patterns correspond to input stimuli that *e.g.* have a slightly different shape than the ones used for learning, thus providing another form of invariance. But all in all, the descriptive capability of FFNN relies on

- the variations found in the training data and
- the ability to learn and represent all those variations.

As a consequence, even though there is an overall large amount of training data, the learning system should be able to learn from very few examples, since various class-specific representations might only have few occurrences. In addition, the learning architecture must be sufficiently powerful and generative to store all those different representations, while still learning discriminative features.

2.2.3 Related Work

In related work, a popular model that is inspired by the neurobiological observations depicted in fig. 2.3 is proposed by Giese and Poggio [38]. In accordance with the biological archetype it consists of two processing streams: The *ventral* (static) and the *dorsal* (dynamic) stream that both contribute to the recognition of biological motion in a fast, mainly *feed-forward driven architecture*. A drawback of the Giese and Poggio model is that the algorithms used to represent the specific areas are neither non-negative nor sparse or direction selective and no inhibition between neighbouring activations is considered.

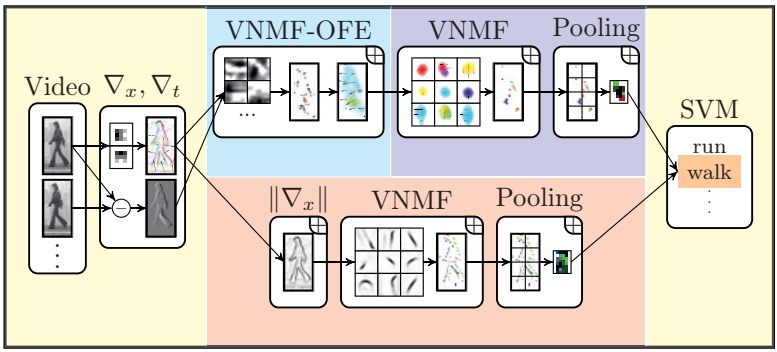

Figure 2.6: Overview of the proposed two stream hierarchical biological motion recognition system. The architecture is inspired by the corresponding neurophysiological model depicted in fig. 2.3. The different colors mark the different processing layers. In the first layer, the spatial (∇_x) and temporal (∇_t) gradients of the incoming video data are calculated. In the motion processing stream, the spatial and temporal gradients are used to estimate the optical flow (VNMF-OFE, cyan) which is thereafter matched onto a set of prelearned optical flow patterns via the VNMF algorithm (blue). In the gradient processing stream (red) the spatial gradient amplitudes ($\|\nabla_x\|$) are calculated and matched onto a set of prelearned gradient patterns with the VNMF algorithm. Both pattern responses are spatially pooled and classified in the final layer that consists of a Support Vector Machine (SVM). Each box with a (+) has a strict non-negative representation.

2.2.4 Proposed Computational Model

The proposed computational model is depicted in fig. 2.6. The two streams of this FFNN, one for the dynamic and one for the static form information, share the first layer, which calculates spatial and temporal image gradients, as well as the last layer, which performs the classification. In between there are two layers for the optical flow stream and one for the gradient stream. Each of these layers consists of multiple steps, *e.g.* a pattern matching or, for the feature extraction, a spatial pooling. Following the discussion in section 1.1.1, the only layer that is learned via supervised learning is the final classification layer. All other layers are learned via unsupervised learning.[1)]

[1)] The first layer is somehow an exception. There is a vast amount of literature which shows that unsupervised learning performed on images produces *Gabor Filters*,

The major difference of the proposed system compared to other FFNN architectures is that each learned layer has its own unsupervised learning algorithm to account for the different objectives encountered on the different layers as discussed in section 2.1.1. Nonetheless, all the unsupervised learning algorithms share the common idea of a *parts-based decomposition* and use a common algorithm based on *non-negativity* and *sparsity* for learning. Thus, the differences between the layers lie in the objective, *i.e.* the energy function, but they share the same underlying coding principle.

The first layer in the motion stream uses the spatiotemporal gradient information of the first layer to calculate an optical flow field. The proposed unsupervised learning algorithm is the VNMF-OFE algorithm. The receptive field sizes in this layer are typically small, since this layer has to be able to represent a large variety of different optical flow fields. *Optical flow estimation* is an ongoing research topic in computer vision that is discussed alongside with related work and the proposed unsupervised learning based solution in chapter 4.

The second layer in the motion stream and accordingly the first layer in the gradient stream are used for *feature extraction*. The unsupervised learning algorithm used in those two stages is the VNMF algorithm. The extracted patterns and their relation to other feature extraction methods, *e.g.* the state-of-the-art HOG/HOF [21] features are discussed in chapter 5.

In the final layer the pooled activations of both streams are classified using a *Support Vector Machine* (SVM) [14][2)]. The classification is performed per frame and the overall result per video is calculated by taking the maximum of the class probabilities provided by the SVM for each frame. The evaluation of the classification performance is done in chapter 6.

i.e. filters that can be used to calculate spatial gradients with different frequencies and orientations (see *e.g.* [52, 54]). For computational reasons, the first layer is modeled by simple, designed gradient filters, instead of learned patterns.

2) SVM's share a lot of properties with the NMF algorithm that is the basis for the VNMF algorithm. The details are discussed in [80, 81].

3 Unsupervised Pattern Learning

The following chapter introduces unsupervised learning algorithms that are applied to learn the patterns for each layer in the proposed hierarchical FFNN. As discussed in the previous chapters, the goal is to learn generative, parts-based patterns. To this end non-negativity, sparsity and novel inhibition functions are included into the learning framework. The algorithms in this thesis build upon *sparse coding* (SC) introduced by Olshausen and Fleet in 1996 [77] and *non-negative matrix factorization* (NMF) introduced three years later by Lee and Seung [66], as well as on its combination, *sparse non-negative matrix factorization* (sNMF) [27, 51, 52] and the transformation invariant extension [28].

The goal of the discussed unsupervised learning algorithms is to find generic patterns, termed *e.g.* principle components, independent components, dictionary elements, basic patterns or basis vectors. These are all synonyms for the same model component which is termed *basis vector* throughout the thesis. The basis vectors are part of a model that is adapted to resemble the given input. The model itself is a linear superposition of the basis vectors. In mathematical terms, the model $\mathcal{R}$ for the given input $\mathcal{V} \in \mathbb{R}^{P \times N}$ is

$$\mathcal{R} = \mathcal{W}\mathcal{H}, \tag{3.1}$$

with the basis vectors $\mathcal{W} \in \mathbb{R}^{P \times J}$ and the activations $\mathcal{H} \in \mathbb{R}^{J \times N}$. The two input dimensions P and N describe the number of pixels and the number of input images, while J is the number of basis vectors. The model parameters $\mathcal{W}$ and $\mathcal{H}$ are learned by minimizing an energy function based on a distance function $d(\,\cdot\,)$, *e.g.* the euclidean distance, between the training data $\mathcal{V}$ and the model $\mathcal{R}$ and additional energy functions $a(\,\cdot\,)$ depending on the model parameters. The resulting optimization problem is

$$\min_{\mathcal{W},\mathcal{H}} \big(d(\mathcal{V}, \mathcal{R}(\mathcal{W},\mathcal{H})) + a(\mathcal{W},\mathcal{H})\big). \tag{3.2}$$

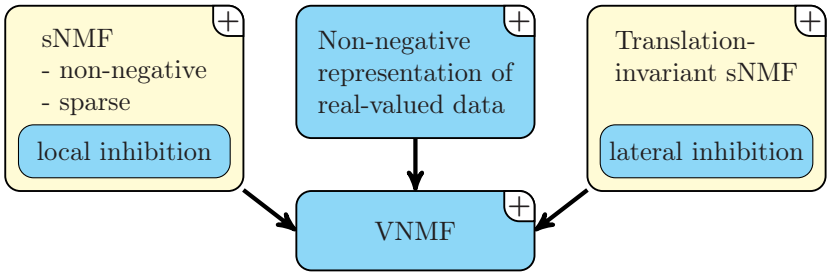

Figure 3.1: Overview of the algorithms presented in this chapter. They are based on sNMF and translation invariant sNMF. The (+) in the corners indicate that all components are non-negative. All novel algorithms are colored in blue. The sNMF algorithm is extended by local inhibition and the translation invariant algorithm is extended by lateral inhibition. The sNMF is further extended in two novel ways to allow for a non-negative representations of real-valued data. The application of the presented algorithm is thus not limited to non-negative inputs. The algorithm which makes use of all the extensions is the novel VNMF algorithm.

The learning algorithms differ in the additional structural restrictions build into the model, in how the model parameters are learned and in the additional energy functions $a(\,\cdot\,)$, but only few methods change the structure of the model itself, *e.g.* translation invariant NMF [28] or trifactor NMF [24].

The chapter is organized as follows: First, related work on pattern learning is discussed, followed by an analysis of constraints that lead to a parts-based representation. Then, sNMF as the basis of the novel algorithms is introduced and it is shown how sNMF can be extended to represent real valued data in a non-negative form. Next, the concept of lateral inhibition that leads to topological sparse representations is discussed along with translation invariant learning. The resulting algorithm is the *vector non-negative matrix factorization* (VNMF). The structure is further illustrated in fig. 3.1.

3.1 Related Work

There exists a vast amount of different unsupervised learning algorithms, whose full review is beyond the scope of this thesis. The review will be limited to two popular methods: *principal component analysis* (PCA) and *independent component analysis* (ICA) as well as extensions of the NMF. Other techniques such as singular value decomposition, which is strongly related to PCA, non-negative PCA, vector quantization, factor analysis and so on, employ other restrictions which are not discussed here. More information concerning the above mentioned methods can be found in [19, 54, 65, 76].

3.1.1 Principal Component Analysis

The oldest and still one of the most popular methods used for the extraction of patterns is *principal component analysis* (PCA). PCA is strongly discussed in the literature, including multiple textbooks [6, 19, 54] and articles [31, 38, 76] and there are numerous ways of how to introduce PCA. Here PCA is discussed as a generative process. PCA learns the basis vectors or *principal components* $\mathcal{W}$ in an iterative fashion, starting with the first principal component $\boldsymbol{W}_1$, then the second *a.s.o.* The basis vectors have to fulfill two conditions: First, the basis vectors have to be orthogonal, thus

$$\boldsymbol{W}_j^\top \boldsymbol{W}_k = 0, \quad \forall j \neq k, \tag{3.3}$$

and second, the basis vectors have to be *normalized*

$$\boldsymbol{W}_j^\top \boldsymbol{W}_j = 1, \quad \forall j \in \{1, ..., J\}. \tag{3.4}$$

The basis vectors are given by the eigenvectors of the input matrix $\mathcal{V}$. The first principal component is defined by the eigenvector with the highest corresponding eigenvalue. In case of a zero mean input, $\boldsymbol{W}_1$ points to the dimension with the highest variance in the data. When applied to images or image patches, PCA produces holistic patterns [66]. PCA can be used for dimensionality reduction or compression, because the amount of image information that can be described by the basis vectors can be calculated from the corresponding eigenvalues [6]. Yet, the basis vectors are holistic and not parts-based and thus not suited for the task at hand, as discussed in [66].

PCA is often applied in conjunction with a preprocessing step known as *whitening* [6, 65], which can be combined with a dimensionality reduction.

Whitening is performed as follows: first, the mean value of each input dimensions is subtracted, which results in a zero-mean distribution of the input data. Second, the data points are projected onto the orthogonal basis provided by the principal components calculated with PCA. Using only a subset of all principal components reduces the dimension accordingly. In the last step the data is divided by the variance. The result is that all dimensions have a similar distribution, which might be beneficial for a qualitatively equal influence during the distance measurement in the penalty function. The underlying assumption is that each input dimension has similar value to the input data.

It is unclear whether this assumption is valid, *e.g.* if an input dimension has only noise values near zero, whitening amplifies a variability in this dimension that beforehand was not in the data. That is why throughout the thesis no whitening is performed on the data.

3.1.2 Independent Component Analysis

Another widespread pattern learning algorithm is *independent component analysis* (ICA) and its extension, the *independent subspace analysis* (ISA) [54, 64]. Here the additional condition is to find a set of basis vectors by maximizing the independence of the activations $\boldsymbol{H}_i$ for each input $\boldsymbol{V}_i$. *Independence* is defined in terms of stochastic variables. Let h_j and h_k be stochastic variables that correspond to the activations $\boldsymbol{H}_j$ and $\boldsymbol{H}_k$ used in our deterministic interpretation. h_j and h_k are independent if the joint probability density function can be split up into a multiplication of the individual probability density functions: $p(h_j, h_k) = p(h_j)p(h_k)$. *I.e.* the presence of the basis vector $\boldsymbol{W}_j$ in an input image is not related to the presence of any other basis vector $\boldsymbol{W}_k, \forall k \neq j$. A detailed analysis is given in [54].

3.1.3 Extensions of NMF

Since the introduction of the original NMF [66, 67] by Lee and Seung, multiple extensions of the NMF haven been proposed. The overall trends have recently been summarized in a textbook [19].

One of the major benefits of the NMF learning algorithm is its guaranteed convergence, which is discussed in [26]. In [63], the uniqueness of the basis vectors, learned with NMF is analyzed.

A popular extensions of the NMF is to add orthogonality constraints on the basis vectors, like in [17] or the LNMF [69]. They report that the

orthogonal basis vectors are more parts-based and achieve better results for face recognition than the original NMF. A similar result is reported when the NMF is combined with sparsity constraints on the activations as proposed in [27, 51, 52]. Alternatively to sparsity and orthogonality constraints, a determinant criteria is proposed in [91].

Structural extensions are proposed in [24] and [84]. The semi-NMF [24] relaxes the non-negativity constraints on the input and the activations, which broadens the possible range of applications for the NMF approach. The authors further show the relations of the semi-NMF to k-means clustering. In [84] a hierarchical NMF is proposed. The hierarchical approach allows for decompositions of increasing complexity, because the activations are themselves represented by an additional NMF layer.

The NMF learns the basis vectors given a finite number of inputs. The algorithm proposed by Lee and Seung is thus a *batch learning* algorithm. To extend the NMF to be able to adapt to novel data *incremental learning* [85] and *online-learning* [72] algorithms are proposed.

Besides the relations of the NMF to k-mean clustering as discussed in [24], there are close relations between NMF and SVM as discussed in [80, 81]. The relations can be used to solve the NMF with the algorithms used to solve SVM and vice versa.

3.2 Properties of Parts-based Representations

The goal of the pattern learning algorithms is to get a parts-based, generative representation of the input. The question at hand is: What are *reasonable constraints* for the learning algorithm to achieve a *parts-based* decomposition? This is already motivated in the introduction and illustrated in fig. 1.1 and fig. 1.2, *i.e.* non-negativity, sparsity and inhibition are desired constraints for the decomposition.

From the discussion of the related work another question arises concerning the properties of the desired constraints: the constraints can either be *strict* or *weak*. The orthogonality constraint on the basis vectors in the PCA is an example for a strict condition, while the independence in the ICA is only maximized, but in general not reached, thus it is a weak condition, that is desired, rather than strictly enforced. In summary there are two questions:

1. *What* constraints are applied on $\mathcal{W}$ and $\mathcal{H}$ and
2. *How* are the constraints applied, either

a) *strict* constraints, that are enforced or

b) *weak* constraints, that are desired, but not necessarily reached.

Weak constraints can be mathematically formulated by energy functions that are included into the optimization process. Thuse, following the idea outlined in (3.2), the overal energy functions is a linear superposition of individual energy functions depending on a subset or all model parameters, *i.e.* the activations $\mathcal{H}$ and the basis vectors $\mathcal{W}$.

$$E = E_1(\mathcal{W}, \mathcal{H}) + E_2(\mathcal{W}, \mathcal{H}) + E_3(\mathcal{W}, \mathcal{H}) + ... \tag{3.5}$$

Each energy function E_i representing a weak constraint. The weak constraints are useful if there exist multiple, probably conflicting constraints, like *e.g.* in the case of ICA, being generative and having independent activations. Furthermore, in case of an iterative learning procedure, weak constraints allow the model to be flexible during the learning process. In contrast, the strict constraints are fixed and thus cannot be optimized, they rather restrict the design space of the optimization.

3.2.1 Basic Constraints

The strongest constraint is in fact the *linear model* with the *fixed number of basis vectors* J. In principle, a linear model can represent any input if J is greater or equal than any of the two input dimensions P and N. In case of a gray valued image with P pixels, the *trivial model* consist of $J = P$ basis vectors, each just representing a different single pixel.[1)] If $J = N$ each basis vector can simply represent one input, which is another trivial solution. However, since the input data should be compressed by the pattern learning algorithm, the model should fulfill the following condition: To achieve a dimensionality reduction, the number of basis vectors J must be smaller than both input dimensions P and N, *i.e.* ,

$$J << P, \tag{3.6}$$

$$J << N. \tag{3.7}$$

This limitation on the amount of available basis vectors restricts the model and makes it less arbitrary than in the case of the trivial solutions.

The next constraint is that the model has to be *generative*. It is a soft constraint, because the full description of the input can in general only

[1)] This will be termed the *trivial solution*, which will play an important role when the translation invariant model is discussed.

be achieved by the trivial solution discussed above. It is enforced by the reconstruction energy

$$E_{\text{rec}} = d(\mathcal{V}, \mathcal{R}), \tag{3.8}$$

which can be considered as the *driving force* of the optimization, because it is directly related to the input. However, being generative is not the only goal of our optimization, to get an invariant representation other, possibly conflicting constraints are required. In fact, all other addition constraints used in this thesis increase the restrictiveness of the model and are thus counterproductive concerning the reconstruction quality.

3.2.2 Non-negativity

The most important strict constraint that is enforced in all algorithms developed in this thesis is the strict *non-negativity* of all components. The concept was introduced in 1994 by Paatero and Tapper [78] as *positive matrix factorization* and became increasingly popular when Lee and Seung introduced their multiplicative learning rules for their new termed *non-negative matrix factorization* (NMF) in 1999 [66]. When applied to face images the NMF learns facial parts which lead to an increased interest in the non-negativity constraints. The assumption is that non-negativity favors parts-based representations.

This assumption, that was already discussed in the introduction, can to some extent be explained by the following gedankenexperiment: non-negativity in the activations and basis vectors removes the capability of the model to subtract elements. If an element is added to the model, there is no possibility to remove it by *e.g.* subtracting a basis vector that provides the part that needs to be removed. Thus, if a basis vector is added to the reconstruction model it must be useful in terms of the reconstruction quality. This can be achieved by restricting the basis vectors to define only *prototypical parts* of the input. Hence, the parts-based properties emerge from the models *inability to subtract elements*.

The non-negative representation further provides a parameter free and fast converging learning algorithm. Learning is achieved by minimizing the energy function with an adapted, parameter free form of iterative gradient descent of the model parameters $\mathcal{W}$ and $\mathcal{H}$. To apply the NMF learning rules the gradients need to be separated into their positive and negative components. In case of the non-negative representation this can be achieved in a straightforward manner as it will be discussed in the next sections.

Beside the computational benefits, a non-negative representation is more biologically plausible than a real valued representation including negative values. A neuron can either be active (spiking) or inactive, but there is no such thing as a negative neural activation.

Non-negativity might seem like a hard restriction because there exist a lot of data with positive and negative values. However, the restriction only applies to the inner representation of the model. In section 3.4 it is shown how to achieve a non-negative representation for any kind of real-valued input data.

3.2.3 Sparsity

The concept of *sparsity* was introduced into pattern learning in 1996 by Olshausen and Fleet [77, 90]. When given natural images as input, their *sparse coding* (SC) algorithm is capable of learning Gabor filters, as they are found in experiments in the early human visual cortex V1. Sparsity constraints have since then been of major interest in pattern learning. They can either be applied to the basis vectors or, what is more common and used in this thesis, on the activations.

Sparsity on the activations is a weak constraint, because it cannot be strictly enforced. The most sparse representation is a model without only one activations, which which may oppose the generative idea of representing the input. Activations are necessary to build a model, so the generative constraint and the sparsity constraint are in competition and have to be balanced by the relative weights of the corresponding energy functions.

The main benefit gained from having sparse activations can be formulated as follows: Enforced sparsity in the activations favors models with *as few active basis vectors as necessary*. As a consequence basis vectors which are spatially extended are favored over those with only few elements. As a consequence, if the sparsity constraint is too strong, the decomposition will be holistic and not a parts-based, because holistic representations require only few activations.

In terms of pattern learning as an optimization problem, the sparsity constraints favor those models, thus minima, that have only few active basis vectors and penalizes those minima with multiple active basis vectors, independent of the reconstruction energy. An example for a sparse and a non-sparse decomposition are given in fig. 3.2.

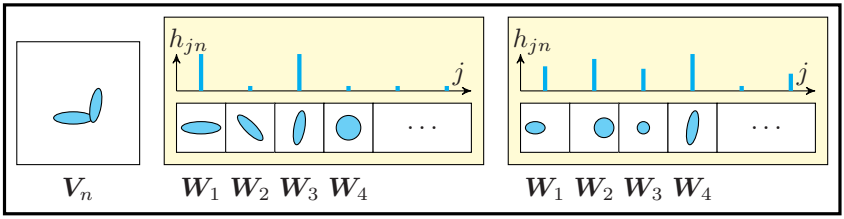

Figure 3.2: Two artifical decompositions of the input $\boldsymbol{V}_n$. From left to right: a sparse decomposition, *i.e.* only few basis vectors are activated to reconstruct the given input. Then a non-sparse decomposition, *i.e.* multiple basis vectors $\boldsymbol{W}_j$ are activated to reconstruct the given input. An enforced sparsity favors the sparse decomposition with fewer activations.

3.2.4 Local and Lateral Inhibition

When applied in a FFNN, basis vectors with overlapping receptive fields are simultaneously activated, which leads to blurry activation patterns that counteracts the idea of sparse activations. This effect is termed the *superposition problem*. There already exist effective methods on how to deal with the superposition problem via non-linear pooling in the complex cells of FFNN, that is now shortly discussed.

In the detection phase of a FFNN the simple cells are followed by the complex cells, a non-linear projection of the simple cell responses with a local and a lateral component. The local competition is between different activities at the same position, for example a norm-, a maximum- or a winner-takes-most [111] operator. The lateral competition is achieved via a max-pooling step. Neighboring activities are projected onto a single activity, which leads to activity images with a reduced resolution. Besides the lateral competition, the pooling has the additional effect of increasing the receptive field size and of introducing an invariance to small shifts in the input space. Thus, the goal of the non-linear post-processing on the activities is to arrive at a representation with sparser activities as well as an increasing translation invariance and larger receptive fields throughout the hierarchy.

A major drawback of the approach is that this non-linear post-processing is not consistent with the learning process, since it is only applied during the detection phase. *I.e.* , the learned basis vectors do not correspond to

a topologically sparse decomposition and may not be best suited for the complex cell type sparsifications.

While there are sparse coding algorithms that incorporate local competition into the learning procedure [69, 90], the lateral competition is not addressed. One reason for this drawback is that the basis vectors are learned on randomly sampled image patches. Due to the sampling process the neighboring dependencies get lost and cannot be addressed during the learning process. To overcome this problem and to achieve a *topological sparse* representation, a *translation invariant learning* procedure with an additional *local and lateral competition* penalty function is proposed later in this chapter.

3.2.5 Resulting Energy Function and Notations

Following the idea that the weak constraints are represented by corresponding energy functions, the algorithms presented in this thesis consist of either all or a subset of terms contained in the energy function

$$E = E_{\text{rec}} + \lambda_{\text{h}} E_{\text{h}} + \lambda_p E_p, \tag{3.9}$$

with E_{rec} as the reconstruction energy term, E_{h} the sparsity term and E_p the inhibition term that further enforces a parts-basedness representation.[2)] The corresponding weighting parameters for the different energy terms are λ_{h} and λ_p. The influence of these parameters will be discussed during the experiments.

The notation applied throughout the algorithmic description in this thesis is now introduced.

$$\mathcal{H} \in \mathbb{R}_+^{J \times N}, \tag{3.10}$$

with $\mathbb{R}_+ = [0, \infty)$, describes the entire matrix of non-negative activations. A subset of this activation matrix, *e.g.* the activations corresponding to the input vector $\boldsymbol{V}_n$ is $\boldsymbol{H}_n$. A single entry of the activation matrix, *e.g.* the activation corresponding to input $\boldsymbol{V}_n$ and the j-th basis vector is h_{jn}. Lower case letters always correspond to scalar values, bold upper case letters represent vectors and matrix are represented by cursive upper case letters.[3)]

[2)] In case of multidimensional input data, such as vector fields representing the optical flow, there is an additional energy function, dealing with effects related to opposing directions. It will be discussed in section 3.4.

[3)] This notation is only valid in the non-translation-invariant case. The notation for the translation-invariant NMF is given in section 3.5.

The reconstruction of the entire input $\mathcal{V}$ is given by

$$\mathcal{R} = \mathcal{W}\mathcal{H} \tag{3.11}$$

and for a single $\boldsymbol{V}_n$ the reconstruction is

$$\boldsymbol{R}_n = \mathcal{W}\boldsymbol{H}_n, \tag{3.12}$$

with the partial reconstruction due to a single basis vector being

$$\boldsymbol{R}_{jn} = \boldsymbol{W}_j h_{jn}. \tag{3.13}$$

The element wise notation for the reconstruction of a single input and pixel is

$$r_{pn} = \sum_j w_{pj} h_{jn}. \tag{3.14}$$

All elements of the input and the model are strictly non-negative, *i.e.*

$$v_{pn} \geq 0, \quad \forall p \in \{1, ..., P\}, \forall n \in \{1, ..., N\}, \tag{3.15}$$
$$r_{pn} \geq 0, \quad \forall p \in \{1, ..., P\}, \forall n \in \{1, ..., N\}, \tag{3.16}$$
$$w_{pj} \geq 0, \quad \forall p \in \{1, ..., P\}, \forall j \in \{1, ..., J\}, \tag{3.17}$$
$$h_{jn} \geq 0, \quad \forall j \in \{1, ..., J\}, \forall n \in \{1, ..., N\}. \tag{3.18}$$

3.3 Sparse Non-negative Matrix Factorization

The basis algorithm for all methods proposed in this thesis is sNMF. It combines the ideas of strict non-negativity and sparsity in the activations. Learning is seen as a optimization problem, which leads to two major questions:

1. What is the optimization criterion?
2. How is the optimization problem solved?

The optimization criterion is the minimization of a given energy function $E(\mathcal{W}, \mathcal{H})$, that depends on the model parameters, *i.e.* the basis vectors $\mathcal{W}$ and the activations $\mathcal{H}$. The energy function consists of varying parts, depending on the constraints, such as sparsity and topological sparsity. Due to the non-negativity of all involved components, the energy function can be minimized by multiplicative gradient descent rules as proposed by

Lee and Seung [66]. In general any optimization technique could be used to solve the learning problem, however, the multiplicative update rules are fast and guarantee convergence [26]. In the following the different components of the energy function for the sNMF algorithm is introduced.

3.3.1 Sparse Activations

There exist multiple methods to favor sparse activations in the learning process. One straightforward method is to penalize the activations with an additional energy function $E_\mathrm{h} = f(\boldsymbol{H})$, *e.g.* the l_1-norm

$$E_\mathrm{h} = \lambda_\mathrm{h} \sum_{n,j} h_{jn}, \tag{3.19}$$

with the positive gradient component

$$(\nabla_{h_{jn}} E_\mathrm{h})^+ = \lambda_\mathrm{h}. \tag{3.20}$$

Other energy functions, including quadratic functions are *e.g.* proposed in [77], but due to its simplicity and effectiveness eq. (3.19) is used in this thesis.

3.3.2 Normalized Basis Vectors

Eq. (3.19) can lead to an undesired scaling effect between the basis vectors and the activations, that circumvents any sparsity enforcing influence. If the activations are reduced by $\frac{1}{\alpha}$, the influence of the scaling on the reconstruction can be averted by scaling the basis vectors up by the factor α. As a consequence, the sparsity energy can be minimized without changing the reconstruction energy. To avoid this undesired scaling effect, the basis vectors are normalized using the euclidean norm

$$\bar{\boldsymbol{W}}_j = \frac{\boldsymbol{W}_j}{\sqrt{\sum_p w_{pj}^2}}, \quad \forall j \in \{1, ..., J\}. \tag{3.21}$$

For normalized basis vectors the following condition is valid, *i.e.*

$$\bar{\boldsymbol{W}}_j^\top \bar{\boldsymbol{W}}_j = 1, \quad \forall \{1, ..., J\}. \tag{3.22}$$

In addition, the normalization influences the gradient calculated for the update procedure. $\bar{\boldsymbol{W}}(\mathcal{W})$ is a function of $\mathcal{W}$, thus, the inner derivation of

this function has to be considered when the basis vectors $\mathcal{W}$ are updated. The resulting positive and negative gradient components, depending on the gradient of the energy functions for the basis vectors is given by

$$(\nabla_{\mathcal{W}} E)^+ = (\nabla_{\bar{\mathcal{W}}} E)^+ + \bar{\mathcal{W}}\bar{\mathcal{W}}^\top (\nabla_{\bar{\mathcal{W}}} E)^-, \tag{3.23}$$

$$(\nabla_{\mathcal{W}} E)^- = (\nabla_{\bar{\mathcal{W}}} E)^- + \bar{\mathcal{W}}\bar{\mathcal{W}}^\top (\nabla_{\bar{\mathcal{W}}} E)^+. \tag{3.24}$$

The indices $(\,\cdot\,)^+$ and $(\,\cdot\,)^-$ indicate the positive and negative part of the gradients. The normalization influences the learning procedure in two ways: the basis vectors have to be normalized after each update using eq. (3.21) and the gradients of the energy functions towards the normalized basis vectors have to be modified according to the equations (3.23) and (3.24) to account for the inner derivation.

3.3.3 Sparse Basis Vectors

Sparsity constraints can be added upon the basis vectors as well, *e.g.* by using the energy function

$$E_{\mathrm{w}} = \lambda_{\mathrm{w}} \sum_{p,j} \bar{w}_{pj}, \tag{3.25}$$

with the gradient

$$(\nabla_{\bar{w}_{pj}} E_{\mathrm{w}})^+ = \lambda_{\mathrm{w}}. \tag{3.26}$$

However, for the proposed algorithms this does not provide benefits for the learning process and is therefore discarded.

3.3.4 Reconstruction Energy

The reconstruction energy function is the driving force of the learning process. In [19] there is a comparison of different distance measures, including several robust functions. A simple and computationally feasible version is the euclidean distance as a measurement for the reconstruction energy

$$E_{\mathrm{rec}} = \frac{1}{2}\|\mathcal{V} - \mathcal{R}\|_F^2 = \frac{1}{2}\|\mathcal{V} - \bar{\mathcal{W}}\mathcal{H}\|_F^2 \tag{3.27}$$

$$= \frac{1}{2}\sum_{n,p}(v_{pn} - r_{pn})^2 = \frac{1}{2}\sum_{n,p}(v_{pn} - \sum_j \bar{w}_{pj} h_{jn})^2. \tag{3.28}$$

The corresponding gradients for the activations are

$$(\nabla_{\mathcal{H}} E_{\text{rec}})^+ = \bar{\mathcal{W}}^\top \mathcal{R}, \tag{3.29}$$
$$(\nabla_{\mathcal{H}} E_{\text{rec}})^- = \bar{\mathcal{W}}^\top \mathcal{V}, \tag{3.30}$$

and for the basis vectors

$$(\nabla_{\bar{\mathcal{W}}} E_{\text{rec}})^+ = \mathcal{R}\mathcal{H}^\top, \tag{3.31}$$
$$(\nabla_{\bar{\mathcal{W}}} E_{\text{rec}})^- = \mathcal{V}\mathcal{H}^\top. \tag{3.32}$$

3.3.5 sNMF Learning Algorithm

Combining the reconstruction energy (3.28) using normalized basis vectors with the sparsity energy function (3.19) results in the overall energy function for the sNMF algorithm

$$E_{\text{sNMF}} = \frac{1}{2}\|\mathcal{V} - \mathcal{R}\|_F^2 + \lambda_{\text{h}} \sum_{n,j} h_{jn}. \tag{3.33}$$

The gradients for the basis vectors are

$$(\nabla_{\mathcal{W}} E_{\text{sNMF}})^+ = \mathcal{R}\mathcal{H}^\top + \bar{\mathcal{W}}\bar{\mathcal{W}}^\top \mathcal{V}\mathcal{H}^\top, \tag{3.34}$$
$$(\nabla_{\mathcal{W}} E_{\text{sNMF}})^- = \mathcal{V}\mathcal{H}^\top + \bar{\mathcal{W}}\bar{\mathcal{W}}^\top \mathcal{R}\mathcal{H}^\top. \tag{3.35}$$

The gradients for the activations are

$$(\nabla_{\mathcal{H}} E_{\text{sNMF}})^+ = \bar{\mathcal{W}}^\top \mathcal{R} + \lambda_{\text{h}}, \tag{3.36}$$
$$(\nabla_{\mathcal{H}} E_{\text{sNMF}})^- = \bar{\mathcal{W}}^\top \mathcal{V}. \tag{3.37}$$

The learning algorithm consists of two parts, the initialization and the updates, which are performed in an iterative process.

The algorithm is:

- Preprocessing
  - Normalize $\mathcal{V} = \frac{\mathcal{V}}{\max(\mathcal{V})}$,
  - initialize $\mathcal{H}$ and $\mathcal{W}$ randomly.
- Loop for i iterations
  1. Calculate $\mathcal{R} = \bar{\mathcal{W}}\mathcal{H}$,

2. update $\mathcal{H} \rightarrow \mathcal{H} \circ \frac{(\nabla_{\mathcal{H}} E_{\text{sNMF}})^-}{(\nabla_{\mathcal{H}} E_{\text{sNMF}})^+}$,
3. calculate $\mathcal{R} = \bar{\mathcal{W}}\mathcal{H}$,
4. update $\mathcal{W} \rightarrow \mathcal{W} \circ \frac{(\nabla_{\mathcal{W}} E_{\text{sNMF}})^-}{(\nabla_{\mathcal{W}} E_{\text{sNMF}})^+}$,
5. normalize $\bar{\boldsymbol{W}}_j = \frac{\boldsymbol{W}_j}{\sqrt{\sum_p w_{pj}^2}}, \quad \forall j \in \{1, ..., J\}$.

One of the benefits of this algorithm is, that it is independent of step size parameters that are typically required in gradient descent methods. The multiplicative update rules in step 2. and 4. can to some extend be considered as finding the optimal step size parameter. There exist different alternatives of how to initialize $\mathcal{W}$ and $\mathcal{H}$ as well as different stopping criteria for the learning loop, both discussed in detail in [19]. Throughout this thesis $\mathcal{W}$ and $\mathcal{H}$ are initialized randomly and the learning is stopped after $i = 300$ iterations, after which in all encountered cases the algorithm lied close to the minima.

3.3.6 Orthogonality and Enforced Parts-Basedness

Another typical extension in form of a weak constraint of the NMF framework is to favor orthogonal basis vectors [17, 69]. The underlying idea is to achieve a more parts-based decomposition. While orthogonality between the basis vectors is appealing (because it penalizes overlaps) it also has two downsides. First, the additional energy functions that enforce orthogonality on $\mathcal{W}$ depend on $\mathcal{W}$ only and do not scale according to the reconstruction energy. Thus, for different input data, novel weighting parameters for the different energy functions have to be found. This makes the approach rather difficult to apply. The second downside is, that potential redundancies in the basis vectors, *e.g.* two different types of noses in the case of face images, will be penalized by the energy function, even if they have no overlapping occurrence in the data.

The novel approach proposed in this thesis is not to enforce orthogonality between the basis vectors, but between partial reconstructions $\boldsymbol{R}_{jn}$ and $\boldsymbol{R}_{kn}$ (corresponding to the contributions of the basis vectors $\boldsymbol{W}_j$ and $\boldsymbol{W}_k$ to the reconstruction $\boldsymbol{R}_n$). Thus, only active overlapping basis vectors are penalized. This form of orthogonality is related to the input driven competition of activations with overlapping receptive fields, *i.e.* basis vectors. The competition can either lead to one winning activation that suppresses all activations that correspond to all overlapping basis vectors or in a change of the basis vectors so that they are no longer overlapping.

The proposed energy function is

$$E_{\text{part}} = \frac{1}{2}\lambda_{\text{part}} \sum_{n,j} \left(\boldsymbol{R}_{jn}^{\top} \sum_{k \neq j} \boldsymbol{R}_{kn} \right) \tag{3.38}$$

$$= \frac{1}{2}\lambda_{\text{part}} \sum_{n,j} \left(\boldsymbol{R}_{jn}^{\top} \boldsymbol{R}_{n} - \boldsymbol{R}_{jn}^{\top} \boldsymbol{R}_{jn} \right) \tag{3.39}$$

$$= \frac{1}{2}\lambda_{\text{part}} \sum_{n} \boldsymbol{R}_{n}^{\top} \boldsymbol{R}_{n} - \frac{1}{2}\lambda_{\text{part}} \sum_{n,j} \boldsymbol{R}_{jn}^{\top} \boldsymbol{R}_{jn}, \tag{3.40}$$

with the two energy components

$$E_{\text{p1}} = \frac{1}{2}\lambda_{\text{part}} \sum_{n} \boldsymbol{R}_{n}^{\top} \boldsymbol{R}_{n}, \tag{3.41}$$

$$E_{\text{p2}} = \frac{1}{2}\lambda_{\text{part}} \sum_{n,j} \boldsymbol{R}_{jn}^{\top} \boldsymbol{R}_{jn} \tag{3.42}$$

$$= \frac{1}{2}\lambda_{\text{part}} \sum_{n,j,p} \bar{w}_{pj}^{2} h_{jn}^{2}. \tag{3.43}$$

The partial gradients for the activations are

$$(\nabla_{h_{jn}} E_{\text{p1}}) = \lambda_{\text{part}} \bar{\boldsymbol{W}}_{j}^{\top} \boldsymbol{R}_{n}, \tag{3.44}$$

$$(\nabla_{h_{jn}} E_{\text{p2}}) = \lambda_{\text{part}} h_{jn} \bar{\boldsymbol{W}}_{j}^{\top} \bar{\boldsymbol{W}}_{j}, \tag{3.45}$$

with eq. (3.22),

$$(\nabla_{\mathcal{H}} E_{\text{p1}}) = \lambda_{\text{part}} \bar{\mathcal{W}}^{\top} \mathcal{R}, \tag{3.46}$$

$$(\nabla_{\mathcal{H}} E_{\text{p2}}) = \lambda_{\text{part}} \mathcal{H}. \tag{3.47}$$

Combined with $(\nabla_{\mathcal{H}} E_{\text{p1}}) - (\nabla_{\mathcal{H}} E_{\text{p2}}) \geq 0$ the resulting positive and negative gradient components are

$$(\nabla_{\mathcal{H}} E_{\text{part}})^{+} = \lambda_{\text{part}} (\bar{\mathcal{W}}^{\top} \mathcal{R} - \mathcal{H}), \tag{3.48}$$

$$(\nabla_{\mathcal{H}} E_{\text{part}})^{-} = 0. \tag{3.49}$$

For the basis vectors the partial gradients are

$$(\nabla_{\bar{\boldsymbol{W}}_j} E_{\mathrm{p1}}) = \lambda_{\mathrm{part}} \sum_n h_{jn} \boldsymbol{R}_n \tag{3.50}$$

$$= \lambda_{\mathrm{part}} \boldsymbol{R} \boldsymbol{H}_j^\top, \tag{3.51}$$

$$(\nabla_{\bar{\boldsymbol{W}}_j} E_{\mathrm{p2}}) = \lambda_{\mathrm{part}} \bar{\boldsymbol{W}}_j \sum_n h_{jn}^2 \tag{3.52}$$

$$= \lambda_{\mathrm{part}} \bar{\boldsymbol{W}}_j \boldsymbol{H}_j^\top \boldsymbol{H}_j, \tag{3.53}$$

and with $(\nabla_{\bar{\boldsymbol{W}}_j} E_{\mathrm{p1}}) - (\nabla_{\bar{\boldsymbol{W}}_j} E_{\mathrm{p2}}) \geq 0$ the positive and negative gradient components are set to

$$(\nabla_{\bar{\boldsymbol{W}}_j} E_{\mathrm{part}})^+ = \lambda_{\mathrm{part}} (\boldsymbol{R} \boldsymbol{H}_j^\top - \bar{\boldsymbol{W}}_j \boldsymbol{H}_j^\top \boldsymbol{H}_j), \tag{3.54}$$

$$(\nabla_{\bar{\boldsymbol{W}}_j} E_{\mathrm{part}})^- = 0. \tag{3.55}$$

3.4 Non-negative Representations of Real-valued Data

In their publication [66], Lee and Seung applied the NMF to gray value images, which are naturally non-negative. However, other data types, *e.g.* vector fields that are used to describe optical flow fields are two dimensional and contain negative alongside with positive values. Still, a parts-based decomposition as achieved with the sNMF algorithm is desirable for this kind of data.

In this section the sNMF is extended to deal with vector fields or multidimensional real valued data in general. To this end, two approaches of how to deal with multidimensional input data are introduced and it is shown how they can be applied to real valued data.

3.4.1 Multidimensional Input

The input matrix $\mathcal{V}$ is extended by an additional feature dimension, so that

$$\mathcal{V} \in \mathbb{R}_+^{P \times N \times F}. \tag{3.56}$$

The element wise formulation is

$$v_{pnf} \geq 0, \quad \forall p \in \{1, ..., P\}, \forall n \in \{1, ..., N\}, \forall f \in [1, ..., F]. \tag{3.57}$$

The additional dimension f has to be represented by the model, *i.e.* , the reconstruction $\mathcal{R} \in \mathbb{R}_+^{P \times N \times F}$. This additional dimension can either be encoded in the basis vectors (later used for the feature learning) or the activations (used during the optical flow estimation). As discussed in the beginning of this chapter, the basis vector as well as the activations each share one of the dimension of the input, which is p for the basis vectors and n for the activations. Adding the new dimension f to the basis vectors is identical to extending the basis vector specific dimension p of the input to $\hat{p} = p \cdot f$. Alternatively, the activation specific dimension n of the input is extended to $\hat{n} = n \cdot f$. Yet, the explicit notation of the new feature dimension is preferable, because it is required when addressing the relations between the features in the case of translation invariant learning in section 3.5. The two scenarios, multidimensional basis vectors or multidimensional activations are introduced in the following.

3.4.2 Multidimensional Basis Vectors

In the first case, the basis vectors dimension is extended $\mathcal{W} \in \mathbb{R}_+^{P \times J \times F}$. Here all feature dimensions are explicitly represented by the basis vector and share a *common activation*. The reconstruction is

$$r_{pnf} = \sum_j \bar{w}_{pjf} h_{jn}, \tag{3.58}$$

or written in vector form

$$\boldsymbol{r}_{pn} = \sum_j \bar{\boldsymbol{w}}_{pj} h_{jn}. \tag{3.59}$$

The reconstruction energy becomes

$$E_{\text{rec}} = \frac{1}{2} \sum_{p,n,f} (v_{pnf} - \sum_j \bar{w}_{pjf} h_{jn})^2, \tag{3.60}$$

with the gradient components for the activations

$$(\nabla_{\mathcal{H}} E_{\text{rec}})^+ = \sum_f \bar{\mathcal{W}}_f^\top \mathcal{R}_f, \tag{3.61}$$

$$(\nabla_{\mathcal{H}} E_{\text{rec}})^- = \sum_f \bar{\mathcal{W}}_f^\top \mathcal{V}_f, \tag{3.62}$$

and the basis vectors

$$(\nabla_{\bar{\mathcal{W}}_f} E_{\text{rec}})^+ = \mathcal{R}_f \mathcal{H}^\top, \tag{3.63}$$

$$(\nabla_{\bar{\mathcal{W}}_f} E_{\text{rec}})^- = \mathcal{V}_f \mathcal{H}^\top. \tag{3.64}$$

The basis vectors have to be normalized over all features, *i.e.*

$$\bar{\boldsymbol{W}}_{jf} = \frac{\boldsymbol{W}_{jf}}{\sqrt{\sum_{p,f} w_{pjf}^2}}, \quad \forall j \in \{1, ..., J\}. \tag{3.65}$$

3.4.3 Multidimensional Activations

In the second case, the dimension of the activations is extended $\mathcal{H} \in \mathbb{R}_+^{J \times N \times F}$ and the features configuration can change depending on the activation. The feature configuration is constant throughout the area defined by the *common basis vector* and the reconstruction is

$$r_{pnf} = \sum_j \bar{w}_{pj} h_{jnf}. \tag{3.66}$$

or written in vector form

$$\boldsymbol{r}_{pn} = \sum_j \bar{w}_{pj} \boldsymbol{h}_{jn}. \tag{3.67}$$

The reconstruction energy is

$$E_{\text{rec}} = \frac{1}{2} \sum_{p,n,f} (v_{pnf} - \sum_j \bar{w}_{pj} h_{jnf})^2, \tag{3.68}$$

with the partial derivations for the activities

$$(\nabla_{\mathcal{H}_f} E_{\text{rec}})^+ = \bar{\mathcal{W}}^\top \mathcal{R}_f, \tag{3.69}$$

$$(\nabla_{\mathcal{H}_f} E_{\text{rec}})^- = \bar{\mathcal{W}}^\top \mathcal{V}_f, \tag{3.70}$$

and the basis vectors

$$(\nabla_{\bar{\mathcal{W}}} E_{\text{rec}})^+ = \sum_f \mathcal{R}_f \mathcal{H}_f^\top, \tag{3.71}$$

$$(\nabla_{\bar{\mathcal{W}}} E_{\text{rec}})^- = \sum_f \mathcal{V}_f \mathcal{H}_f^\top. \tag{3.72}$$

3.4.4 Sparse Activation Amplitudes

Since for multidimensional inputs the activation $\boldsymbol{h}_{jn}$ for each image and basis vector is a vector and no longer a scalar value, the sparsity function has to be adapted, otherwise each vector element h_{jnf} would be treated independently. Instead of penalizing the scalar vector elements h_{jnf}, the vector amplitude $\|\boldsymbol{h}_{jn}\|_2$ can be addressed by a sparsity function. This new sparsity function is

$$E_{\mathrm{h}} = \lambda_{\mathrm{h}} \sum_{n,j} \|\boldsymbol{h}_{jn}\|_2 = \lambda_{\mathrm{h}} \sum_{n,j} \sqrt{\sum_f h_{jnf}^2}, \tag{3.73}$$

with the corresponding gradients

$$(\nabla_{h_{jnf}} E_{\mathrm{h}})^+ = \lambda_{\mathrm{h}} \frac{h_{jnf}}{\sqrt{\sum_{f'} h_{jnf'}^2}}. \tag{3.74}$$

3.4.5 Positive and Negative Input

In general, the input data $\bar{\mathcal{V}} \in \mathbb{R}^{P \times N \times D}$ is not restricted to non-negative data, but can contain *positive and negative* values. A non-negative representation can still be achieved by splitting all input elements into positive and negative components, by

$$v_{pnd+} = \frac{|\bar{v}_{pnd}| + \bar{v}_{pnd}}{2}, \tag{3.75}$$

$$v_{pnd-} = \frac{|\bar{v}_{pnd}| - \bar{v}_{pnd}}{2}. \tag{3.76}$$

This simple trick allows a non-negative representation of any kind of input data. The positive as well as the negative values can each be represented by a non-negative representation

$$v_{pnd+}, v_{pnd-} \geq 0, \quad \forall p \in \{1, ..., P\}, \forall n \in \{1, ..., N\}, \forall d \in [1, ..., D]. \tag{3.77}$$

This comes at the cost of an additional dimension, because the positive and negative values each have an individual representation.

The multidimensional representation can now either be treated like the multidimensional input data as discussed in the previous section, which results in a *strict* constraint, *i.e.* a separation of the positive and negative

values. The positive and negative values of the same position are then treated as individual features without any specific relation. In the following this is termed *strict non-negativity*. Contrariwise, the positive and negative values can only have a separated representation, but are used in combination, *e.g.* $v_{pnd+} - v_{pnd-}$, in the energy functions. Any desired interaction between the two parts can then be addressed via *weak* constraints, *i.e.* additional energy functions. In the following this is termed *weak non-negativity*. A comparison of the two types of constraints can be found in [42].

3.4.6 Strict Non-negativity

In the strict non-negative representation the positive and negative components are treated as two independent features. With the new feature dimension $F = 2 \cdot D$ the reconstruction energy is

$$E_{\text{strict-nn}} = \frac{1}{2} \sum_{p,n,f} (v_{pnf} - r_{pnf})^2. \tag{3.78}$$

The feature dimension can either be represented by the basis vectors or activities as discussed in section 3.4.3 and section 3.4.2.

3.4.7 Weak Non-negativity

In case of the weak non-negative representation, there exist non-negative feature dimensions for the input as well as for the reconstruction, the activations and the basis vectors. The additional feature dimension of the reconstruction can again either be represented by the basis vectors or activities as discussed in section 3.4.3 and section 3.4.2. However, in the reconstruction energy function

$$E_{\text{weak-nn}} = \frac{1}{2} \sum_{p,n,d} (v_{pnd+} - v_{pnd-} - r_{pnd+} + r_{pnd-})^2, \tag{3.79}$$

the negative values are subtracted from the positive values.

If the additional feature dimension is represented by the basis vectors, the energy function is

$$E_{\text{weak-nn}} = \frac{1}{2} \sum_{p,n,d} (v_{pnd+} - v_{pnd-} - \sum_{j} h_{jn}(\bar{w}_{pd+} + \bar{w}_{pd-}))^2, \tag{3.80}$$

with the gradient components for the basis vectors

$$(\nabla_{\bar{\mathcal{W}}_+} E_{\text{weak-nn}})^+ = \mathcal{R}_- \mathcal{H}^\top + \mathcal{V}_+ \mathcal{H}^\top, \tag{3.81}$$
$$(\nabla_{\bar{\mathcal{W}}_+} E_{\text{weak-nn}})^- = \mathcal{R}_+ \mathcal{H}^\top + \mathcal{V}_- \mathcal{H}^\top, \tag{3.82}$$
$$(\nabla_{\bar{\mathcal{W}}_-} E_{\text{weak-nn}})^+ = (\nabla_{\bar{\mathcal{W}}_+} E_{\text{weak-nn}})^-, \tag{3.83}$$
$$(\nabla_{\bar{\mathcal{W}}_-} E_{\text{weak-nn}})^- = (\nabla_{\bar{\mathcal{W}}_+} E_{\text{weak-nn}})^+, \tag{3.84}$$

and the activities

$$(\nabla_{\mathcal{H}} E_{\text{weak-nn}})^+ = \bar{\mathcal{W}}_+^\top \mathcal{R}_+ + \bar{\mathcal{W}}_-^\top \mathcal{R}_- + \bar{\mathcal{W}}_-^\top \mathcal{V}_+ + \bar{\mathcal{W}}_+^\top \mathcal{V}_-, \tag{3.85}$$
$$(\nabla_{\mathcal{H}} E_{\text{weak-nn}})^- = \bar{\mathcal{W}}_+^\top \mathcal{R}_- + \bar{\mathcal{W}}_-^\top \mathcal{R}_+ + \bar{\mathcal{W}}_-^\top \mathcal{V}_- + \bar{\mathcal{W}}_+^\top \mathcal{V}_+. \tag{3.86}$$
$$\tag{3.87}$$

If the additional feature dimension is represented by the activations, the energy function is

$$E_{\text{weak-nn}} = \frac{1}{2} \sum_{p,n,d} (v_{pnd+} - v_{pnd-} - \sum_j (h_{jn+} - h_{jn-}) \bar{w}_{pd})^2, \tag{3.88}$$

with the gradient components for the basis vectors

$$(\nabla_{\bar{\mathcal{W}}} E_{\text{weak-nn}})^+ = \mathcal{R}_- \mathcal{H}_-^\top + \mathcal{V}_+ \mathcal{H}_-^\top + \mathcal{R}_+ \mathcal{H}_+^\top + \mathcal{V}_- \mathcal{H}_+^\top, \tag{3.89}$$
$$(\nabla_{\bar{\mathcal{W}}} E_{\text{weak-nn}})^- = \mathcal{R}_+ \mathcal{H}_-^\top + \mathcal{V}_- \mathcal{H}_-^\top + \mathcal{R}_- \mathcal{H}_+^\top + \mathcal{V}_+ \mathcal{H}_+^\top, \tag{3.90}$$
$$\tag{3.91}$$

and the activities

$$(\nabla_{\mathcal{H}_+} E_{\text{weak-nn}})^+ = \bar{\mathcal{W}}^\top \mathcal{R}_+ + \bar{\mathcal{W}}^\top \mathcal{V}_-, \tag{3.92}$$
$$(\nabla_{\mathcal{H}_+} E_{\text{weak-nn}})^- = \bar{\mathcal{W}}^\top \mathcal{R}_- + \bar{\mathcal{W}}^\top \mathcal{V}_+, \tag{3.93}$$
$$(\nabla_{\mathcal{H}_-} E_{\text{weak-nn}})^+ = (\nabla_{\mathcal{H}_+} E_{\text{weak-nn}})^-, \tag{3.94}$$
$$(\nabla_{\mathcal{H}_-} E_{\text{weak-nn}})^- = (\nabla_{\mathcal{H}_+} E_{\text{weak-nn}})^+. \tag{3.95}$$
$$\tag{3.96}$$

In both cases, the sign for the gradients of the component with the positive values is opposite to the sign of the gradients of the component with the negative values. Hence, if *e.g.* a positive basis vector element w_{pj+} is increasing, its negative counterpart w_{pj-} will be decreasing by the same factor.

The subtraction in the model counteracts the idea of the parts-basedness achieved with a strict non-negative representation, because the purely additive model is the main reason for a parts-based decomposition, as discussed in section 3.2.2. However, the weak non-negativity allows for an increased flexibility of the model, especially when the energy function cannot be written in die strict form as in eq. (3.78). An example is the optical flow estimation algorithm that is discussed later on in chapter 4.

3.4.8 Orthogonality between Positive and Negative Reconstructions

To avoid an interaction between the positive and negative representations of a weak non-negative model, an orthogonality constraint can be imposed upon the two reconstructions, using the energy function

$$E_{\text{orthoSign}} = \frac{1}{2}\mathcal{R}_+^\top \mathcal{R}_- = \frac{1}{2}\sum_{p,n,d} r_{pnd+} r_{pnd-}. \tag{3.97}$$

The additional feature dimension of the reconstruction can again either be represented by the basis vectors or activities as discussed in section 3.4.3 and section 3.4.2.

If the additional feature dimension is represented by the basis vectors, the energy function becomes

$$E_{\text{orthoSign}} = \frac{1}{2}\sum_{p,n,d}\sum_{j} h_{jn}^2 (\bar{w}_{pd+}\bar{w}_{jpd-}), \tag{3.98}$$

with the gradient components for the basis vectors

$$(\nabla_{\bar{\mathcal{W}}_+} E_{\text{orthoSign}})^+ = \mathcal{R}_- \mathcal{H}^\top, \tag{3.99}$$

$$(\nabla_{\bar{\mathcal{W}}_+} E_{\text{orthoSign}})^- = 0, \tag{3.100}$$

$$(\nabla_{\bar{\mathcal{W}}_-} E_{\text{orthoSign}})^+ = \mathcal{R}_+ \mathcal{H}^\top, \tag{3.101}$$

$$(\nabla_{\bar{\mathcal{W}}_-} E_{\text{orthoSign}})^- = 0, \tag{3.102}$$

and the activities

$$(\nabla_{\mathcal{H}} E_{\text{orthoSign}})^+ = \bar{\mathcal{W}}_+^\top \mathcal{R}_- + \bar{\mathcal{W}}_-^\top \mathcal{R}_+, \tag{3.103}$$

$$(\nabla_{\mathcal{H}} E_{\text{orthoSign}})^- = 0. \tag{3.104}$$

$$\tag{3.105}$$

If the additional feature dimension is represented by the activations, the energy function becomes

$$E_{\text{orthoSign}} = \frac{1}{2} \sum_{p,n,d} \sum_{j} h_{jn+} h_{jn-} \bar{w}_{jpd}^2, \tag{3.106}$$

with the gradient components for the basis vectors

$$(\nabla_{\bar{\mathcal{W}}} E_{\text{orthoSign}})^+ = \mathcal{R}_+ \mathcal{H}_-^\top + \mathcal{R}_- \mathcal{H}_+^\top, \tag{3.107}$$
$$(\nabla_{\bar{\mathcal{W}}} E_{\text{orthoSign}})^- = 0, \tag{3.108}$$
$$\tag{3.109}$$

and the activities

$$(\nabla_{\mathcal{H}_+} E_{\text{orthoSign}})^+ = \bar{\mathcal{W}}^\top \mathcal{R}_-, \tag{3.110}$$
$$(\nabla_{\mathcal{H}_+} E_{\text{orthoSign}})^- = 0, \tag{3.111}$$
$$(\nabla_{\mathcal{H}_-} E_{\text{orthoSign}})^+ = \bar{\mathcal{W}}^\top \mathcal{R}_+, \tag{3.112}$$
$$(\nabla_{\mathcal{H}_-} E_{\text{orthoSign}})^- = 0. \tag{3.113}$$
$$\tag{3.114}$$

3.5 Translation-invariant NMF

A lot of the basis vectors learned with regular unsupervised learning algorithms, like sNMF, are simply shifted versions of few unique structured basis vectors. To reduce this redundancy, translation-invariant (also known as shift-invariant) learning algorithms [28] can be used. A translation invariant learning scheme is in good accordance with a FFNN, because the simple cell responses are also detected in a convolutional manner.

To achieve a translation invariant learning, each basis vectors is repeatedly shifted in a way that for each shift, the basis vector is centered at a new position in the entire input image. There are now P activations for a single basis vector in each input image, *i.e.* one activation for each shift. This allows the reconstruction to *activate* each basis vector at any position in the reconstruction, thus to be translation invariant. The activations store the absolute position of a pattern in the input vector and the corresponding basis vector describes the structure of the pattern.

During the learning process, the gradient for each shifted version of a single basis vector is reshifted to a reference position and all the reshifted

gradients are accumulated. As a consequence, the statistics gathered throughout the entire input image are used to update the basis vectors.

The benefits of having a translation invariant learning algorithm can be summarized as follows:

- Learning is performed on the entire input image, thus fewer training data is required and no key-point detection is needed to search for meaningful patterns in the input.
- Fewer basis vectors are required, because redundant translated versions are omitted.
- Instead of multiple local models, one global model is used to represent the entire input image. This global approach allows to influence the interactions between neighboring activations and therefore to address the superposition problem and topological sparsity.
- The basis vectors can be spatially restricted, which further enforces a parts-based decomposition and local receptive fields.

Due to change in dimensions for the activations, the notation for the translation-invariant NMF, introduced in section 3.2.5 needs to be adapted. The activation that corresponds to an input $\boldsymbol{V}_n$ and the basis vector $\boldsymbol{W}_j$ is now no longer a scalar value, but the activation vector $\boldsymbol{H}_{jn}$.

3.5.1 Reconstruction Energy

In case of a two dimensional, spatially arranged input, like an image or a vector field, each $p \in \{1, ..., P\}$ represents a two dimensional coordinate

$$\boldsymbol{x} = \begin{pmatrix} x \\ y \end{pmatrix}, \tag{3.115}$$

with $x \in [1, ..., X]$ and $y \in [1, ..., Y]$, $X, Y :=$ number of pixels in horizontal and vertical direction respectively and the relation $P = X \cdot Y$. The two dimensional coordinate is valid inside the interval $\boldsymbol{x} \in [\mathbf{1}, ..., \boldsymbol{X}]$, with

$$\boldsymbol{X} = \begin{pmatrix} X \\ Y \end{pmatrix}. \tag{3.116}$$

Now the translation invariant NMF is introduced as a special case of the transformation invariant NMF [28]. The reconstruction $r_n(\boldsymbol{x})$ at the two

dimensional pixel coordinate $\boldsymbol{x}$ is

$$r_n(\boldsymbol{x}) = \sum_{j,\boldsymbol{m}} r_{j\boldsymbol{m}n}(\boldsymbol{x}) = \sum_{j,\boldsymbol{m}} h_{jn}(\boldsymbol{m})\big(T(\boldsymbol{m})\bar{w}_j(\boldsymbol{x})\big), \tag{3.117}$$

with the set of matrices $T(\boldsymbol{m}), \boldsymbol{m} \in [\boldsymbol{1}, ..., \boldsymbol{X}]$, that describe shift operations which are applied to the basis vectors $\bar{\boldsymbol{W}}_j$. With

$$T(\boldsymbol{m})\bar{w}_j(\boldsymbol{x}) = \bar{w}_j(\boldsymbol{x} - \boldsymbol{m}) \tag{3.118}$$

we get the pixel value of the $j$the basis vector at the two dimensional position $(\boldsymbol{x} - \boldsymbol{m})$ and shift it by $\boldsymbol{m}$ to reconstruct the pixel $\boldsymbol{x}$. The corresponding activity is $h_{jn}(\boldsymbol{m})$. By combining eqs. (3.117), (3.118) and the two dimensional convolution operation

$$\sum_{\boldsymbol{m}} a(\boldsymbol{m})b(\boldsymbol{x} - \boldsymbol{m}) = \mathrm{conv}_2(\boldsymbol{A}, \boldsymbol{B})(\boldsymbol{x}) \tag{3.119}$$

the reconstruction for each pixel $r_n(\boldsymbol{x})$ and the image reconstruction $\boldsymbol{R}_n$ becomes

$$r_n(\boldsymbol{x}) = \sum_{j,\boldsymbol{m}} h_{jn}(\boldsymbol{m})\bar{w}_j(\boldsymbol{x} - \boldsymbol{m}) = \sum_j \mathrm{conv}_2(\boldsymbol{H}_{jn}, \bar{\boldsymbol{W}}_j)(\boldsymbol{x}), \tag{3.120}$$

$$\boldsymbol{R}_n = \sum_j \mathrm{conv}_2(\boldsymbol{H}_{jn}, \boldsymbol{W}_j). \tag{3.121}$$

The activations $\boldsymbol{H}_{jn} \in \mathbb{R}^{X \times Y}$ are now images with as many elements as there are pixels in the input. The anchors in the convolution in eq. (3.121) are set so that the activation $h_{jn}(\boldsymbol{m})$ corresponds to a shift of the center pixel of the corresponding basis vector to the pixel position $(\boldsymbol{m})$. The process is illustrated in fig. 3.3. Notice that for the reconstruction, each basis vector can be shifted to each pixel position. Therefore, the basis vectors can be spatially restricted by setting a *maximum receptive field size* (mRFS).

The new reconstruction in (3.121) leads to the following reconstruction energy term

$$E_{\mathrm{rec}} = \frac{1}{2}\sum_n \|\boldsymbol{V}_n - \boldsymbol{R}_n\|_2^2 = \frac{1}{2}\sum_n \|\boldsymbol{V}_n - \sum_j \mathrm{conv}_2(\boldsymbol{H}_{jn}, \bar{\boldsymbol{W}}_j)\|_2^2, \tag{3.122}$$

with the gradients for the activities

$$\nabla_{\boldsymbol{H}_{jn}} E_{\mathrm{rec}} = \underbrace{\mathrm{corr}_2(\boldsymbol{R}_n, \bar{\boldsymbol{W}}_j)}_{:=\left(\nabla_{\boldsymbol{H}_{jn}} E_{\mathrm{rec}}\right)^+} - \underbrace{\mathrm{corr}_2(\boldsymbol{V}_n, \bar{\boldsymbol{W}}_j)}_{:=\left(\nabla_{\boldsymbol{H}_{jn}} E_{\mathrm{rec}}\right)^-}, \tag{3.123}$$

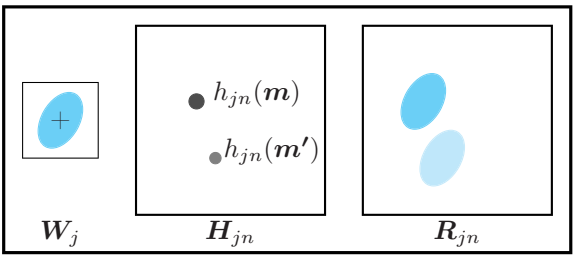

Figure 3.3: Illustration of the translation invariance. For each input, each basis vector $\boldsymbol{W}_j$ has a corresponding activation image $\boldsymbol{H}_{jn}$. For the partial reconstruction $\boldsymbol{R}_{jn}$ the basis vector is shifted in a way that the center of the basis vector is located on the position of a single activation. For each activation pixel $\boldsymbol{m}$ there is a partial reconstruction $\boldsymbol{R}_{jn\boldsymbol{m}}$ and the combined partial reconstruction is $\boldsymbol{R}_{jn} = \sum_{\boldsymbol{m}} \boldsymbol{R}_{jn\boldsymbol{m}}$.

with the two dimensional correlation

$$\sum_{\boldsymbol{m}} a(\boldsymbol{m}) b(\boldsymbol{x} + \boldsymbol{m}) = \mathrm{corr}_2(\boldsymbol{A}, \boldsymbol{B})(\boldsymbol{x}). \tag{3.124}$$

The gradients for the basis vectors are

$$\nabla_{\bar{\boldsymbol{W}}_j} E_{\mathrm{rec}} = \underbrace{\sum_n \mathrm{corr}_2(\boldsymbol{R}_n, \boldsymbol{H}_{jn})}_{:= \left(\nabla_{\bar{\boldsymbol{W}}_j} E_{\mathrm{rec}}\right)^+} - \underbrace{\sum_n \mathrm{corr}_2(\boldsymbol{V}_n, \boldsymbol{H}_{jn})}_{:= \left(\nabla_{\bar{\boldsymbol{W}}_j} E_{\mathrm{rec}}\right)^-}. \tag{3.125}$$

The derivation of the gradients can be found in appendix C.1.

3.5.2 Sparse Activations

The sparsity energy function has to be adapted, due to the new dimensionality of the activations. For the translation invariant NMF the energy function becomes

$$E_{\mathrm{h}} = \lambda_{\mathrm{h}} \sum_{n,j,\boldsymbol{m}} h_{jn}(\boldsymbol{m}), \tag{3.126}$$

with the positive gradient component

$$\left(\nabla_{h_{jn}(\boldsymbol{m})} E_{\mathrm{h}}\right)^+ = \lambda_{\mathrm{h}}. \tag{3.127}$$

3.5.3 Orthogonality between Positive and Negative Representation

In case of a multidimensional input, the translation invariant version of orthogonality between the positive and negative components, as discussed in section 3.4.8 is

$$E_{\text{orthoSign}} = \frac{1}{2}\mathcal{R}_+^\top \mathcal{R}_- = \frac{1}{2}\sum_{\boldsymbol{x},n,d} r_{nd+}(\boldsymbol{x}) r_{nd-}(\boldsymbol{x}). \tag{3.128}$$

The additional feature dimension of the reconstruction can again either be represented by the basis vectors or activities as discussed in section 3.4.3 and section 3.4.2.

If the additional feature dimension is represented by the basis vectors, the energy function becomes

$$E_{\text{orthoSign}} = \frac{1}{2}\sum_{\boldsymbol{x},n,d}\sum_{j,\boldsymbol{m}} h_{jn}(\boldsymbol{m})^2 (\bar{w}_{d+}(\boldsymbol{x}-\boldsymbol{m})\bar{w}_{d-}(\boldsymbol{x}-\boldsymbol{m})), \tag{3.129}$$

with the gradient components for the basis vectors

$$(\nabla_{\bar{\boldsymbol{W}}_{j+}} E_{\text{orthoSign}})^+ = \sum_n \text{corr}_2(\boldsymbol{R}_{n-}, \boldsymbol{H}_{jn}), \tag{3.130}$$

$$(\nabla_{\bar{\boldsymbol{W}}_{j+}} E_{\text{orthoSign}})^- = 0, \tag{3.131}$$

$$(\nabla_{\bar{\boldsymbol{W}}_{j-}} E_{\text{orthoSign}})^+ = \text{corr}_2(\boldsymbol{R}_{n+}, \boldsymbol{H}_{jn}), \tag{3.132}$$

$$(\nabla_{\bar{\boldsymbol{W}}_{j-}} E_{\text{orthoSign}})^- = 0, \tag{3.133}$$

and the activities

$$(\nabla_{\boldsymbol{H}_{jn}} E_{\text{orthoSign}})^+ = \text{corr}_2(\boldsymbol{R}_{n-}, \bar{\boldsymbol{W}}_{j+}) + \text{corr}_2(\boldsymbol{R}_{n+}, \bar{\boldsymbol{W}}_{j-}), \tag{3.134}$$

$$(\nabla_{\boldsymbol{H}_{jn}} E_{\text{orthoSign}})^- = 0. \tag{3.135}$$

If the additional feature dimension is represented by the activations, the energy function becomes

$$E_{\text{orthoSign}} = \frac{1}{2}\sum_{\boldsymbol{x},n,d}\sum_{j} h_{jn+}(\boldsymbol{m}) h_{jn-}(\boldsymbol{m}) \bar{w}_{jd}(\boldsymbol{c}-\boldsymbol{m})^2, \tag{3.136}$$

with the gradient components for the basis vectors

$$\begin{aligned}(\nabla_{\bar{\boldsymbol{W}}_j} E_{\text{orthoSign}})^+ = &\sum_n \text{corr}_2(\boldsymbol{R}_{n+}, \boldsymbol{H}_{jn-}) \\ &+ \text{corr}_2(\boldsymbol{R}_{n-}, \boldsymbol{H}_{jn+}),\end{aligned} \tag{3.137}$$

$$(\nabla_{\bar{\boldsymbol{W}}_j} E_{\text{orthoSign}})^- = 0, \tag{3.138}$$

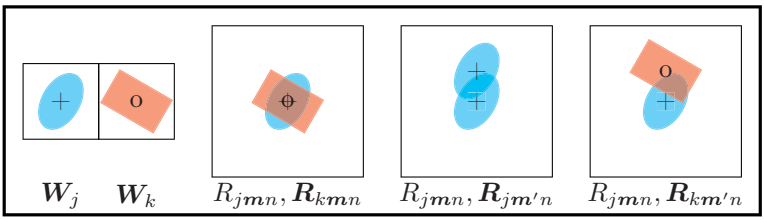

Figure 3.4: Visualization of overlapping receptive fields. On the left are two example basis vectors and on the right three examples for overlapping receptive fields. From left to right: Overlap of two different basis vectors anchored at the same position, overlap of the partial reconstruction of the same basis vector anchored at different positions and overlap of the partial reconstruction of the two different basis vectors anchored at different positions.

and the activities

$$(\nabla_{\boldsymbol{H}_{jn+}} E_{\text{orthoSign}})^+ = \text{corr}_2(\boldsymbol{R}_{n-}, \bar{\boldsymbol{W}}_j), \tag{3.139}$$
$$(\nabla_{\boldsymbol{H}_{jn+}} E_{\text{orthoSign}})^- = 0, \tag{3.140}$$
$$(\nabla_{\boldsymbol{H}_{jn-}} E_{\text{orthoSign}})^+ = \text{corr}_2(\boldsymbol{R}_{n+}, \bar{\boldsymbol{W}}_j), \tag{3.141}$$
$$(\nabla_{\boldsymbol{H}_{jn-}} E_{\text{orthoSign}})^- = 0. \tag{3.142}$$

This completes the derivation of the reconstruction-based orthogonality in case of the translation invariant NMF.

3.5.4 Enforced Topological Sparsity

One of the benefits of having a translation invariant learning algorithm is that interactions between neighboring activations can be addressed. In section 3.3.6, the idea of enforcing parts-basedness is introduced in terms of a *local* competition, *i.e.* inhibition, between different basis vectors. This idea is now extended for a *lateral* competition between spatial neighboring activations. The combined local and lateral competition favors *topological sparse activations* and as a consequence *parts-based and prototypical basis vectors*.

The topological sparsity is achieved via a weak constraint, *i.e.* an additional energy function that penalizes overlapping receptive fields.There are three cases as depicted in fig. 3.4. Overlaps between different basis vectors $\bar{\boldsymbol{W}}_j$ and $\bar{\boldsymbol{W}}_k$ at the same spatial position, overlaps between shifted

versions of the same basis vector $\bar{\boldsymbol{W}}_j$ and overlaps between shifted versions of different basis vectors $\bar{\boldsymbol{W}}_j$ and $\bar{\boldsymbol{W}}_k$. Or, in simpler terms, any overlap of a basis vector $\bar{\boldsymbol{W}}_j$ shifted by $\boldsymbol{m}$ and the reconstruction at the same position, except the part of the reconstruction that belongs to the identical basic vector and the identical shift $\boldsymbol{m}$.

With $\sum_{k,\boldsymbol{m}'} \boldsymbol{R}_{k\boldsymbol{m}'n} = \boldsymbol{R}_n$ the energy function for local and lateral competition is

$$E_p = \sum_{n,j,\boldsymbol{m}} \boldsymbol{R}_{j\boldsymbol{m}n}^\top \Big(\sum_{k\neq j} \boldsymbol{R}_{k\boldsymbol{m}n} + \sum_{\boldsymbol{m}'\neq\boldsymbol{m}} \boldsymbol{R}_{j\boldsymbol{m}'n} + \sum_{k\neq j,\boldsymbol{m}'\neq\boldsymbol{m}} \boldsymbol{R}_{k\boldsymbol{m}'n}\Big) \tag{3.143}$$

$$= \frac{1}{2} \sum_{n,j,\boldsymbol{m}} \boldsymbol{R}_{j\boldsymbol{m}n}^\top \Big(\boldsymbol{R}_n - \boldsymbol{R}_{j\boldsymbol{m}n}\Big) \tag{3.144}$$

$$= \underbrace{\frac{1}{2} \sum_i \boldsymbol{R}_n^\top \boldsymbol{R}_n}_{:=E_{p1}} - \underbrace{\frac{1}{2} \sum_{n,j,\boldsymbol{m}} \boldsymbol{R}_{j\boldsymbol{m}n}^\top \boldsymbol{R}_{j\boldsymbol{m}n}}_{:=E_{p2}}, \tag{3.145}$$

with the gradients for the activities

$$\nabla_{\boldsymbol{H}_{jn}} E_p = \nabla_{\boldsymbol{H}_{jn}} E_{p1} - \nabla_{\boldsymbol{H}_{jn}} E_{p2} \tag{3.146}$$

$$= \mathrm{corr}_2(\boldsymbol{R}_n, \bar{\boldsymbol{W}}_j) - \boldsymbol{H}_{jn} \bar{\boldsymbol{W}}_j^\top \bar{\boldsymbol{W}}_j \tag{3.147}$$

and the gradients for the basis vectors

$$\nabla_{\bar{\boldsymbol{W}}_j} E_p = \nabla_{\bar{\boldsymbol{W}}_j} E_{p1} - \nabla_{\bar{\boldsymbol{W}}_j} E_{p2} \tag{3.148}$$

$$= \sum_n \mathrm{corr}_2(\boldsymbol{R}_n, \boldsymbol{H}_{jn}) - \bar{\boldsymbol{W}}_j \sum_n \boldsymbol{H}_{jn}^\top \boldsymbol{H}_{jn}. \tag{3.149}$$

The detailed derivation of the gradients can be found in appendix C.2. The gradients for the first part of the competition energy term E_{p1} (3.146), (3.148) are identical with the positive components of the gradients of the translation invariant reconstruction term (3.123), (3.125) and therefore do not need to be computed again for the energy function E_p. Thus, the gradients of the competition term come with negligible additional computational costs.

3.5.5 VNMF Learning Algorithm

The translation invariant learning algorithm can be combined with all the extensions introduced in the section on non-negative representations of

multidimensional data 3.4. One particular case, the *vector non-negative matrix factorization* (VNMF) is used to learn the generic, prototypical patterns for the proposed FFNN hierarchy. It is a combination of translation invariant reconstructions, sparsity in the activations, strong non-negativity and enforced parts-basedness. The energy function is

$$\begin{aligned} E_{\text{VNMF}} = & \frac{1}{2} \sum_{n,f} \| \boldsymbol{V}_{nf} - \sum_{j} \text{conv}_2(\boldsymbol{H}_{jn}, \bar{\boldsymbol{W}}_{jf}) \|_2^2 \\ & + \lambda_p \frac{1}{2} \sum_{n,f} \sum_{j,\boldsymbol{m}} \boldsymbol{R}_{j\boldsymbol{m}nf}^{\top} \Big(\boldsymbol{R}_{nf} - \boldsymbol{R}_{j\boldsymbol{m}nf} \Big) \\ & + \lambda_{\text{h}} \sum_{n,j,\boldsymbol{m}} h_{jn}(\boldsymbol{m}). \end{aligned} \tag{3.150}$$

The gradients can be directly derived from the equations (3.123), (3.125), (3.127), (3.146) and (3.148). The learning algorithm for the VNMF is in principle identical to the one proposed for the sNMF algorithm in section 3.3.5. However, due to the translation invariance, the gradient components are calculated per image, resulting in a slightly modified learning algorithm:

- Preprocessing
  - Normalize $\mathcal{V} = \frac{\mathcal{V}}{\max(\mathcal{V})}$,
  - initialize $\mathcal{H}$ and $\mathcal{W}$ randomly.
- Loop for i iterations
  1. Loop for each of the N inputs
     a) Calculate $\boldsymbol{R}_n = \sum_j \text{conv}_2(\boldsymbol{H}_{jn}, \bar{\boldsymbol{W}}_{jf})$,
     b) update $\boldsymbol{H}_{jn} \rightarrow \boldsymbol{H}_{jn} \circ \frac{(\nabla_{\boldsymbol{H}_{jn}} E_{\text{VNMF}})^-}{(\nabla_{\boldsymbol{H}_{jn}} E_{\text{VNMF}})^+}, \quad \forall j \in \{1, ..., J\}$,
     c) calculate $\boldsymbol{R}_n = \sum_j \text{conv}_2(\boldsymbol{H}_{jn}, \bar{\boldsymbol{W}}_{jf})$,
     d) calculate $(\nabla_{\mathcal{W}} E_{\text{VNMF}})_n^+$ and $(\nabla_{\mathcal{W}} E_{\text{VNMF}})_n^-$
  2. update $\mathcal{W} \rightarrow \mathcal{W} \circ \frac{(\nabla_{\mathcal{W}} E_{\text{VNMF}})^-}{(\nabla_{\mathcal{W}} E_{\text{VNMF}})^+}$, with
     $(\nabla_{\mathcal{W}} E_{\text{VNMF}})^+ = \sum_n (\nabla_{\mathcal{W}} E_{\text{VNMF}})_n^+$ and
     $(\nabla_{\mathcal{W}} E_{\text{VNMF}})^- = \sum_n (\nabla_{\mathcal{W}} E_{\text{VNMF}})_n^-$,
  3. normalize $\bar{\boldsymbol{W}}_j = \frac{\boldsymbol{W}_j}{\sqrt{\sum_p w_{pj}^2}}, \quad \forall j \in \{1, ..., J\}$.

The algorithm has two kinds of free parameters. First, the *basis vector parameters*, *i.e.* the number of basis vectors J and the maximum receptive field size $mRFS$. And second, the *energy parameters*, *i.e.* the sparsity parameter λ_{h} and the parameter λ_p for the enforced parts-basedness. The influence of these parameters on the pattern learning and the classification will be discussed in the corresponding chapters.

One of the benefits of the proposed energy function is that all three energy functionals scale with the activation, thus their relative contribution to the overall energy is coupled. As a consequence, the parameters λ_{h} and λ_p can be set independent of the total number of input images as well as the size of the images and the occurrences of each basis vector. This makes the algorithm robust and easy to parametrize. The default parameters for the energy function are set to $\lambda_{\mathrm{h}} = 0.1$ and $\lambda_p = 0.2$.

3.6 Algorithm Summary

The algorithms introduced in the chapter are used to learn the simple cell patterns of the proposed FFNN for biological motion recognition. For the optical flow estimation (discussed in chapter 4), the proposed OFE algorithm is based on the multidimensional, translation-invariant VNMF algorithm. The multidimensional vector field is represented by weak non-negativity (see section 3.4.7 and the additional feature dimension is represented in the activations (see section 3.4.3). To care for the ambiguous representations encountered with the weak non-negativity, the penalization of opposing directions (see section 3.5.3) is included into the optimization.

For the feature layers, the VNMF algorithm is applied to learn the simple cell patterns. In case of the optical flow patterns, the multidimensional vector fields are represented by strict non-negativity (see section 3.4.6) and therefore do not require a penalization of the opposing motion directions. As opposed to the OFE algorithm, the additional dimensions are represented in the basis vectors (see section 3.4.2) and not in the activations. For the static stream, thus the gradient patterns, the input is not multidimensional and the algorithm does not need an explicit multidimensional representation.

All three algorithms[4)] enforce non-negativy and have the sparsity and inhibition term included into the optimization.

[4)] In appendix D there is a fourth, non-translation-invariant algorithm based on sNMF, adapted for spatio-temporal data.

4 Optical Flow Estimation

As discussed in the introduction, motion is an useful source of information for a variety of tasks, such as the recognition of human actions, gestures or face expressions. Motion couples two important properties, form and temporal variations. Rigid forms can only change consistently, thus motion patterns are highly correlated to the underlying structure of the moving object.

There are multiple ways to describe motion, *e.g.* by spatio-temporal trajectories of key-points or via dynamical models. A representation that captures the relations between movements and shapes are *optical flow fields* (OF-fields). OF-fields describe the movement of every pixel in an image $\mathcal{I}(\boldsymbol{x})$ with a two dimensional motion vector

$$\mathcal{V}(\boldsymbol{x}) = \begin{pmatrix} \mathcal{V}_x(\boldsymbol{x}) \\ \mathcal{V}_y(\boldsymbol{x}) \end{pmatrix}. \tag{4.1}$$

Due to the dense representation the OF-fields are independent of pre-processing such as key-point or object detection that are necessary for trajectories or model-based representations of motion. Since every pixel has a motion vector $\mathcal{V}(\boldsymbol{x})$, OF-fields can be easily combined with other pixel-based information, such as gradient- or color information. Unlike the color values of an image, an OF-field cannot be measured, but has to be estimated.

Optical flow estimation (OFE) is a vivid research topic that was originally introduced by Horn and Schunk in 1981 [50]. To successfully estimate an OF-field multiple assumptions have to be made. Some of them lead to rather reasonable simplifications while others lead to systematical errors, highlighting the need for robust OFE-algorithms.

In this chapter a novel OFE-algorithm based on non-negative, translation invariant sparse coding is introduced. The chapter is organized as follows: First, the problem of OFE is introduced together with necessary conceptual assumptions and conditions for OFE-algorithms. The section on related work critically discusses recent approaches on OFE including benefits and

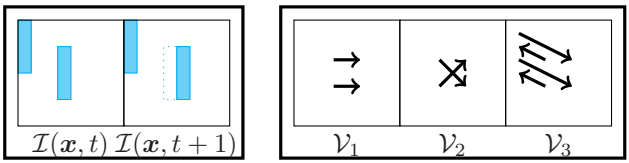

Figure 4.1: Illustrative examples for the ill-constraint nature of OFE. All three OF-fields $\mathcal{V}_1$, $\mathcal{V}_2$ and $\mathcal{V}_3$ can explain the displacement between the two consecutive images $\mathcal{I}(t)$ and $\mathcal{I}(t+1)$.

drawbacks in relation to the proposed method. Then the conceptual idea of our OFE-algorithm and its mathematical formulation is stated. Finally examples on real world data are shown.

4.1 Problem Formulation

In the following the basic ideas and assumptions underlying OFE are discussed on the example that is illustrated in fig. 4.1.

Looking at the two consecutive images $\mathcal{I}(t)$ and $\mathcal{I}(t+\Delta T)$ the *intuitive* interpretation is that the *bar* in the middle is moving *one step* to the right, which is represented in the optical flow field $\mathcal{V}_1$. But this is not the only possible optical flow field that explains the difference between the two images. Alternatively, the upper part of the bar could be moving to the lower right, while the lower part is moving to the upper right position ($\mathcal{V}_2$) or the bar could actually be moving to the corner and the bar in the corner could be moving into the middle of the image ($\mathcal{V}_3$). Even though there is no *correct* answer our human intuition favors $\mathcal{V}_1$ over the others, because we make implicit assumptions to find the most reasonable explanation, *i.e.* the explanation that is most consistent with our past experience. The first and most important is

Assumption 4.1 (Brightness Consistency (BCA))**.** The gray value of each pixel element between two time steps t and $t+\Delta T$ does not change:

$$\mathcal{I}(\boldsymbol{x}, t) = \mathcal{I}(\boldsymbol{x} + \mathcal{V}(\boldsymbol{x})\Delta T), t + \Delta T). \tag{4.2}$$

I.e. gray value changes are only caused by motion and not *e.g.* by illumination changes.

Assumption A4.1 is reasonable if there are only small movements between the two images, because for larger movements it is more likely that novel objects enter the image, objects leave the image or objects get occluded. In all three cases eq. (4.2) is no longer valid, because the novel or lost pixel cannot be found in the corresponding image. However, those violations of assumption A4.1 depend on the nature of the movement as well as on the time step ΔT and are neglectable under

Condition 4.1. The time between two frames ΔT is so small that the motion $\mathcal{V}$ of each pixel is smaller than the change in the local image structure.

The next assumption is

Assumption 4.2 (Continuous Motion)**.** Motion is a continuous process and thus the movement of objects between two consecutive frames is locally bounded.

As long as condition 4.1 is fulfilled assumption A4.2 is valid for all physical objects human observe in their natural surrounding. Due to assumption A4.2 we intuitively favor $\mathcal{V}_1$ over $\mathcal{V}_3$.

The first two assumptions, especially the BCA are of fundamental importance and almost all OFE-algorithms are based on them. However, they are simplifications and are systematically violated *e.g.* in case of occlusions (where new objects can appear or disappear) or when objects enter or leave the image. The consequence is that OFE-algorithms have to be highly robust, not only to image noise, but to violations of the underlying assumptions as well.

Because of assumptions A4.1 and A4.2 the OF-field $\mathcal{V}_3$ is discarded, yet there are still two possible options, $\mathcal{V}_1$ and $\mathcal{V}_2$. The reason why humans favor $\mathcal{V}_1$ over $\mathcal{V}_2$ is that the two blue pixels are not seen independently, but are grouped together to become a vertical *bar*, that is a non-deformable rigid object. For rigid objects, there is the assumption

Assumption 4.3 (Rigid Object Motion)**.** On a local scale, pixels of rigid objects have similar motion.

Assumption A4.3 is again motivated by physical constraints. However, like the first two assumptions it is only valid under condition 4.1 and depending on the similarity model only for local regions.*I.e.* with the projection equations it can be shown that rigid objects movements parallel to the projection plane can be described by affine models.

If all three assumptions are applied to the problem depicted in fig. 4.1, the OF-field $\mathcal{V}_1$ is the most probable interpretation. Unfortunately, assumption A4.3 has a crucial condition, *i.e.* it must be known which pixels belong to the same object. This requires a perfect image segmentation not only for objects, but for rigid objects. *E.g.* it is not enough to separate a human from a background, but his limbs must be segmented as well. Such a fine segmentation is not available in general. The problem at hand is actually a hen-and-egg dilemma, because if the OF would be known, the knowledge could be used with the model assumption to segment the rigid objects. On the other hand if the segmentation is known, it can be combined with assumption A4.3 to solve the OFE-problem.

Assumption A4.3 is of fundamental importance for OFE-algorithms, because it describes a way *spatial relations* can be incorporated into the OFE-problem. There are *temporal relations* for OF-fields as well, *e.g.* that movements and thus OF-fields are often temporally smooth. Here we encounter a similar problem as for the introduced spatial relations. Because the temporal smoothness is not valid at the beginning and at the end of movements, just like the spatial relations are not valid at the beginning and end of rigid objects. So in addition to a spatial segmentation, a *temporal segmentation* is required.

The analysis of the optical flow estimation problem as illustrated in fig. 4.1 can be summerized as follows:

Optical flow estimation is an ill posed problem that can only be solved by including reasonable assumptions derived from physical constraints found in the real world. The first and most important is the brightness constancy assumption. Further assumptions include spatial or temporal relations.

Before the discussion about the spatial relations is continued, a mathematical formulation based on the BCA equation (4.2) is introduced together with two algorithmic approaches of how to tackle the OFE-problem.

4.1.1 General Algorithmic Approaches

There are two different algorithmic approaches how to solve the OFE-problem using the BCA equation (4.2), *correlation methods* and *differential methods*. While correlation methods try to find similarities between two consecutive images $\mathcal{I}(\boldsymbol{x}, t)$ and $\mathcal{I}(\boldsymbol{x}, t+1)$[1)], differential methods focus on the differences, thus the temporal derivation $\mathcal{I}_t(\boldsymbol{x}, t) = \mathcal{I}(\boldsymbol{x}, t+1) - \mathcal{I}(\boldsymbol{x}, t)$. In other words, the correlation methods try to find correspondence where

[1)] Without the loss of generality the the time step is set to $\Delta T = 1$.

the difference is low, while the differential methods try to find areas with high temporal difference and then search for the motion that explains this temporal derivation. Even though they are both derived from the BCA the two methods follow different principles. To successfully estimate the optical flow, both methods relay on *image structure* which will be discussed for each method.

4.1.2 Correlation Methods

Correlation methods or *patch matching algorithms* take a pattern $A(\boldsymbol{x})$ within a window around pixel $\boldsymbol{x}$ out of image $\mathcal{I}(\boldsymbol{x}, t)$ and try to find it within a local search area $S(\boldsymbol{x})$ around pixel $\boldsymbol{x}$ in the next image $\mathcal{I}(\boldsymbol{x}, t+1)$. The position $\boldsymbol{y} \in S(\boldsymbol{x})$ with the highest correlation is then used to calculate the optical flow vector $\mathcal{V}(\boldsymbol{x}) = \boldsymbol{y} - \boldsymbol{x}$. This local pattern search is done for each pixel $\boldsymbol{x} \in \boldsymbol{X}$ in the entire image. The limited search radius is a direct consequence of the continuous motion assumption A4.2.

The right choice of an appropriate window $A(\boldsymbol{x})$ is crucial. The assumption is that all pixels in this window have the same translational movement and it is thus directly related to assumption A4.3. However, assumption A4.3 refers to pixels on the same rigid object and the knowledge about pixel-to-object correspondence is in general unknown, so a simplified assumption is

Assumption 4.4 (Predefined Local Neighborhood)**.** Pixels on a predefined local neighborhood $A(\boldsymbol{x})$ around pixel $\boldsymbol{x}$ share the same translational movement $\mathcal{V}(\boldsymbol{x})$.

Assumption A4.4 is of course violated on object borders, but often a valid simplification. The choice of the pattern size depends on the local image structure. Correlation methods are *local methods* because the translational model is only valid on a local scale, so the spatial relations are restricted to small local areas. Local methods are prone to the *aperture problem*. In a local window, even with spatial relations, the movement direction is not always clearly defined. Only in the context of a larger structure this problem can be solved.

Correlation methods are widely applied, because they are simple and easy to implement. However, to find unique correspondences correlation methods require image structures. The translational model restricts the spatial relations to local areas that are analyzed independently, which makes correlations methods suffer the aperture problem.

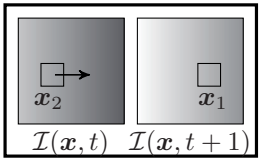

Figure 4.2: Illustrative example for the differential method. The pixel at position $\boldsymbol{x}_2$ marked in image $\mathcal{I}(\boldsymbol{x}, t)$ moves to position $\boldsymbol{x}_1$ in image $\mathcal{I}(\boldsymbol{x}, t+1)$. Thus, the gray value at position $\boldsymbol{x}_1$ changes, which results in a temporal gradient.

4.1.3 Differential Methods

Correlation methods take the BCA and try to find similar patterns in two consecutive frames. The displacement between the two positions is then the movement of that particular pattern. Differential methods have a different approach. The BCA states that all temporal changes in an image sequence can only be caused by pixel movements. Wherever a temporal variation occurs, movement takes place. Differential methods try to find the motion that is related to that temporal derivations.

Image structure is crucial for differential methods, because if movements occur in homogenous regions, there is no temporal derivation at all. Only pixels with different gray values can impose temporal variations. Motion is in general only observable in structured image regions.

One characteristic of image structure is the presence of spatial derivations $\mathcal{I}_x = \frac{\partial \mathcal{I}}{\partial x}$ and $\mathcal{I}_y = \frac{\partial \mathcal{I}}{\partial y}$. It is now discussed on a simple example how differential methods use the temporal and spatial image derivations to estimate an optical flow field $\mathcal{V}$.

Image $\mathcal{I}(\boldsymbol{x}, t)$ as shown in fig. 4.2, has a constant linear gradient $\mathcal{I}_x = c$ along the x-Axis. The motion for each pixel in the entire image is $\mathcal{V}_x = 1$ and $\mathcal{V}_y = 0$. The shifted image $\mathcal{I}(\boldsymbol{x}, t+1)$ for the next time step is shown in fig. 4.2. The gray value of the pixel at position $\boldsymbol{x}_1$ for time step t is $\mathcal{I}(\boldsymbol{x}_1, t)$. Because of the constant image gradient $\mathcal{I}_x$ the gray value of the pixel at position $\boldsymbol{x}_1$ will change its gray value linearly depending on the spatial difference between $\boldsymbol{x}_1$ and the former position $\boldsymbol{x}_2$ of the pixel that is now at position $\boldsymbol{x}_1$. In mathematical terms

$$\mathcal{I}(\boldsymbol{x}_1, t+1) = \mathcal{I}(\boldsymbol{x}_1, t) + \mathcal{I}_x(\boldsymbol{x}_1)(\boldsymbol{x}_2 - \boldsymbol{x}_1) \tag{4.3}$$

$$\mathcal{I}(\boldsymbol{x}_1, t+1) - \mathcal{I}(\boldsymbol{x}_1, t) = \mathcal{I}_x(\boldsymbol{x}_1)(\boldsymbol{x}_2 - \boldsymbol{x}_1) \tag{4.4}$$

$$\mathcal{I}_t(\boldsymbol{x}_1) = -\mathcal{I}_x(\boldsymbol{x}_1)\mathcal{V}_x(\boldsymbol{x}_1), \tag{4.5}$$

with the given temporal derivation $\mathcal{I}_t(\boldsymbol{x}_1) = \mathcal{I}(\boldsymbol{x}_1, t+1) - \mathcal{I}(\boldsymbol{x}_1, t)$ and spatial derivation $\mathcal{I}_x(\boldsymbol{x}_1)$ and the unknown motion vector $\mathcal{V}_x(\boldsymbol{x}_1) = \boldsymbol{x}_1 - \boldsymbol{x}_2$. Eq. (4.5) can be reformulated to

$$\mathcal{V}_x(\boldsymbol{x}_1) = -\mathcal{I}_t(\boldsymbol{x}_1)/\mathcal{I}_x(\boldsymbol{x}_1). \tag{4.6}$$

For a positive gradient $c > 0$ and motion in positive x-direction $\mathcal{V}_x$ the temporal gradient is negative $\mathcal{I}_t < 0$, because the former pixel with the higher gray value will get replaced by a pixel with a lower gray value. This relation is mathematically described by eq. (4.6), which can be solved to get the optical flow vector $\mathcal{V}_x(\boldsymbol{x}_1)$.

The example at hand illustrates how the optical flow can be calculated out of the temporal and spatial image derivatives. However, two relevant simplifications are made in this first example. First, only one-dimensional movement was considered. For the two dimensional problem eq. (4.5) becomes

$$\mathcal{I}_x(\boldsymbol{x})\mathcal{V}_x(\boldsymbol{x}) + \mathcal{I}_y(\boldsymbol{x})\mathcal{V}_y(\boldsymbol{x}) + \mathcal{I}_t(\boldsymbol{x}) = 0, \tag{4.7}$$

which is the linear *brightness constancy equation* (BCE). It can be derived directly from the general BCA eq. (4.2) by a first order Taylor series approximation. The BCE is the basis for almost all OFE-algorithms based on differential methods and also of significant importance for the approach proposed in this thesis. It is the linearized version of the BCA, which brings us to the second simplification used in the above example. Only *linear* spatial gradients were considered, which is reflected by the

Assumption 4.5 (Linear Spatial Gradients)**.** On a local scale all spatial gradients $\mathcal{I}_{\boldsymbol{x}}(\boldsymbol{x})$ of image $I(\boldsymbol{x})$ are linear.

Assumption A4.5 is only valid for small local scales and a simplification in most cases. A simple way to make image gradients more linear is to filter the image with an Gaussian filter before calculating the image gradients.

A closer look at the BCE eq. (4.7) reveals that for each pixel $\boldsymbol{x}$ there is one equation and two unknowns $\mathcal{V}_x(\boldsymbol{x})$ and $\mathcal{V}_y(\boldsymbol{x})$, so the problem is *ill-constrained* and needs additional assumptions like *i.e.* the spatial relations (assumption A4.3). Also, if there are no spatial derivations the BCE eq. (4.7) yields no information which again shows the dependence on image structures for differential methods.

4.1.4 Method Comparison

Correlation methods directly search similar patterns and do not need any linearization assumptions like the differential methods. They are limited to translational models, because pattern deformations are hard to include into a correlation scheme. Moreover correlation methods are always locally bound and suffer from the aperture problem. It is easier to include more complex motion models into differential methods, because the OFE-problem for differential methods is not a patch matching, but an optimization problem. The BCE 4.7 is can be minimized by different optimization techniques. A very simple way is to use an euclidean error measure and minimize the differential and convex function

$$\min_{\mathcal{V}_x,\mathcal{V}_y} \sum_{\boldsymbol{x}} \|\mathcal{I}_x(\boldsymbol{x})\mathcal{V}_x(\boldsymbol{x}) + \mathcal{I}_y(\boldsymbol{x})\mathcal{V}_y(\boldsymbol{x}) + \mathcal{I}_t(\boldsymbol{x})\|_2^2 \tag{4.8}$$

with gradient descent. Because the differential methods allow the use of optimization techniques learning algorithms can be incorporated in a straightforward manner. This is the main reason differential methods are chosen as the algorithmic basis for the OFE-algorithm in this thesis. The learning algorithms introduced in chapter 3 are all based on multiplicative gradient descent and thus need a differential energy function to be applied to.

However, as discussed in the above section, the differential method (as well as the correlation method) is only an algorithmic approach for the BCA and additional assumptions are required. OFE consists of two sub-problems:

1. Conceptual problem formulation (physical assumption and the general algorithmic approach).

2. Solving the formulated problem (robust optimization methods).

The proposed learning based approach tackles both sub-problems. A set of local neighborhoods which fulfill assumption A4.3 is learned and the OFE problem is solved with an algorithm derived from the NMF algorithms introduced in chapter 3.

Next, related work on how to include the spatial relations to the BCE and how they are solved is discussed. The discussion will be focused on a conceptual problem formulation rather then on the optimization techniques.

4.2 Related Work

In the OFE literature there are numerous optimization techniques that describe how to robustly solve the BCE and its additional energy functionals. *E.g.* Sun et al. [98] show that if state-of-the-art optimization methods are applied to the original formulated OFE algorithm introduced by Horn and Schunk [50] it is competitive with most of the current OFE-algorithms. The conceptual ideas of how to incorporate additional constraints to the BCE are analyzed in the following, especially how to include spatial relations, thus assumption A4.3.

4.2.1 Horn and Schunk

The BCE (4.7) was originally formulated by Horn and Schunk [50] with an additional term that is related to the spatial assumption A4.3 but that is not restricted to neighbors on the same rigid object, but rather to every pixel neighbors.

Assumption 4.6 (Horn and Schunk (HS))**.** All neighbors move similar.

The corresponding energy function is

$$E = E_{BCE} + E_{HS} = \sum_{\boldsymbol{x}} \|\mathcal{I}_x(\boldsymbol{x})\mathcal{V}_x(\boldsymbol{x}) + \mathcal{I}_y(\boldsymbol{x})\mathcal{V}_y(\boldsymbol{x}) + \mathcal{I}_t(\boldsymbol{x})\|_2^2 \tag{4.9}$$

$$+ \lambda_{hs}(\|\mathcal{V}_x(\boldsymbol{x}) - \mathcal{V}_x(\boldsymbol{x}-\mathbf{1})\|_2^2 + \|\mathcal{V}_y(\boldsymbol{x}) - \mathcal{V}_y(\boldsymbol{x}-\mathbf{1})\|_2^2). \tag{4.10}$$

The additional energy function E_{HS} is characteristic for all OFE-algorithms based on the HS assumption. It has two interesting properties. First, the optical flow gets smoothed, which leads to smooth motion borders. The second property is, that the HS energy function finds solutions for unstructured areas and thus gives fully dense optical flow fields. In

homogenous areas the BCE gives no information about the optical flow, because the spatial derivations are zero, only the E_{HS} energy function is active which results in highly regularized OF-fields. This can be fairly reasonable in some cases but does not guarantee correct estimations since the assumption A4.6 is of course not valid for all cases, because it completely discards any information about the local object segmentation.

4.2.2 Lukas and Kanade

Another method for OFE was proposed by Lukas and Kanade [71] which introduced a local method in contrast to the global HS approach. Here the idea is that the BCE is not solved for each pixel independently, but that a predefined area $A(\boldsymbol{x})$ around each pixel $\boldsymbol{x}$ has an identical motion vector $\mathcal{V}(\boldsymbol{x})$.

Assumption 4.7 (Lukas and Kanade (LK))**.** All pixels in a predefined neighborhood $A(\boldsymbol{x})$ move similarly.

So instead of the under constrained problem of a single equation and the two unknowns $\mathcal{V}_x(\boldsymbol{x})$ and $\mathcal{V}_y(\boldsymbol{x})$ for each pixel $\boldsymbol{x}$ there is an over-constrained problem with a linear system of equations with the size of the pixels in $A(\boldsymbol{x})$ and the two unknowns $\mathcal{V}_x(\boldsymbol{x})$ and $\mathcal{V}_y(\boldsymbol{x})$. The corresponding energy function is

$$E = \sum_{\boldsymbol{x}} \sum_{\boldsymbol{y} \in A(\boldsymbol{x})} \|\mathcal{I}_x(\boldsymbol{y})\mathcal{V}_x(\boldsymbol{x}) + \mathcal{I}_y(\boldsymbol{y})\mathcal{V}_y(\boldsymbol{x}) + \mathcal{I}_t(\boldsymbol{y})\|_2^2. \tag{4.11}$$

Eq. (4.11) can be solved directly with least squares solvers. Unlike the HS approach the LK algorithm is a local method. On the one hand the LK approach gives no reliable estimates in unstructured areas and is more prone to the aperture problem. On the other hand it is more robust in structured areas and does not tend to over-regularize as compared to the HS approach.

4.2.3 Extensions of the Classical Methods

There are several extensions of the two *classical methods* mentioned above. *E.g.* Bruhn et al. [11] combine the local LK and global HS approach to get the density and regularization properties of the HS energy functional eq. (4.10) and the more robust local LK eq. (4.11). The conceptual problems related to the assumptions A4.6 and A4.7 are not considered.

Another typical extension is the use of more complex models than the translational motion model with the two parameters $\mathcal{V}_x(\boldsymbol{x})$ and $\mathcal{V}_y(\boldsymbol{x})$. *E.g.* affine motion models with its six free parameters can better describe local optical flow fields than translational models [16, 44]. While affine and other polynomial methods are better suited to model the local neighborhoods they still are spatially smooth and cannot model motion boundaries.

4.2.4 Multi-Scale Methods

A major problem for the application of OFE algorithms is that condition 4.1, thus the restriction to small displacements between two consecutive frames, is often violated. Consequently the BCA A4.1 and the linearization assumption A4.5 are not valid, which leads to erroneous estimations.

To overcome this problem, *multi-scale* methods [3, 11, 98] are commonly applied together with the differential approach. The underlying idea is that larger displacements correspond to small displacements on lower image scales. By reducing the image resolution and hence the level of detail in the image, the BCA 4.1 and the linearization assumption A4.5 become valid again. The optical flow results for this low-scale images are projected onto the underlying higher resolution image where the detailed optical flow is estimated. This process is continued throughout all scales until the original high resolution image is reached.

4.2.5 Other OFE-algorithms

There exist various other OFE-algorithms. Interesting examples are methods that make use of *learned models* to describe the local neighborhood. The algorithm proposed in [31] applies local optical flow patterns that are learned with PCA. A similar method proposed in [75] uses global PCA patterns, while in [57] local patterns are learned with a sparse coding algorithm. All these methods have in common, that they need optical flow fields to learn their patterns. The local method proposed in [31] uses the learned model to directly describe the OF-field, so the optimization problem becomes a model parameter estimation problem. In [57, 75] the optical flow has to fit a model defined by the prelearned patterns. The reconstruction error, thus the error between the estimated OF-field and the model build with the prelearned model parameters, is an additional constraint incorporated into the optimization as an additional energy function.

Besides the learning based algorithms, there exist OF-methods that combine *segmentation* and optical flow estimation [98, 99] or that exploit *temporal relations* [112, 113].

4.3 VNMF-OFE Approach

The concept of parts-based models is now used to find locally bound spatial relations based on the image statistics. The main idea is that

Assumption 4.8. All natural occurring optical flow fields can be described by a sparse, non-negative, linear superposition of local, shift-invariant patterns. Each pattern describes a local neighborhood with a consistent flow field.

The set of basic patterns is learned by directly solving the BCE with the model assumption using multiplicative gradient descent. Unlike related approaches the OF-field is directly restricted to be the model rather than adding a constraint that the OF-field is similar to a model. The additional constraints influence the nature of the model, favoring parts-based, segmentation like representations, because they best represent the physical constraints formulated in the spatial assumption A4.3. The main additional constraint on the model is non-negativity for all components, including the activations that describe the motion direction of the learned local neighborhoods. Non-negativity is achieved by the direction selective representation introduced in section 3.4. In addition to the non-negativity, sparsity on the amplitudes of the activations, orthogonality between the opposing directions and lateral competition are enforced by additional energy functionals. Once the basis vectors that describe the local patterns are learned, the OFE-problem is reduced to a parameter estimation problem, similar to [31]. The approach is now introduced in a formal manner.

4.3.1 Restrict Optical Flow Field to Model

The basis for the VNMF-OFE algorithm is the linearized BCE 4.7 with an euclidean penalty function

$$E_{BCE} = \frac{1}{2} \sum_{\boldsymbol{x}} \Big(\mathcal{I}_x(\boldsymbol{x}) \mathcal{V}_x(\boldsymbol{x}) + \mathcal{I}_y(\boldsymbol{x}) \mathcal{V}_y(\boldsymbol{x}) + \mathcal{I}_t(\boldsymbol{x}) \Big)^2 . \tag{4.12}$$

The optical flow components $\mathcal{V}_x$ and $\mathcal{V}_y$ are now restricted to global models

$$\mathcal{V}_x = \mathcal{R}_x(\mathcal{H}_x, \bar{\mathcal{W}}), \tag{4.13}$$

$$\mathcal{V}_y = \mathcal{R}_y(\mathcal{H}_y, \bar{\mathcal{W}}), \tag{4.14}$$

that depend on the basis vector set $\bar{\mathcal{W}}$ and the corresponding direction specific activation sets $\mathcal{H}_x$ and $\mathcal{H}_y$. Eq. (4.14) inserted in the energy function eq. (4.12) gives the new BCE

$$E_{BCE} = \frac{1}{2} \sum_{\boldsymbol{x}} \Big(\mathcal{I}_x(\boldsymbol{x}) \mathcal{R}_x(\boldsymbol{x}) + \mathcal{I}_y(\boldsymbol{x}) \mathcal{R}_y(\boldsymbol{x}) + \mathcal{I}_t(\boldsymbol{x}) \Big)^2. \tag{4.15}$$

The model itself is a linear superposition of shift invariant basis vectors as described in section 3.5

$$\mathcal{R}_x(\boldsymbol{x}) = \sum_{j,\boldsymbol{m}} h_{jx}(\boldsymbol{m}) \bar{w}_j(\boldsymbol{x} - \boldsymbol{m}), \tag{4.16}$$

$$\mathcal{R}_y(\boldsymbol{x}) = \sum_{j,\boldsymbol{m}} h_{jy}(\boldsymbol{m}) \bar{w}_j(\boldsymbol{x} - \boldsymbol{m}), \tag{4.17}$$

where both directions x and y share the common basis vectors $\bar{\mathcal{W}}_j$, but have individual activations $\mathcal{H}_{jx}$ and $\mathcal{H}_{jy}$.

4.3.2 Enforced Non-Negativity

To achieve a parts-based composition and to be able to apply the NMF update rules introduced in chapter 3 all components in the BCE (4.15) are forced to be strictly non-negative. Since the optical flow fields can contain positive and negative values, the model has to be split up as discussed in the algorithmic section 3.4.3. The basis vectors should describe the local neighborhoods, so the additional dimensions that encode the different directions are represented in the activations of the model. The condition for the non-negative basis vectors is

$$\mathcal{W} \geq 0. \tag{4.18}$$

The models which must be able to represent positive and negative values for the different movement directions become non-negative by splitting each component into its positive and negative representation, thus

$$\begin{aligned} \mathcal{R}_x &= \mathcal{R}_{x+} - \mathcal{R}_{x-}, \\ \mathcal{R}_y &= \mathcal{R}_{y+} - \mathcal{R}_{y-}, \\ \mathcal{R}_{ds} &\geq 0, \quad d \in \{x, y\}, s \in \{+, -\}. \end{aligned} \tag{4.19}$$

For the four directions *right, left, up, down* the corresponding activations are

$$\mathcal{H}_{jds} \geq 0, \quad d \in \{x, y\}, s \in \{+, -\}. \tag{4.20}$$

To make every component in the BCE (4.15) non-negative it is not sufficient to apply these conditions to the model components alone, but to split up the gradients $\mathcal{I}_x$, $\mathcal{I}_y$ and $\mathcal{I}_t$ as well.

$$\begin{aligned} \mathcal{I}_x &= \mathcal{I}_{x+} - \mathcal{I}_{x-}, \\ \mathcal{I}_y &= \mathcal{I}_{y+} - \mathcal{I}_{y-}, \\ \mathcal{I}_t &= \mathcal{I}_{t+} - \mathcal{I}_{t-}, \\ \mathcal{I}_{gs} &\geq 0, \quad g \in \{x, y, t\}, s \in \{+, -\}. \end{aligned} \tag{4.21}$$

The non-negative representation of the BCE (4.15) is

$$\begin{aligned} E_{BCE} = & \frac{1}{2} \sum_{\boldsymbol{x}} \Big((\mathcal{I}_{x+}(\boldsymbol{x}) - \mathcal{I}_{x-}(\boldsymbol{x}))(\mathcal{R}_{x+}(\boldsymbol{x}) - \mathcal{R}_{x-}(\boldsymbol{x})) \\ & + (\mathcal{I}_{y+}(\boldsymbol{x}) - \mathcal{I}_{y-}(\boldsymbol{x}))(\mathcal{R}_{y+}(\boldsymbol{x}) - \mathcal{R}_{y-}(\boldsymbol{x})) \\ & + (\mathcal{I}_{t+}(\boldsymbol{x}) - \mathcal{I}_{t-}(\boldsymbol{x})) \Big)^2. \end{aligned} \tag{4.22}$$

Analyzing the non-negative BCE (4.22) reveals that even though the representation, thus all the components fulfill the non-negative constraints, the model itself allows overlaps between the non-negative components that represent the positive and negative parts. It is thus a form of the *weak non-negativity* as discussed in section 3.4.7. As a consequence subtractions in the reconstruction are possible which is contrary to the idea of a purely additive model, *i.e.* a parts-based decomposition. To privilege but not restrict the model to purely additive behavior additional constraints are added via additional energy functionals.

4.3.3 Penalize Opposing Directions

In section 3.4.8 a penalty function for opposing directions is introduced. The same energy function is now applied to penalize overlaps of opposing directions s and $\bar{s}$ ($\bar{s} = -$, if $s = +$ and $\bar{s} = +$, if $s = -$). The energy function is

$$E_R = \frac{1}{2} \sum_{d,\boldsymbol{x}} \mathcal{R}_{ds}(\boldsymbol{x}) \mathcal{R}_{d\bar{s}}(\boldsymbol{x}). \tag{4.23}$$

Throughout the experiments the E_R has the same weighting factor as the E_{BCE} term. Like the brightness constancy energy E_{BCE}, the energy function E_R is directly depending on the reconstruction and therefore both energy functions scale accordingly, so that adaptation of the weighting factor is not necessary. As a consequence, the term for the penalty of the opposing direction is parameter free.

4.3.4 Sparse Activity Amplitudes

To make the activations of the model sparse the energy cost function is extended by an additional sparsity energy term. As discussed in the section on multidimensional activations, section 3.4.3, the classical sparsity energy contribution

$$E_h = \sum_{j,\boldsymbol{m},d,s} \|h_{jds}(\boldsymbol{m})\|_1 \tag{4.24}$$

is not directly applicable, because each of the four directions is penalized independently of the others. This favors movements that have a high amplitude towards one of the four directions and would discard small angular variations towards any of the two neighboring directions. Instead of eq. 4.24 the energy function proposed in section 3.4.4, that penalizes the amplitude of the activity vector, rather then the single directions is applied. The energy function for the sparsity of the activation amplitudes is

$$E_H = \sum_{j,\boldsymbol{m},d,s} \frac{\mathcal{H}_{jds}(\boldsymbol{m})}{\sqrt{\sum_{d',s'} \mathcal{H}_{jd's'}(\boldsymbol{m})^2}}. \tag{4.25}$$

In the case of one dimensional activations the two equations (4.24) and (4.25) are identical.

4.3.5 Lateral Competition

As discussed in section 3.5.4 a topological sparse representation for the translation-invariant model can be favored by a penalty on the overlaps of the partial reconstructions. In contrast to penalty of the opposing directions in eq. (4.23), the penalty on the partial reconstructions is performed for each of the four directions independently. The energy function is

$$E_P = \frac{1}{2} \sum_{\boldsymbol{x},d,s} \sum_{j,\boldsymbol{m}} \mathcal{R}_{dsj\boldsymbol{m}}(\boldsymbol{x}) \Big(\sum_{k,\boldsymbol{n}} \mathcal{R}_{dsk\boldsymbol{n}}(\boldsymbol{x}) - \mathcal{R}_{dsj\boldsymbol{m}}(\boldsymbol{x}) \Big). \tag{4.26}$$

4.3.6 VNMF-OFE Learning Algorithm

The overall energy function consist of four parts. The non-negative BCE (4.22), the penalty function for the opposing directions in eq. (4.23), eq. (4.25) for sparse activity amplitudes and the lateral competition term in eq. (4.26) for topological sparseness and an enforced parts-based decomposition. The combination of the four energy functions allows the learning of the local neighborhoods with translational motion for which the spatial relation assumption A4.3 is valid. The overall energy function is

$$
\begin{aligned}
E_{\text{VNMF-OFE}} &= E_{BCE} + E_R + \lambda_{\text{h}} E_H + \lambda_{\text{part}} E_P \qquad (4.27)\\
&= \frac{1}{2} \sum_{\boldsymbol{x}} \Big((\mathcal{I}_{x+}(\boldsymbol{x}) - \mathcal{I}_{x-}(\boldsymbol{x}))(\mathcal{R}_{x+}(\boldsymbol{x}) - \mathcal{R}_{x-}(\boldsymbol{x}))\\
&\quad + (\mathcal{I}_{y+}(\boldsymbol{x}) - \mathcal{I}_{y-}(\boldsymbol{x}))(\mathcal{R}_{y+}(\boldsymbol{x}) - \mathcal{R}_{y-}(\boldsymbol{x}))\\
&\quad + (\mathcal{I}_{t+}(\boldsymbol{x}) - \mathcal{I}_{t-}(\boldsymbol{x})) \Big)^2\\
&\quad + \frac{1}{2} \sum_{d,\boldsymbol{x}} \mathcal{R}_{ds}(\boldsymbol{x}) \mathcal{R}_{d\bar{s}}(\boldsymbol{x})\\
&\quad + \lambda_{\text{h}} \sum_{j,\boldsymbol{m},d,s} \frac{h_{jds}(\boldsymbol{m})}{\sqrt{\sum_{d',s'} h_{jd's'}(\boldsymbol{m})^2}}\\
&\quad + \lambda_{\text{part}} \frac{1}{2} \sum_{\boldsymbol{x},d,s} \sum_{j,\boldsymbol{m}} \mathcal{R}_{dsj\boldsymbol{m}}(\boldsymbol{x}) \Big(\sum_{k,\boldsymbol{n}} \mathcal{R}_{dsk\boldsymbol{n}}(\boldsymbol{x}) - \mathcal{R}_{dsj\boldsymbol{m}}(\boldsymbol{x}) \Big).
\end{aligned}
$$

To estimate the optical flow with eq. (4.28) the unknown basis vector set $\mathcal{W}$ has to be learned and the activations $\mathcal{H}$ have to be detected. While the basis vectors describe the local neighborhoods and only have to be learned once, the activations are the parameters required to get the actual optical flow field. A key difference to most learning based approaches in the literature is that for both tasks the same energy function eq. (4.28) and the same algorithm is used. Both learning algorithms consists of two parts, the initialization and the updates, which are performed in an iterative process. The algorithm for learning the basis vector set $\mathcal{W}$ is:

- Preprocessing
  - Normalize $\mathcal{I} = \frac{\mathcal{I}}{\max(\mathcal{I})}$,
  - calculate the image gradients $\mathcal{I}_{x+}, \mathcal{I}_{x-}, \mathcal{I}_{y+}, \mathcal{I}_{y-}, \mathcal{I}_{t+}, \mathcal{I}_{t-}$,
  - initialize $\mathcal{H}$ and $\mathcal{W}$ randomly.

- Loop for i iterations
  1. Loop for each of the N inputs
     a) Calculate $\mathcal{R}_{nd} = \sum_j \text{conv}_2(\mathcal{H}_{jnd}, \bar{\mathcal{W}}_j)$,
     b) update $\mathcal{H}_{jnd} \rightarrow \mathcal{H}_{jnd} \circ \frac{(\nabla_{\mathcal{H}_{jnd}} E_{\text{VNMF-OFE}})^-}{(\nabla_{\mathcal{H}_{jnd}} E_{\text{VNMF-OFE}})^+}$,
     c) calculate $\mathcal{R}_{nd} = \sum_j \text{conv}_2(\mathcal{H}_{jnd}, \bar{\mathcal{W}}_{jf})$,
     d) calculate $(\nabla_{\mathcal{W}} E_{\text{VNMF-OFE}})_n^+$ and $(\nabla_{\mathcal{W}} E_{\text{VNMF-OFE}})_n^-$
  2. update $\mathcal{W} \rightarrow \mathcal{W} \circ \frac{(\nabla_{\mathcal{W}} E_{\text{VNMF-OFE}})^-}{(\nabla_{\mathcal{W}} E_{\text{VNMF-OFE}})^+}$, with
     $(\nabla_{\mathcal{W}} E_{\text{VNMF-OFE}})^+ = \sum_n (\nabla_{\mathcal{W}} E_{\text{VNMF-OFE}})_n^+$ and
     $(\nabla_{\mathcal{W}} E_{\text{VNMF-OFE}})^- = \sum_n (\nabla_{\mathcal{W}} E_{\text{VNMF-OFE}})_n^-$,
  3. normalize $\bar{\mathcal{W}}_j = \frac{\mathcal{W}_j}{\sqrt{\sum_p w_{pj}^2}}, \quad \forall j \in [1, ..., J]$.

The gradients for the basis vectors and activations are:

$$\left(\nabla_{\mathcal{H}_{jnd}} E_{\text{VNMF-OFE}}\right)^+ = \text{corr}_2\left((\mathcal{I}_{nd} \circ \boldsymbol{A} + \mathcal{I}_{n\hat{d}} \circ \boldsymbol{B}), \bar{\mathcal{W}}_j\right), \tag{4.28}$$

$$\left(\nabla_{\mathcal{H}_{jnd}} E_{\text{VNMF-OFE}}\right)^- = \text{corr}_2\left((\mathcal{I}_{nd} \circ \boldsymbol{B} + \mathcal{I}_{n\hat{d}} \circ \boldsymbol{A}), \bar{\mathcal{W}}_j\right), \tag{4.29}$$

with

$$\boldsymbol{A} = \mathcal{I}_{nt+} + \sum_d \mathcal{I}_{nd} \circ \mathcal{R}_{nd}, \tag{4.30}$$

$$\boldsymbol{B} = \mathcal{I}_{nt-} + \sum_d \mathcal{I}_{nd} \circ \mathcal{R}_{n\hat{d}}, \tag{4.31}$$

where $\hat{d}$ describes the opposing direction to d (*e.g.* $\hat{d} = x-, d = x+$). The gradient for the sparsity energy function is

$$\left(\nabla_{\mathcal{H}_{jnd}} E_H\right)^+ = \frac{\mathcal{H}_{jnd}}{\|\boldsymbol{h}_{jn}(\boldsymbol{m})\|_2}. \tag{4.32}$$

The gradients for the penalization of the opposing directions are

$$\left(\nabla_{\mathcal{H}_{jnd}} E_R\right)^+ = \text{corr}_2\left(\mathcal{R}_{nd}, \bar{\mathcal{W}}_j\right) \tag{4.33}$$

Figure 4.3: Six example images of a *smiling* sequence of the MMI dataset [104].

and

$$\left(\nabla_{\mathcal{W}_j} E_R\right)^+ = \sum_d \operatorname{corr}_2\left(\mathcal{R}_{n\hat{d}}, \mathcal{H}_{jnd}\right). \tag{4.34}$$

The two parameters λ_{h} and λ_{part} control the influence of the sparsity and parts-based terms. They both scale relative to E_{BCE} and are therefore easy to parametrize as will be discussed throughout the experiments.

4.3.7 VNMF-OFE Algorithm for Activation Inference

Once the basis vectors are learned only the activations have to be calculated to estimate the optical flow. The algorithm for estimating the optical flow with prelearned basis vectors is:

- Preprocessing
  - Normalize $\mathcal{I} = \frac{\mathcal{I}}{\max(\mathcal{I})}$,
  - calculate the image gradients $\mathcal{I}_{x+}, \mathcal{I}_{x-}, \mathcal{I}_{y+}, \mathcal{I}_{y-}, \mathcal{I}_{t+}, \mathcal{I}_{t-}$,
  - initialize $\mathcal{H}$ randomly.
- Loop for N iterations
  1. Calculate $\mathcal{R}_{nd} = \sum_j \operatorname{conv}_2(\mathcal{H}_{jnd}, \bar{\mathcal{W}}_j)$,
  2. update $\mathcal{H}_{jnd} \to \mathcal{H}_{jnd} \circ \frac{(\nabla_{\mathcal{H}_{jnd}} E_{\text{VNMF-OFE}})^-}{(\nabla_{\mathcal{H}_{jnd}} E_{\text{VNMF-OFE}})^+}$,

The number of iterations for the optical flow estimation is set to $N = 30$. The gradients are identical to the ones used for learning the basis vectors.

4.4 Learning the Basis Vectors

In the following the parameter variations for learning the basis vectors $\mathcal{W}$, thus the local receptive fields with consistent motion, are discussed. The experiments are performed on a video of a smiling person from the MMI face expression recognition dataset [104] as depicted in fig. 4.3. Since the focus of the OFE in this thesis is on finding suitable features for biological motion recognition, the evaluation of the optical flow is not performed on typically used benchmarks for OFE. The evaluation is instead focused on how the algorithm is capable of preserving small details of the performed movements, *e.g.* subtle movements of the lip during smiling.

A comparison for the full combination of all possible parameters is computationally too extensive, so a baseline set of parameters is defined and then single parameters are varied to analyze their influence on the extracted patterns and the resulting optical flow field. The baseline parameters are

$$J = 8, \qquad \lambda_{\mathrm{h}} = 0.001,$$
$$mRFS = 8 \times 8, \qquad \lambda_{\mathrm{part}} = 0.$$

The number of iterations for basis vector learning is set to $N = 150$ for all experiments.

4.4.1 Varying Model Parameters

Fig. 4.4 shows six different learned basis vector sets for varying model parameters J and $mRFS$.

For the smallest receptive field size ($mRFS = 4 \times 4$) and for a low amount of basis vectors ($J = 1$) no interpretable structures emerge in the basis vectors. For a larger number of basis vectors bar-like structures are learned, but the overall appearance of the basis vectors is not very specific.

The brightness constancy energy function over the 100 iterations for the different model parameters is displayed in fig. 4.5. The learning algorithm that uses the simple multiplicative update rules is converging in all cases. In fact, in all performed experiments the algorithms always converged, which is surprising, because the proof of convergence is only given for the original reconstruction energy function used by Lee and Seung [26].

Increasing the number of basis vectors to any value $J > 1$ decreases the energy function, because more details can be explicitly represented by the basis vectors. However, this effect saturates for $J > 4$ while the computational time is increasing linearly with J. The different $mRFS$ have no significant impact on the BCE function.

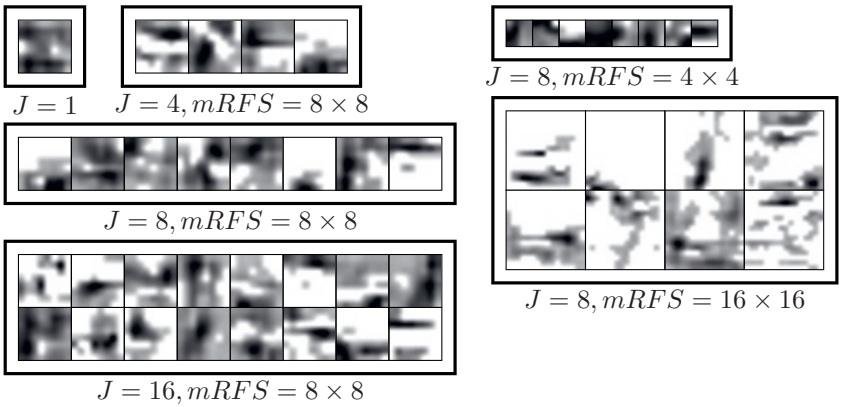

Figure 4.4: Six basis vector sets for varying model parameters J and $mRFS$ learned on the face sequence depicted in fig. 4.3. The black parts indicate the areas with coherent motion.

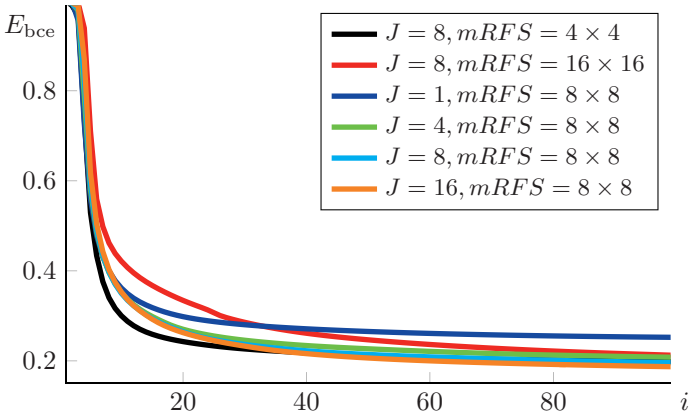

Figure 4.5: Normalized brightness constancy energy function for six different parameter configurations over 100 iterations.

4.4.2 Varying Energy Parameters

The learned basis vector sets for a variation of the two energy parameters λ_{h} and λ_{part} are shown in fig. 4.6. For the first parameter set (set1:

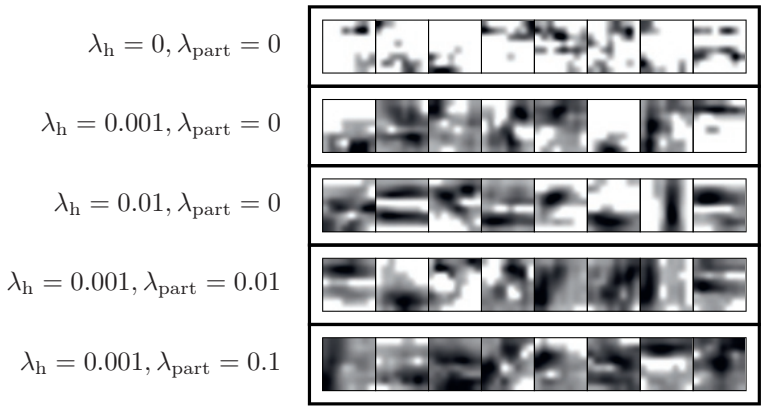

Figure 4.6: Five basis vector sets for varying energy parameters λ_h and λ_part learned on the face sequence depicted in fig. 4.3.

$\lambda_\mathrm{h} = 0 \wedge \lambda_\mathrm{part} = 0$), the basis vectors are very sparse and only represent a small local neighborhood. Due to the limited spatial structure, the learned basis vectors only describe the parts of the image with a strong spatial gradient and do not regularize well. Increasing the sparsity parameter (set2: $\lambda_\mathrm{h} = 0.001 \wedge \lambda_\mathrm{part} = 0$), penalizes the use of multiple activations, which directly leads to spatially extended basis vectors. The basis vectors start to represent local image structures. This effect can be further intensified by increasing the sparsity parameter (set3: $\lambda_\mathrm{h} = 0.01 \wedge \lambda_\mathrm{part} = 0$) or by using the parts-based energy function (set4: $\lambda_\mathrm{h} = 0.001 \wedge \lambda_\mathrm{part} = 0.01$). For a higher value of the weighting parameter of the parts-based energy function (set5: $\lambda_\mathrm{h} = 0.001 \wedge \lambda_\mathrm{part} = 0.1$), the basis vectors tend to represent holistic structures.

The effect of the parts-based term E_part can be better understood if the energy functions are analyzed. In fig. 4.7 the BCE energy function and the parts-based energy function E_part are shown for the five parameter sets. The basis vectors learned without a penalization of the activations (blue, set1: $\lambda_\mathrm{h} = 0 \wedge \lambda_\mathrm{part} = 0$) have the smallest error for the BCE, because the spatially sparse basis vectors are trivial solutions that can represent any given model. They neglect the spatial relations of assumption A4.3 that are necessary to solve the ill-constraint OFE-problem. In addition, the trivial solution has the highest overlap between the partial reconstructions, which is measured by the parts-based energy E_part. Increasing the sparsity

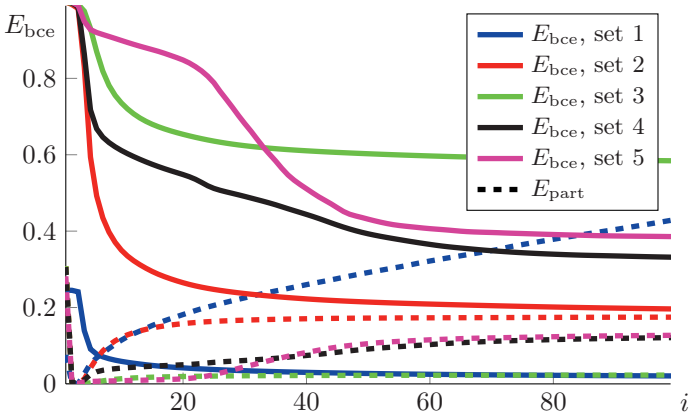

Figure 4.7: Normalized brightness constancy (solid lines) and parts-based (dashed lines) energy function for five different parameter sets(different colors) over 100 iterations.

parameter increases the BCE as well (red, set2: $\lambda_{\text{h}} = 0.001 \wedge \lambda_{\text{part}} = 0$), while the overlap is reduced. The highest sparsity parameter (green, set3: $\lambda_{\text{h}} = 0.01 \wedge \lambda_{\text{part}} = 0$) has the worst BCE and the lowest parts-based energy. Instead of increasing the sparsity parameter, a weakened parts-based energy can be achieved by using the parts-based parameter (black, set4: $\lambda_{\text{h}} = 0.001 \wedge \lambda_{\text{part}} = 0.01$). The benefit of the λ_{part} parameter is that it directly addresses the overlap. A further increase of the parts-based parameter (pink, set5: $\lambda_{\text{h}} = 0.001 \wedge \lambda_{\text{part}} = 0.1$) does not result in a more parts-based decomposition and only worsens BCE.

The energy parameters directly influence the activations and the optical flow as visualized in fig. 4.8. An increasing sparsity parameter λ_{h} leads to an increased sparseness of the activations and thus to less dense optical flow fields as well. A similar effect is given by an increasing parts-based parameter λ_{part}. Here the activations are topological sparse and the local reconstructions are limited to single basis vectors. As a result, the corresponding basis vectors get more specific and are able to represent more details, as it can be seen in the zoomed optical flow in fig. 4.8. However, the restriction makes the model less flexible with the direct consequence, that the optical flow field is less dense and that the BCE error measure defined by the energy function is higher than without the E_{part} term.

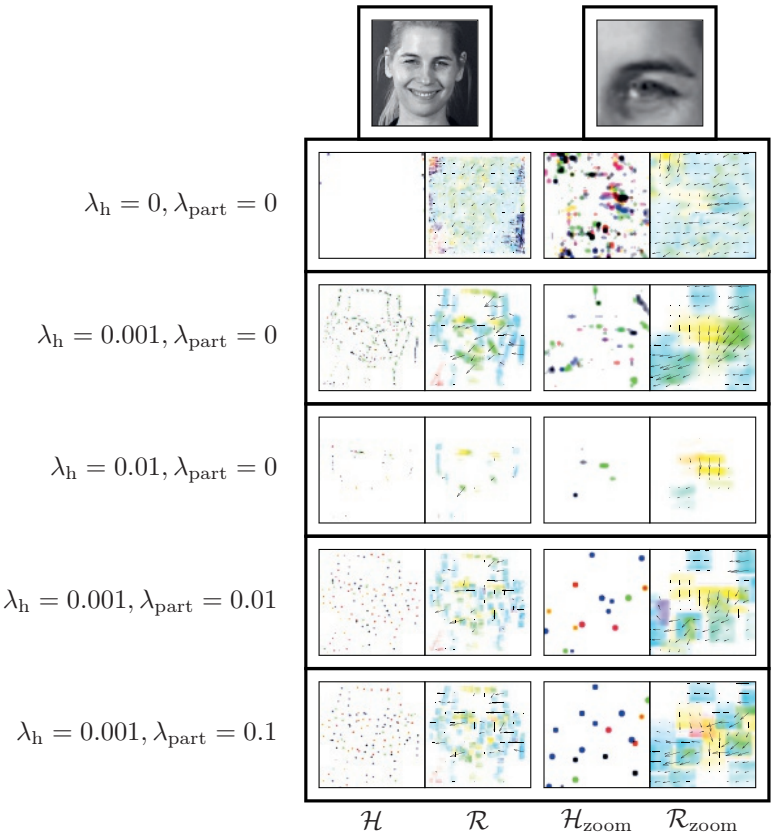

Figure 4.8: The activations $\mathcal{H}$ and optical flow fields $\mathcal{R}$ of the smiling sequence for different energy parameters are shown for the entire image and for the zoomed in eye area. The different colors for the activations indicate the use of different basis vectors. The color code for the optical flow fields is taken from [98] and works as follows: Each color indicates a direction (red = right, purple = up, blue = left, yellow = down) while the velocity is encoded in the saturation.

4.4.3 Learned vs Designed Basis Vectors

To further show the benefits of learning the basis vectors, the learned basis vectors are compared to a set of designed basis vectors. Two different sets of designed basis vectors are chosen: First, the trivial case of one Gaussian

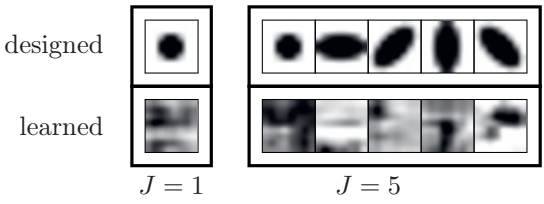

Figure 4.9: Upper row: Two sets of designed basis vectors. Lower row: Two sets of learned basis vectors with the same J.

basis vector and second, five designed basis vectors, including the Gaussian basis vector and four rotated ellipsoidal basis vectors. For comparison two learned basis vector sets with the same amount of basis vectors as the designed sets are visualized together with the designed sets in fig. 4.9.

The single learned basis vector differs strongly from the designed Gaussian basis vector. For increasing number of basis vectors different structures emerge. In fig. 4.10 example activations along with the resulting optical flows are shown. The learned basis vectors adapt to the structure present in the images and are therefore more detail preserving than the designed basis vectors. However, the differences between the flow fields are rather insignificant.

Learning the basis vectors leads to a slightly lower BCE function compared to using the designed basis vectors as depicted in fig. 4.11. However, it is noteworthy that using designed basis vectors also provides reasonably good results. Most details are preserved and the algorithm still converges. Similar to the learned basis vectors, the designed basis vectors benefit from the robust optimization due to the non-negativity, sparsity and the global translation-invariant model.

4.4.4 Discussion of the Parameter Settings

To ensure a robust estimation with few misestimations the sparsity parameter should be set in a range of $\lambda_{\mathrm{h}} \in \{0.0001, 0.01\}$. An increased sparsity parameter results in a less dense optical flow field and the estimations focus on the most prominent movements. The sparsity parameter is thus a tool to regulate the robustness and the density of the optical flow estimation. The parts based parameter λ_{part} leads to more specific basis vectors and optical flow estimations. On the one hand, this allows the algorithm to preserve

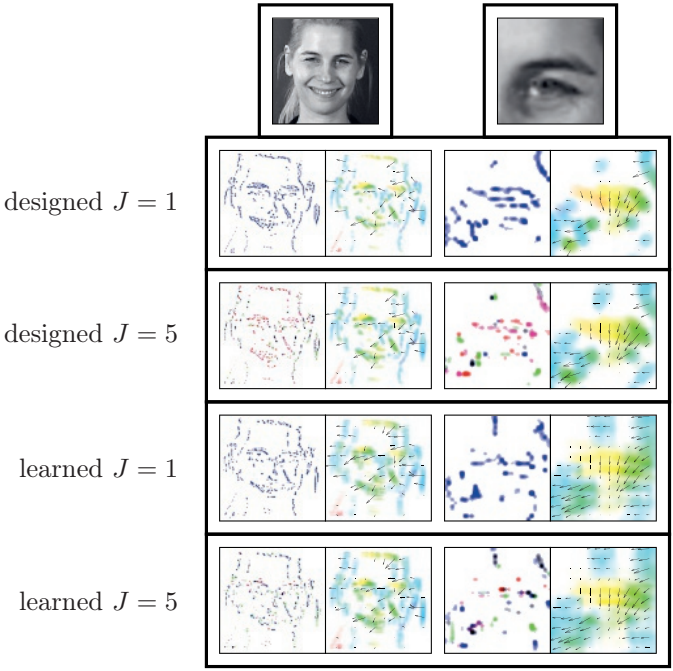

Figure 4.10: The activations $\mathcal{H}$ and optical flow fields $\mathcal{R}$ of the smiling sequence for the learned and designed basis vector sets are shown for the entire image and for the zoomed in eye area. The different colors for the activations indicate the use of different basis vectors.

fine detailed movements, like the motion of the eye lids. On the other hand, the basis vectors get less generic and a large amount of basis vectors J is required to preserve multiple details. The computational time is depending linear on the number of basis vectors, thus preserving movement details will increase the computational time.

In summary, setting $\lambda_{\mathrm{h}} = 0.001$, $\lambda_{\mathrm{part}} = 0$ and $J = 8$ will lead to a fast, generic and robust optical flow estimation algorithm. If a finer optical flow is required the parameters should be set to $\lambda_{\mathrm{part}} = 0.01$ and J must be increased. Alternatively, the five designed patterns shown in fig. 4.9 give reasonable results over different datasets. All following optical flow fields are estimated with the five designed basis vectors along with the parameters $\lambda_{\mathrm{h}} = 0.001$ and $\lambda_{\mathrm{part}} = 0$.

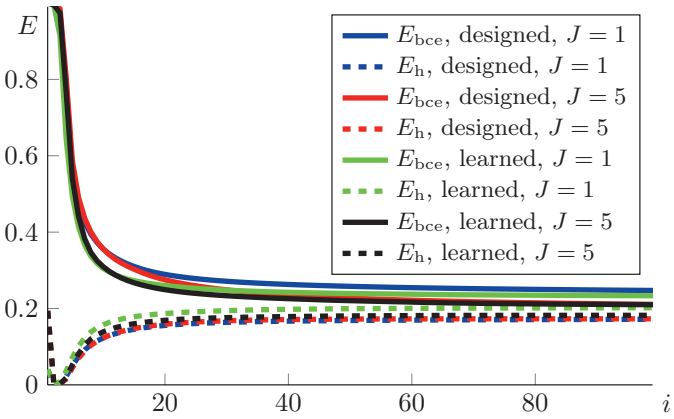

Figure 4.11: Normalized brightness constancy (solid lines) and sparsity (dashed lines) energy function for the designed and learned basis vector sets (different colors) over 100 iterations.

4.5 Comparison & Results

The two main characteristics of the VNMF-OFE algorithm are: First, the learned detail preserving local receptive fields, represented by the basis vectors. And second, the robust optimization due to one global model that allows for interactions between the local receptive fields and the non-negativity and sparsity constraints that eliminate miss estimations. The first point has already been discussed in section 4.4. Next, the focus is on the robustness of the proposed VNMF-OFE algorithm. The estimated optical flow is compared to other algorithms on a face expression recognition dataset. In addition, the optical flow of different full human body movements is estimated. The videos contain cluttered background and camera motion. Throughout the experiments in the following chapters the five designed basis vectors are chosen[2)], along with the parameters $\lambda_{\text{h}} = 0.001$ and $\lambda_{\text{part}} = 0$.

[2)] The designed basis vectors are choosen to make it easier to reconstruct the results in the subsequent chapters.

Table 4.1: Subset of possible AU and their related emotions, according to [29].

Emotion	AU	Description
Fear, Sadness, Surprise	1	Inner brow raise
Fear, Surprise	2	Outer brow raise
Anger, Sadness,	4	Brow lower
Happiness	6	Cheek raise
Anger	7	Lid tightened
Disgust	9	Nose wrinkle
Happiness	12	Lip corner puller
Disgust, Sadness	15	Lip corner depressor
Disgust	16	Lower lip depressor
Fear, Surprise	26	Jaw drop

4.5.1 Comparison to Related Work

The optical flow estimated with the VNMF-OFE algorithm is now compared to the two classic algorithms from Lukas and Kanade (LK) and Horn and Schunk (HS) as well as to a state-of-the-art algorithm from Sun et al. (Sun) [98]. The results of this section have already been reported in [48]. The dataset used for the comparison is a face expression recognition dataset [104].

According to the psychologist Eckman, there exist six cultural universal emotional facial expressions: *anger*, *disgust*, *fear*, *happiness*, *sadness* and *surprise* [29]. Each facial expression has characteristic movements, so called *action units* (AU), of different face parts, such as the inner or the outer eyebrow, the corners of the mouth, etc. A subset of the set of action units and their relation to each of the six emotional states is given in table 4.1. Examples of this face expressions and the corresponding characteristic movements are depicted in fig. 4.12. Also shown is the estimated optical flow for each of the four algorithms.

The focus of the comparison is on whether the AU movement is preserved, thus, if the movement can be locally assigned to a facial part. The HS OFE can capture most of the movements, but looses all local shape relations due to its strong dependency on the spatial regularization term. The LK OFE can preserve most of the local relations, but has a lot of miss-estimations

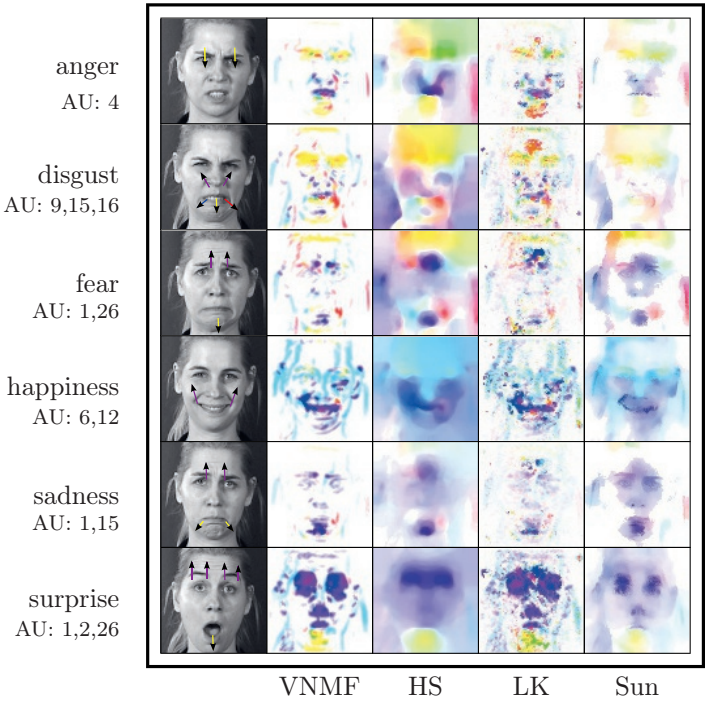

Figure 4.12: Method comparison: The first column presents the first image of the apex phase (emotion fully expressed) for each emotion. In addition some of the emotion-related AU are drawn onto the image (arrows). The other columns show the color-coded integrated flow fields for the four different algorithms, *i.e.* the integrated flow fields of each consecutive image pair of the onset phase (face movement from neutral to apex).

and is the least robust method.[3)] The highly sophisticated OFE by Sun et al. [98] is capable of preserving an impressive amount of detail due to its local segmentation. Unfortunately similar to the HS algorithm it tends to over-regularize in low contrast areas as *e.g.* the cheeks and thus looses most of the local relations.

[3)] In the experiments outliers are discarded and a bias to zero movement was inserted by adding a constant noise image to all images of a sequence.

In contrary to the above mentioned methods, the VNMF-OFE algorithm is capable of extracting the main AU related movements and keeps the local shape relations robustly. *E.g.* the subtle narrowing of the eyebrows in the anger sequence is detected (small red and blue patterns) as well as AU 1 (inner brow raise) during the fear sequence (small purple patterns).

4.5.2 VNMF-OFE for Human Actions

To highlight the robustness and easy parametrization of the VNMF-OFE algorithm the optical flow is visualized for different kinds of videos that include varying lighting conditions, clutter backgrounds and image noise, different kind of articulated movements and moving cameras. For all the videos the identical basis vector set and parameters are used. Parts of the videos with the corresponding optical flow fields are shown in fig. 4.13.

For the two videos of the UCF-Sports dataset, the estimated optical flow is very sparse and the estimation is erroneous, because the frame rate is relatively low compared to the speed of the performed movements. Still, the main movements are captured by the algorithm. In case of fast movements, the algorithm focuses on the borders of the moving structure and is less dense compared to the optical flow estimated *e.g.* on the Weizmann dataset.

4.6 Summary & Discussion

As discussed in the introduction of this chapter, OFE is an ill-posed problem that via the brightness constancy assumption can be interpreted as a correspondence problem. The proposed VNMF-OFE algorithm is a differential method that uses the physical constraint that neighbors move in a similar way. The novel element is the use of a global translation-invariant VNMF model for the optical flow that is included into the BCE and that allows the learning of the local neighborhoods.

The experimental results show that learning the local neighborhood in form of the basis vectors is possible and leads to a detail preserving optical flow estimation. What is more important is the robustness of the entire VNMF-OFE algorithm. The experiments show that the same parameters and basis vectors can be applied for a variety of different videos. The fact that the algorithm remains stable without any fine tuning of the parameters highlights the robustness and usability of the approach and the underlying assumptions.

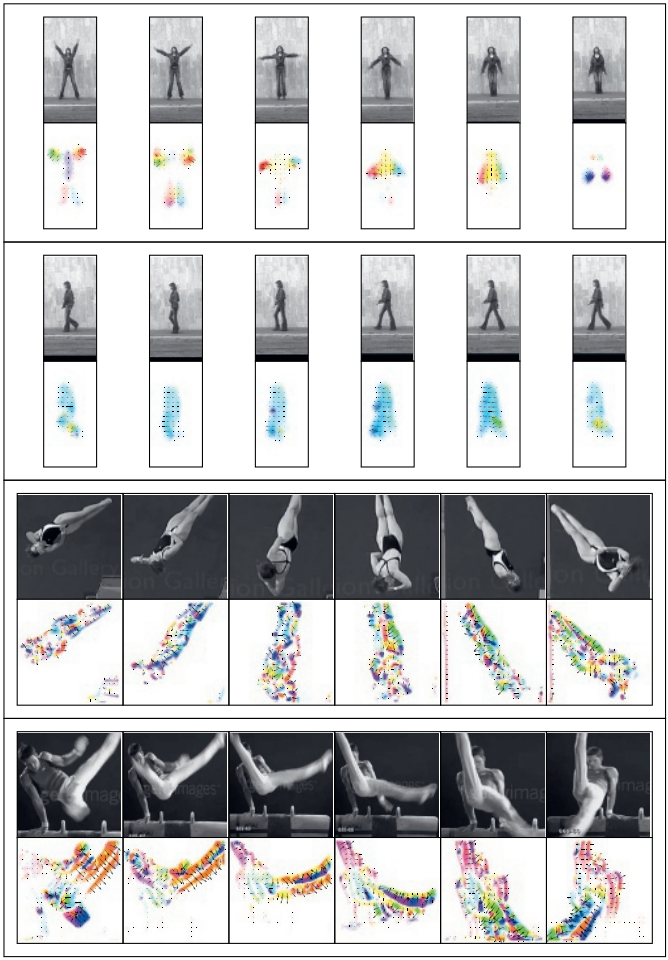

Figure 4.13: The figure shows six consecutive images of a video along with the corresponding optical flow fields. From top to button: A *jumping jack* sequence from the Weizmann dataset [8], a *walking* sequence of the same dataset, a *diving* sequence from the UCF-Sports dataset [88] and a *gymnastics* sequence from the same dataset.

In comparison to state-of-the-art OFE algorithms, the VNMF-OFE algorithm is not as dense and not as accurate as the highly tuned approaches.

However, these highly optmized algorithms, *e.g.* [98], are not easy applicable to human action recognition datasets, because they are not robust enough to deal with the different video qualities and movement speeds encountered in these datasets. That is why the VNMF-OFE algorithm has been choosen to provide the optical flow fields for the subsequent feature extraction.

5 Feature Extraction

A central part of the proposed system for biological motion recognition is the feature extraction. The features are given by the complex cell response of the simple cell patterns. The simple cell patterns need to be learned beforehand. To understand the feature extraction process the two aspects are analyzed in the following:

1. Batch learning of the simple cell patterns and
2. Calculation of the simple cell and complex cell response during the detection.

In the first two sections of this chapter the pattern learning on two types of input data is discussed. First, optical flow patterns that combine shape and motion information, thus *dynamic form patterns* and second, gradient amplitude patterns, that describe local image structures, thus *static form patterns*. The third section introduces and compares the simple cell/complex cell response to the popular HOG/HOF [21] feature descriptors.

As discussed in chapter 3, the goal is to learn local parts-based patterns; for this purpose the VNMF algorithm as introduced in section 3.5.5 is applied to learn the dynamic and static form patterns. The algorithm and the learning procedure have multiple variables, such as the dependence on the data the patterns are learned on, parameters of the different energy term contributions, and the model parameters. The evaluation of the extracted patterns is split up into two parts. In this chapter the visual observable attributes, such as the parts-basedness and the topological sparsity of the activations, in addition to the energy terms, are discussed. The classification properties of the learned patterns are analyzed in detail in chapter 6.

Due to the multiple parameters that influence the learning process, it is neither computational feasible, nor does it bring any deeper insight, if all combinatorial possible parameter combinations are analyzed. Hence, baseline settings are defined and applied if not mentioned otherwise. The

number of iterations is set to 300 and the baseline settings are

$$J = 16, \qquad \lambda_{\mathrm{h}} = 0.1,$$
$$mRFS = 16 \times 16, \qquad \lambda_{\mathrm{part}} = 0.2.$$

5.1 Optical Flow Patterns

To learn the dynamic form patterns, the input for the VNMF are the optical flow fields learned with the VNMF-OFE algorithm. Any details lost during the OFE influence the quality of the learned dynamic form patterns, because the VNMF, as an unsupervised learning algorithm, depends on the data given for learning. It is crucial that the previous layer is capable of preserving the important details. And further on, it is important which videos are used as the training input for the VNMF. Since the goal is to learn patterns related to human actions, it is natural to learn the patterns on a human action recognition dataset. Here a subset of the Weizmann human action recognition dataset [8] is chosen. The subset contains 4 persons performing 4 different actions: *walking*, *running*, *jumping jack* and *waving with two arms* as shown in fig. 5.1. The performed actions include a large variety of natural limb and full body motions which make them well suited for the learning task at hand.

5.1.1 Preprocessing

To achieve a non-negative representation, the optical flow fields are split up into four distinct directions *right*, *up*, *left*, *down* as discussed in section 3.4.3. An example is shown in fig. 5.2. The input has four feature dimensions which are represented in the basis vectors.[1)] The resulting basis vectors are optical flow fields and the activations are scalar values. This allows the basis vectors to describe not only translational, but any kind of optical flow patterns.

In addition, the amplitude of the optical flow field is normalized using the maximum norm for each video. The normalization guarantees that the input values all lie in a similar range, which makes the parameterization easy. However, the absolute velocity value of the optical flow field is lost.

[1)] Note that in case of the VNMF-OFE the additional feature dimension was stored in the activations.

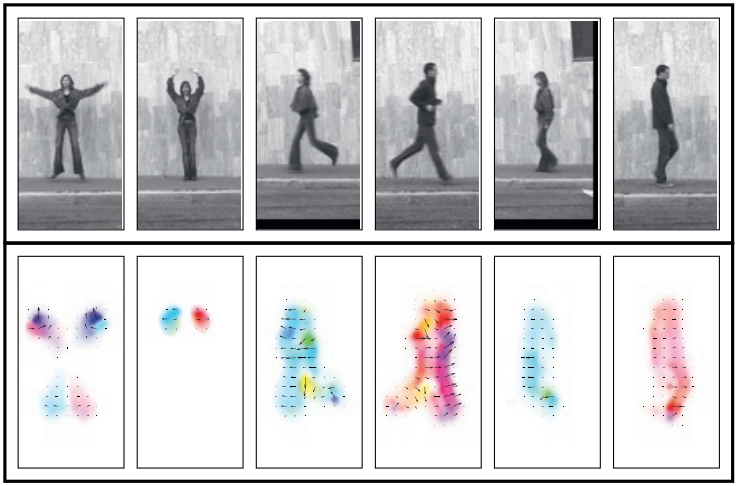

Figure 5.1: Six images and the corresponding optical flow fields of the videos used for learning the basis vectors. The actions from left to right: *jumping jack*, *waving with two arms*, *running left*, *running right*, *walking left* and *walking right*.

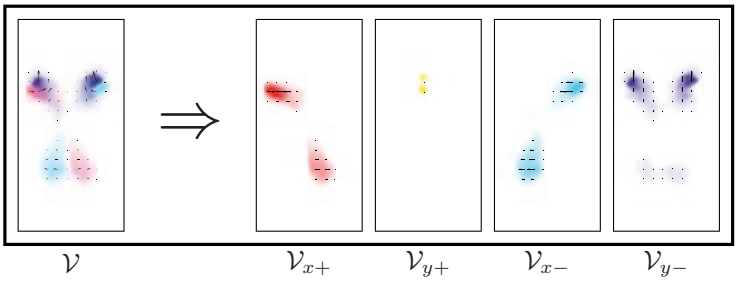

Figure 5.2: To achieve a non-negative representation of the optical flow fields, the input $\mathcal{V}$ is split up into four distinct directions, each with its own non-negative representation.

5.1.2 Varying Energy Parameters

An important set of parameters are the energy parameters controlling the sparsity (λ_{h}) and topological sparsity (λ_{part}) of the decomposition. The

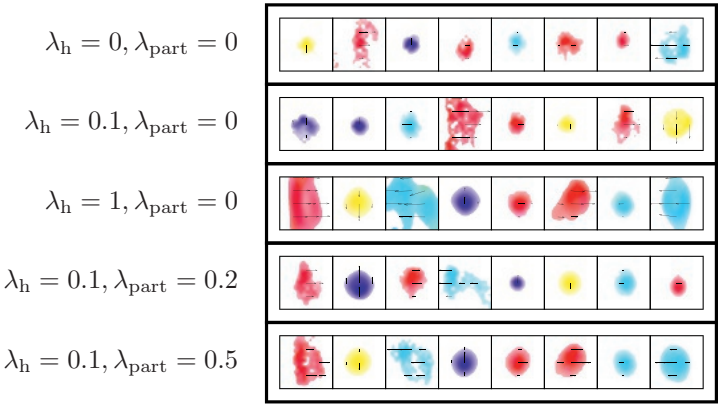

Figure 5.3: Five basis vector sets containing $J = 8$ basis vectors. For a better visualization, the size of the basis vectors is doubled compared to the corresponding input data throughout this chapter.

influence of the two energy terms on the decomposition is analyzed in the following.

Fig. 5.3 depicts basis vector sets containing $J = 8$ basis vectors, each set learned with different parameter combinations. The basis vector sets for $\lambda_h = 0$ all include the trivial solution of a basis vector that spans just a single pixel. Due to the translation invariance these basis vectors are too generic and do not yield any useful information. Another trivial solution is achieved when λ_h is set to one, then the penalty energy term dominates the reconstruction energy and no meaningful reconstruction is achieved. The results show that a sparsity parameter of $\lambda_h > 0$ is required to learn meaningful basis vectors and that the upper bound is approximately $\lambda_h < 0.5 \wedge \lambda_{part} < 0.5$.[2)]

The effect of the topological sparsity parameter λ_{part} can best be examined by comparing the activity patterns depicted in fig. 5.4. The activity images learned with $\lambda_{part} = 0.2$ are topologically sparse and yield a small number of dominant and sharply localized activations that are located all over the moving body parts, *e.g.* on the limbs or on the head. Since only a single activity is used to reconstruct a specific area, the corresponding basis vectors tend to represent this specific part, *e.g.* the head or a limb. The

[2)] Note that these parameters only work when the input has been normalized using *e.g.* the maximum norm during the preprocessing step.

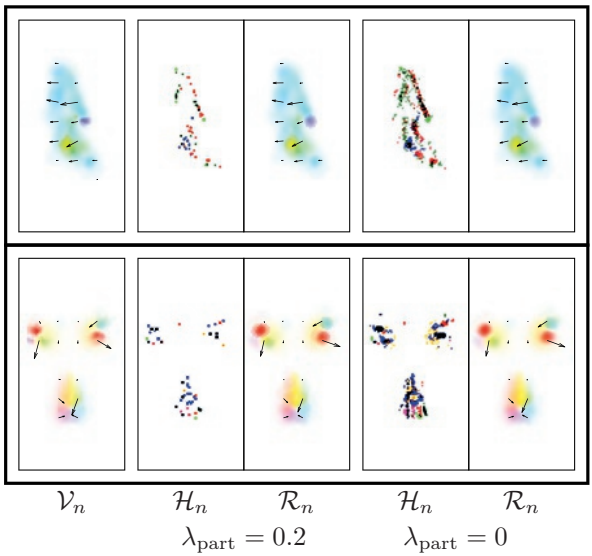

Figure 5.4: Two reconstructions, one with and one without the enforced parts-basedness λ_{part}, for two example inputs. In the upper row the reconstruction is performed on a *walking* input and in the lower row on an input of a *jumping jack* sequence. From left to right: the optical flow input $\mathcal{V}_n$, the summed activation image $\mathcal{H}_n$ (different colors correspond to different basis vectors) and the corresponding reconstruction $\mathcal{R}_n$, first for $\lambda_{\text{part}} = 0.2$ and then for $\lambda_{\text{part}} = 0$.

activity patterns obtained with $\lambda_{\text{part}} = 0$ are much more blurry, therefore the corresponding basis vectors are less distinct.

For a quantitative analysis the average reconstruction error, the sparsity per input and the parts-basedness of the reconstruction for both parameter settings are compared in fig. 5.5 for varying J. For $\lambda_{\text{part}} = 0.2$ the focus on topological sparsity comes at the cost of reconstruction quality, which results in a larger reconstruction error. Since the model is restricted to a limited number of basis vectors with bounded receptive fields, not all possible patterns can be explicitly represented. In other words: the degrees of freedom for the learning algorithm is decreased by enforcing a non-negative and topological sparse representation. As a compensation the restrictions can be relaxed otherwise, *e.g.* by increasing the number of basis vectors. For $\lambda_{\text{part}} = 0.2$ the VNMF algorithm needs a larger J to

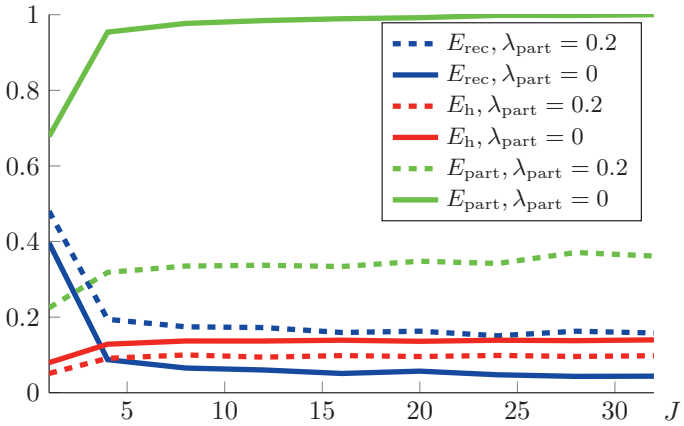

Figure 5.5: Normalized reconstruction (blue), sparsity (red) and parts-basedness (green) energy for varying number of basis vectors J. The green parts-based energy visualized shows the overlap of partial reconstructions, even though it is not penalized during optimization (*i.e.* $\lambda_{part} = 0$).

achieve the same reconstruction quality as for $\lambda_{\text{part}} = 0$, since it is enforced to generate a topologically sparse representation. The sparsity and the parts-based energy is always lower for $\lambda_{\text{part}} = 0.2$ compared to $\lambda_{\text{part}} = 0$, because the parts-basedness energy term forces the algorithm to use fewer activations.

5.1.3 Varying Basis Vector Parameters

Next, the effects of different basis vector parameters J and $mRFS$ on the learned basis vector sets $\mathcal{W}$ is analyzed in more detail. The number of basis vectors is varied in the range $J \in \{8, 16, 24\}$ and the maximum receptive field size in the range $mRFS \in \{8 \times 8, 16 \times 16, 24 \times 24\}$.[3)]

Fig. 5.6 shows basis vector sets learned for the different parameters. The main observation is that by increasing the $mRFS$, more discriminative basis vectors can be learned and therefore the algorithm benefits from an increased J. For the smallest $mRFS$ (8×8) only a few different basis vectors are represented and several basis vectors are redundant. The middle sized (16×16) basis vectors already make use of the increased J and up

[3)] The results have been previously reported in [47].

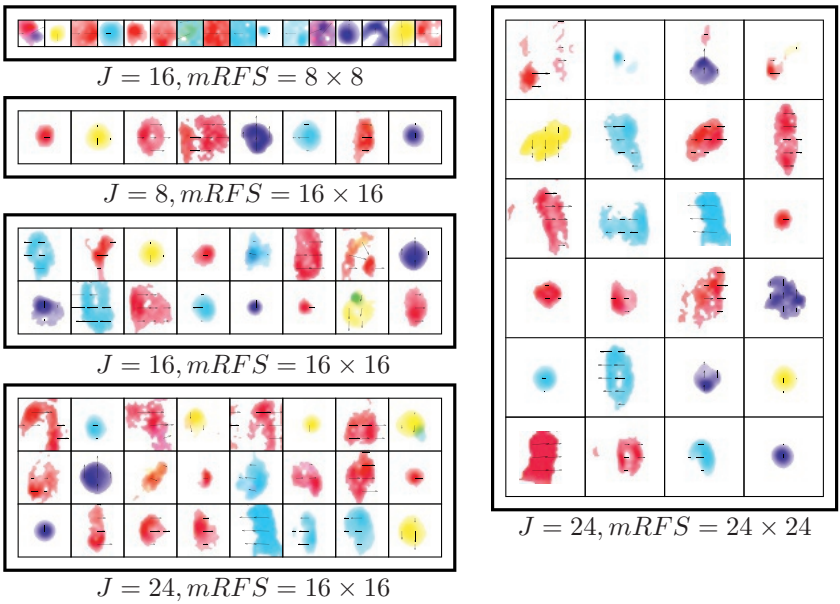

Figure 5.6: Five basis vector sets with varying number of basis vectors J and different maximum receptive field sizes $mRFS$.

to 16 different basis vectors are extracted. However, a further increase of J produces mostly redundant basis vectors. The small and middle sized basis vector sets are rather homogeneous concerning the expressed shape size of each individual basis vector. When the $mRFS$ is further increased (24×24) two kinds of patterns emerge. On the one hand, large and highly prototypical patterns that describe almost entire human body parts and on the other hand, small patterns with similar shapes as extracted for the smaller $mRFS$.

5.1.4 Detailed Analysis of the Learning Process

To get a better understanding on how the different energy terms affect the basis vectors during the learning process, the energy terms over the number of iterations are visualized in fig. 5.7 for the baseline parameters. First, the reconstruction is optimized, as indicated by the large drop of the reconstruction energy. In addition, the sparsity energy is highly decreasing,

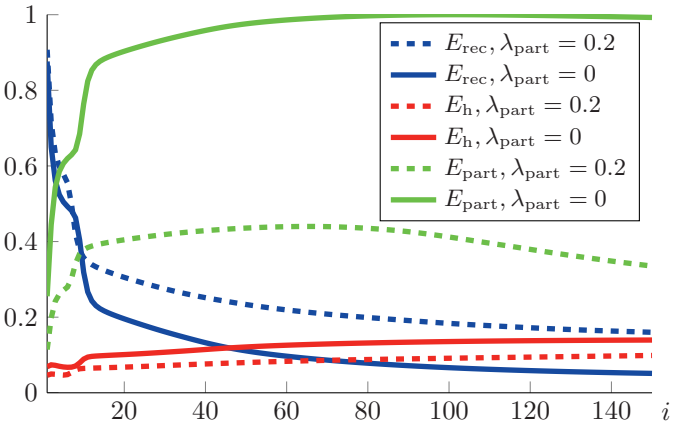

Figure 5.7: The normalized reconstruction (blue, E_{rec}), sparsity (red, E_{h}) and parts-based (green, E_{part}) energy terms over $N = 150$ iterations, for a learning with (dashed, $\lambda_{\text{part}} = 0.2$) and without (solid, $\lambda_{\text{part}} = 0$) the parts-based energy term. The green parts-based energy visualized shows the overlap of partial reconstructions, even though it is not penalized during optimization (*i.e.* $\lambda_{part} = 0$).

because the randomly initialized activations are not very sparse. Throughout the optimization the basis vectors and thus the related reconstruction converge and therefore the corresponding activations increase, which leads to a decrease in the sparsity. Thus, the sparsity energy term is again increasing over time.

The contribution of the parts-based energy term is increasing at the beginning of the learning process. The properties of the basis vectors throughout the learning process are visualized in fig. 5.8. Because the basis vectors are not parts-based at the beginning an overlap of the partial reconstructions is required to achieve a reasonable quality of the optical flow. Thus, the reconstruction energy dominates the parts-based energy. Once the reconstruction converged to a desirable level, the activations start to get more topological sparse and the basis vectors get more parts-based and as a consequence the parts-based energy is decreasing.

In summary the optimization can be split into three stages. First, after the initialization all energies drop to the initial optimization step. Then, the reconstruction is optimized, suppressing the sparsity and parts-based term,

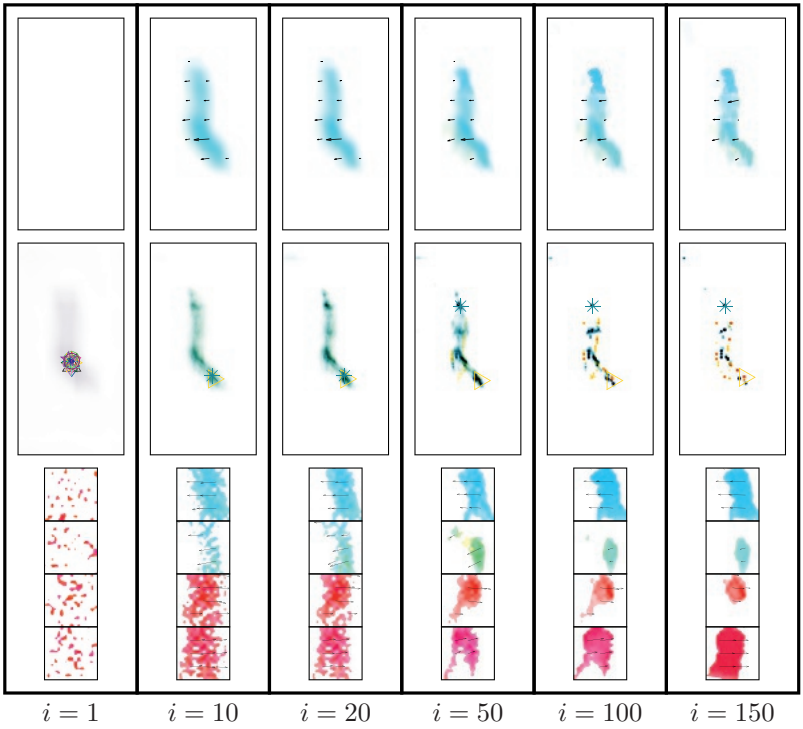

Figure 5.8: From upper row to lower row: The reconstruction, summed activations and four example basis vectors at different iteration steps of one learning process. The symbols (start, triangle,...) on the activations mark the activations with the highest amplitudes. Different symbols correspond to different basis vectors.

which are increasing. When the reconstruction quality has reached a critical level, the optimization focuses on sparsifing the activations topologically, while retaining the same or even further optimizing the reconstruction quality.

5.1.5 Comparison to PCA and sNMF

The VNMF algorithm is now compared to two non-translation-invariant algorithms, the classic PCA and the sNMF algorithm described in sec-

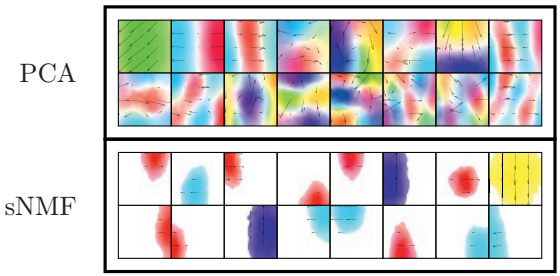

Figure 5.9: The upper block visualizes the first 16 basis vectors or *principal components* learned with PCA on randomly selected 16×16 optical flow patches. The lower block shows 16 basis vectors learned with the multidimensional extended sNMF algorithm for the same input data. For comparable VNMF results see fig. 5.6.

tion 3.3.5, using the non-negative representation for the multidimensional input as introduced in section 3.4.2. For the input of the PCA and sNMF a large set of 16×16 patches is extracted at random positions from the optical flow fields.

The basis vectors learned with the two algorithms are shown in fig. 5.9. In all sets basis vectors with horizontal motion dominate those with vertical motion, which is in good accordance with the intuitive observation that horizontal human movements like walking, running, *a.s.o.* are statistically more frequent than vertical movements like jumping or hand waving. Due to the non-negativity and sparsity constraints, the sNMF basis vectors are more parts-based than the holistic PCA patterns. A further distinction between the sNMF and the PCA is, that the sNMF favors purely translational patterns, even though the basis vectors are not restricted concerning the distribution throughout the different movement directions. This property is rather a result from the inherent motion statistics contained in the data, since all elements of rigid body parts yield consistent translational movements. The main distinction between the different basis vectors is thus the form and overall movement direction. Compared to the basis vectors learned with sNMF the VNMF basis vectors, as shown in fig. 5.6, have more ellipsoid-like forms and are better related to limb forms.

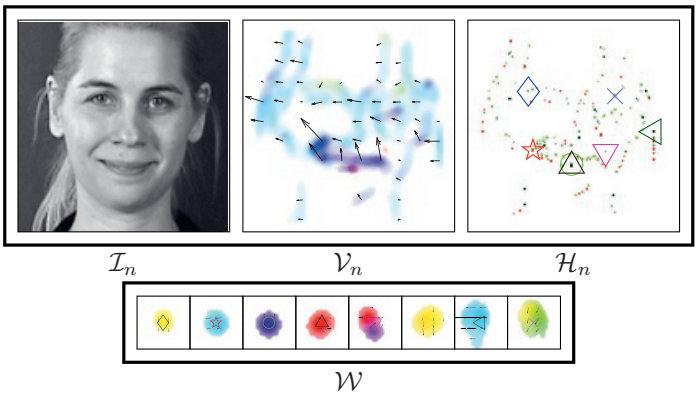

Figure 5.10: From left to right: Input image $\mathcal{I}_i$ from the MMI dataset [104], corresponding optical flow $\mathcal{V}_i$, summed activity image $\mathcal{H}_n = \sum_j \mathcal{H}_{jn}$ (different colors correspond to different $\mathcal{H}_{jn}$) and extracted basis vector set $\mathcal{W}$. The activations with the highest amplitudes are marked with symbols that correspond to specific basis vectors. The activations are located on moving face parts such as the corners of the lips, corresponding to facial action units.

5.1.6 Basis Vectors learned on Face Data

To show the generality of the VNMF algorithm another kind of input data is analyzed. The VNMF algorithm learns a set of $J = 8$ basis vectors on a dataset showing face movements. As discussed in section 4.5.1, action units, *i.e.* the movement of distinct face parts, such as the eyebrows or the lips, are strongly related to emotional states. Face movements are thus very important features in inter-human non-verbal communication.

In fig. 5.10, 8 learned basis vectors and the corresponding activities for one example input motion field are shown. Similarly to the activation patterns learned on the human full body movements, the activities are topologically sparse. Changes in the number of basis vectors and the $mRFS$ have the same effect as for the human full body movements. The level of detail preserved by the VNMF algorithm relays highly on the quality of the underlying optical flow estimation, which unfortunately is not able to preserve all detailed movements of the action units. Nonetheless, the movements obtained by the optical flow are represented by the basis vectors and further localized by the corresponding activations.

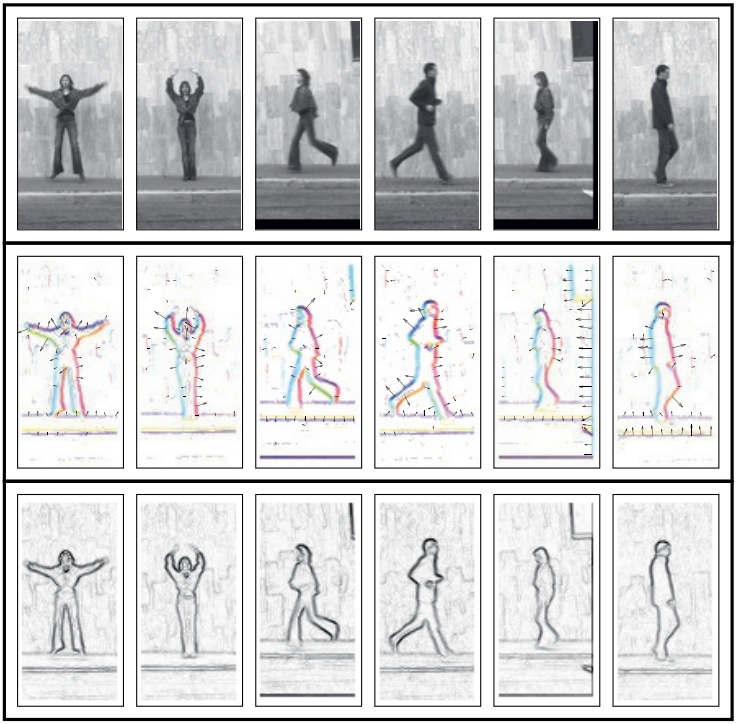

Figure 5.11: Six images (upper row) and the corresponding spatial gradients (middle row) and gradient amplitudes (lower row) of the videos used for learning the basis vectors for static pattern information. The actions from left to right: *jumping jack*, *waving with two arms*, *running left*, *running right*, *walking left* and *walking right*.

5.2 Gradient Patterns

In addition to the dynamic form patterns based on the optical flow, a set of static form patterns is learned with the VNMF algorithm based on the spatial gradients. The static patterns do not require any movement and serve as basis for recognizing the parts of a pose that are not in motion.

The dataset that is chosen as input for the learning algorithm is the same as for the optical flow fields and depicts four humans performing four different full body movements as shown in the upper row of fig. 5.11.

5.2.1 Preprocessing

The preprocessing of each input image consist of three steps. First, the spatial gradients are calculated and in the second step the gradient amplitudes are calculated based on gradient vector field. The third and final step is the normalization of each image using the maximum norm of the gradient amplitudes. In the following the first two steps are shortly discussed.

The spatial gradients are calculate with simple gradient filters. For an image $\mathcal{I}$ the gradients $\mathcal{I}_x$ in x-direction are calculated using the gradient filter $f_x = \begin{pmatrix} -1 & 0 & 1 \end{pmatrix}$ and for the gradients $\mathcal{I}_y$ in y-direction the gradient filter $f_y = \begin{pmatrix} -1 & 0 & 1 \end{pmatrix}^\top$ is applied. The filter operation is the two dimensional convolution

$$\mathcal{I}_x = \mathrm{conv}_2(\mathcal{I}, f_x), \tag{5.1}$$

$$\mathcal{I}_y = \mathrm{conv}_2(\mathcal{I}, f_y). \tag{5.2}$$

The result is a two dimensional vector field that contains positive and negative values. A negative value corresponds to a gradient from bright to dark and a positive value corresponds to a gradient from dark to bright. The gradient vector field has thus the same dimensionality and range of value as the optical flow fields and similarly, the multidimensional VNMF could be applied directly on the gradient field. However, the sign of the gradients is different *e.g.* when a person is moving with bright clothes in front of a dark background or when a person is moving with dark clothes in front of a bright background. To be invariant towards these kind of brightness dependencies, the gradient *amplitude* is used as input for the VNMF algorithm. For each element of $\mathcal{I}(\boldsymbol{x})$ of an input image $\mathcal{I}$ the non-negative gradient amplitude value is

$$\mathcal{I}_{\mathrm{abs}}(\boldsymbol{x}) = \sqrt{\mathcal{I}_x(\boldsymbol{x})^2 + \mathcal{I}_y(\boldsymbol{x})^2}. \tag{5.3}$$

Along with the sign of the gradient vector field, the direction is neglected when using the gradient amplitude alone. The gradients are typically orthogonal to the image structure and the image structure is to some extend preserved in the learned gradient patterns. An additional representation of the gradient direction is not required and would only increase the input dimensionality. The gradient amplitudes for examples of the input data used for the learning algorithms are shown in fig. 5.11 together with the corresponding images and the image gradients.

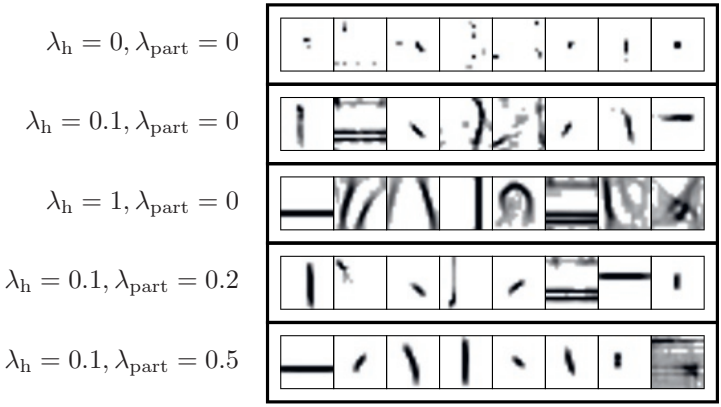

Figure 5.12: Five basis vector sets, each with $J = 8$ basis vectors.

5.2.2 Varying Energy Parameters

The learned basis vector set for variations in the energy parameters, *i.e.* the sparsity parameter and the parts-based parameter, are visualized in fig. 5.12. The parametrization of the resulting basis vectors shows an analogous dependency as the optical flow patterns, *e.g.* a sparsity parameter set to $\lambda_h = 0.1$ results in meaningful patterns, while for a sparsity parameter $\lambda_h = 0$ the trivial basis vectors emerge. The parts-based parameter λ_{part} has a similar effect as the sparsity parameter and further enforces prototypical structures in the basis vectors.

The activity patterns for two cases are depicted in fig. 5.13. Again, the parameter dependencies are similar to the ones observed during the learning of the optical flow patterns. The activity images learned with $\lambda_{part} = 0.2$ are topologically sparse and the corresponding basis vectors resemble local gradient structures. The activity patterns obtained with $\lambda_{part} = 0$ are much more dense and the learned basis vectors are less distinct. Unlike the activations from the optical flow patterns, which are located on moving body parts, the activations for the gradient patterns are located along both sides of the limbs and define the borders of the body shape, rather than the body shape itself.

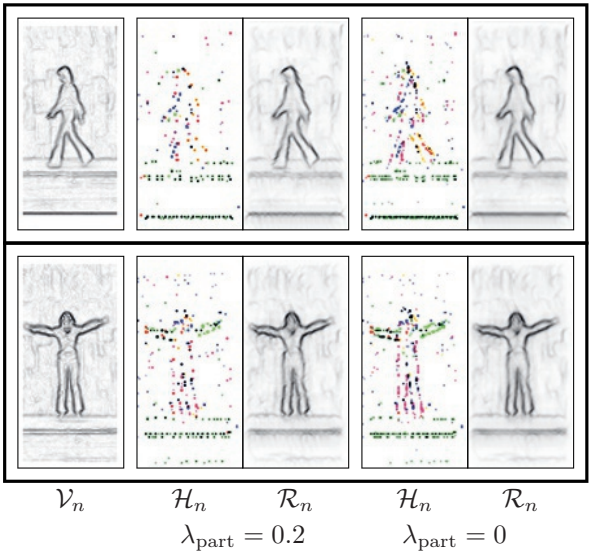

Figure 5.13: Two reconstructions, one with and one without the enforced parts-basedness λ_{part}, for two example inputs. In the upper row the reconstruction is performed on a *walking* input and in the lower row on an input of a *jumping jack* sequence. From left to right: the optical flow input $\mathcal{V}_n$, the summed activation image $\mathcal{H}_n$ (different colors correspond to different basis vectors) and the corresponding reconstruction $\mathcal{R}_n$, first for $\lambda_{\text{part}} = 0.2$ and then for $\lambda_{\text{part}} = 0$.

5.2.3 Varying Basis Vector Parameters

In the following, basis vector sets learned with different basis vector parameters J and $mRFS$ are analyzed. Five basis vector sets for varying number of basis vectors J and maximum receptive field sizes $mRFS$ are visualized in fig. 5.14.

The parameter dependency is again similar to optical flow patterns. Increasing the $mRFS$ results in an increasing number of different basis vectors. For $J = 8$ mostly small, generic basis vectors emerge. By increasing the number of basis vectors more specific patterns are learned. The largest amount of basis vectors, *i.e.* $J = 24$, shows prototypical patterns that resemble complete parts of the input. A key difference to the optical flow patterns is that the amount of redundant basis vectors is very low. The gradient structures are more complex than the movement structures and

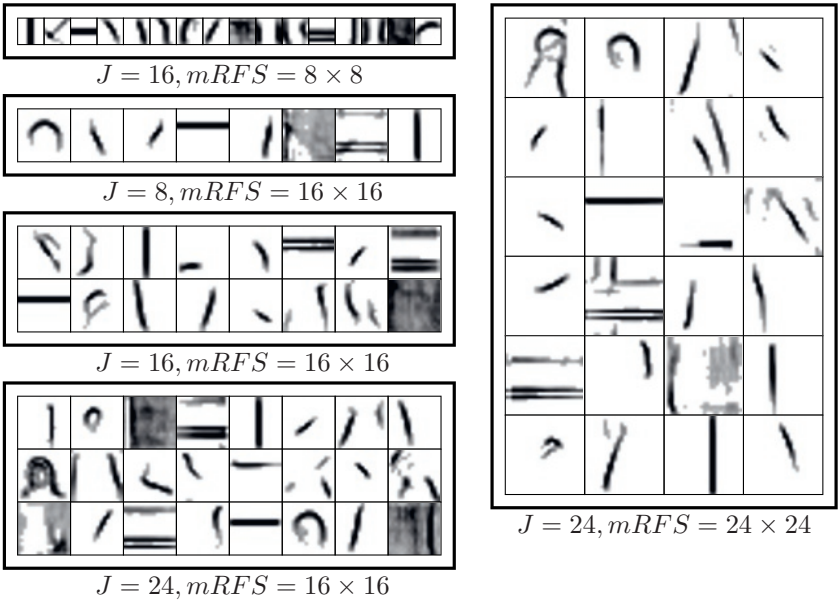

Figure 5.14: Five learned basis vectors sets with varying number of basis vectors J and different maximum receptive field sizes $mRFS$.

thus more specific to the underlying texture information. As a result a larger number of different basis vectors emerged during the learning process.

5.2.4 Detailed Analysis of the Learning Process

Fig. 5.15 shows the reconstruction, activations and four examplary basis vectors for different iteration steps during a learning process. After 10 iterations, three of the randomly initialized basis vectors already exhibit a bar-like structure and the reconstruction depicts the human figure and the gradient at the border of the image. At the beginning, most of the activations are grouped at the strongest input structures and the highest activation amplitudes of multiple basis vectors are in the same location. As a consequence, the basis vectors resemble a similar structure. After 50 iterations the basis vectors become more specific and the corresponding activations are spread throughout the image and not as spatially concentrated as during the beginning of the learning process. This high flexibility

of the learning process is directly related to the multiplicative update rules, which can lead to drastic changes in the activations and basis vectors. As a consequence, the activations and basis vectors can change strongly throughout the learning process and the overall algorithm is not depending on a specific initialization.

Similar to the learning of the optical flow patterns, the effect of the parts-based energy term that enforces the topological sparsity influences the activations and the basis vectors at the end of the learning process. Once the activations are spatially distributed, the basis vectors become more and more specific and the penalty due to overlapping receptive fields results in the sharply localized activations as shown for iteration numbers 100 and 150.

5.2.5 Comparison to PCA and sNMF

The patterns learned with two non-translation-invariant algorithms, the classic PCA and the sNMF algorithm described in section 3.3.5, on the gradient amplitudes are now discussed. For the input of the PCA and sNMF 16×16 patches are randomly extracted from the gradient amplitude images.

The basis vectors learned with the two algorithms are shown in fig. 5.16. The PCA patterns resemble different holistic filters with frequencies and orientations. The sNMF basis vectors are local blocks without any specific structure. Unlike the basis vectors learned with the translation-invariant VNMF algorithm, neither PCA nor sNMF learns prototypical structures, like horizontal and vertical bars.

5.2.6 Basis Vectors learned on Face Data

In addition to the human full body movements, a set of basis vectors is learned on a dataset that shows human faces [104].

In fig. 5.17, 8 learned basis vectors and the corresponding activities for one example input video are shown. Similar to the activation patterns learned on the human full body movements, the activities are topologically sparse and parameter variations have the same effects. The basis vectors are again divided into few highly generic parts and prototypical, template-like basis vectors, that resemble face parts. Due to the high image contrast on the left part of the input image, the gradient amplitudes on the left side are higher than on the right side. As a consequence, all of the marked

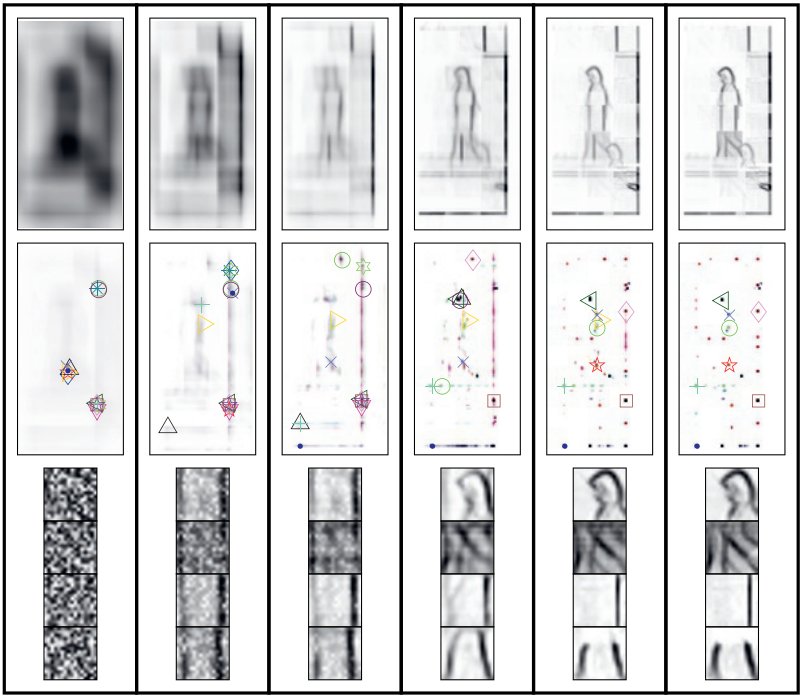

Figure 5.15: Development of gradient-based basis vectors during learning, each column showing a different iteration step. From upper row to lower row: The reconstruction, activations and four examplary basis vectors during different iteration steps of a single learning process. The symbols on the activations mark the activations with the highest amplitudes (one symbols for each basis vector).

activations, *i.e.* the activations with the highest amplitude, are located in the left part of the image.

5.3 VNMF as Feature Descriptor

Once the optical flow and gradient patterns are learned, they can be applied in the FFNN to compute the features. Thus, the two kinds of feature descriptors used are the simple cell/complex cell responses related to the

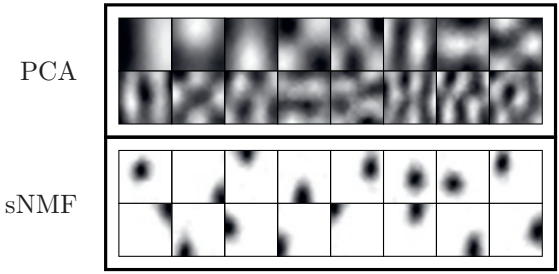

Figure 5.16: The upper block visualizes the first 16 basis vectors or *principal components* learned with PCA on randomly selected 16×16 optical flow patches. The principal components contain positive and negative values, in the visualization gray values correspond to zero, white pixels to negative and black to positive values. The lower block shows 16 basis vectors learned with the sNMF algorithm for the same input data.

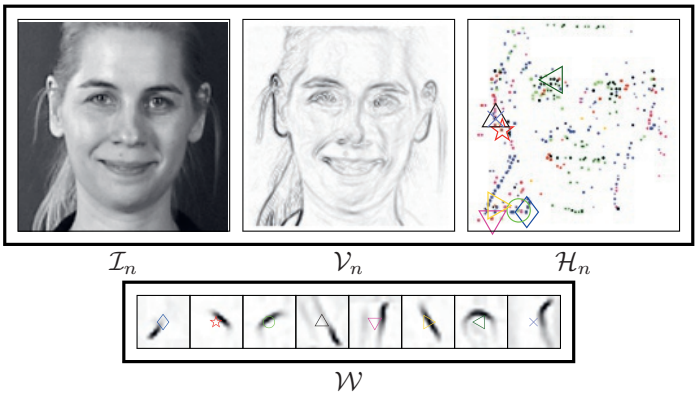

Figure 5.17: From left to right: Input image $\mathcal{I}_n$ from the MMI dataset [104], corresponding gradient amplitude $\mathcal{V}_n$, summed activity image $\mathcal{H}_n = \sum_j \mathcal{H}_{jn}$ (different colors correspond to different $\mathcal{H}_{jn}$) and extracted basis vector set $\mathcal{W}$. Unlike the movement specific activations of the optical flow patterns shown in fig. 5.10, the activations for the gradient patterns are located all over the face.

patterns learned on the two input types, *i.e.* optical flow and gradient amplitudes. The calculation of a simple cell/complex cell layer of a FFNN as discussed in section 2.2.1 consists of four stages: *preprocessing*, the *simple*

cell response, a *non-linear post processing* and the *complex cell response*. The simple cell response is basically a similarity measure between an input and each of the simple cell patterns. While the simple cells are *selective* for specific patterns, the complex cell patterns *pool* a limited set of simple cell responses inside an overlapping block grid to a common activation.

The preprocessing for the feature descriptors is identical to the preprocessing for the pattern learning, *i.e.* the input is normalized using the maximum norm. Once the simple cell patterns $\mathcal{W}$ have been learned, the simple cell response $\mathcal{H}_n$ for a given input $\mathcal{V}_n$ can be calculated. In the following two different kinds of simple patterns are compared: First, a simple correlation between the patterns and the input and second a refinement of the $\mathcal{H}_n$ using the VNMF update rules for the activations. This refinement is a non-linear interaction of the individual activations. The simple cell response calculations for the optical flow and the gradient amplitude input are illustrated in fig. 5.18. The complex cell response is an overlapping spatial pooling operation, that is later discussed in section 5.3.2 and shown in fig. 5.21.

One of the interesting properties of the VNMF features is the compact representation of local gradient or optical flow structures. Popular feature descriptors like SIFT [70] or HOG/HOF [21] lose the exact structure information, because they are based on local histograms. The similarities and differences of the VNMF and the HOG/HOF features will be discussed at the end of this section.

5.3.1 Simple Cell Response

Before the simple cell response can be calculated, a set of basis patterns $\mathcal{W}$ has to be selected. Fig. 5.19 shows 24 optical flow and gradient amplitude patterns that are selected out of the various sets learned in the previous two sections. The examples visualized in this section use these selected basis vectors.

A simple way to calculate the simple cell response $\mathcal{H}_{\text{corr}}$ for a given input $\mathcal{V}_n$ to a set of patterns $\mathcal{W}$ is to use the two dimensional correlation

$$\mathcal{H}_{\text{corr}} = \mathcal{H}_{jn} = \text{corr}_2(\mathcal{V}_n, \mathcal{W}_j), \quad \forall j \in [1, ..., J]. \tag{5.4}$$

The resulting activation pattern for an example optical flow input are shown in fig. 5.20. The activations are very blurry, because slightly shifted patterns still have a high correlation value and there is no interaction between the single activations. The VNMF learning algorithm includes

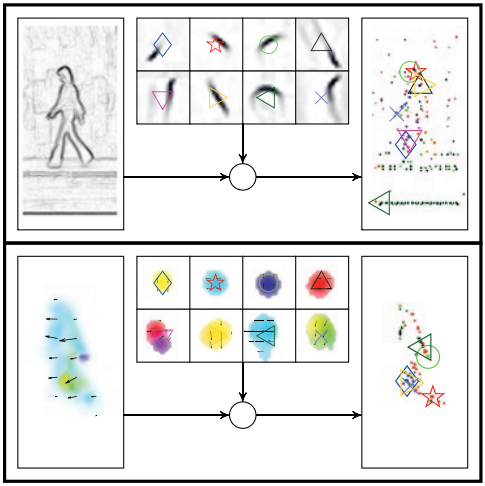

Figure 5.18: The simple cell calculation for gradient (upper row) and optical flow (lower row) inputs. The input data is reconstructed using a set of prelearned basis vectors and the corresponding activations are then fed into a classification stage. Different colors in the activation images (left side) correspond to different basis vectors. The activations with the highest amplitudes are marked. The maximal number of iterations is set to 150.

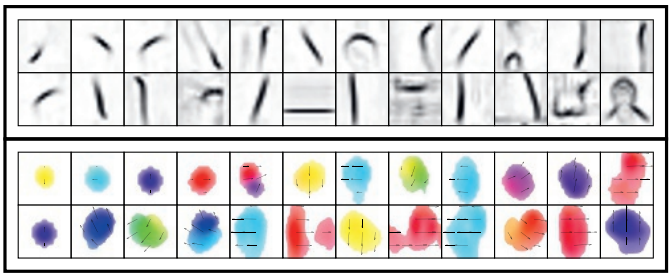

Figure 5.19: Two sets of 24 selected basis vectors learned with the VNMF algorithm. The upper set contains basis vectors learned on spatial gradient amplitudes and the lower set contains optical flow patterns. The basis vectors are selected from various sets learned with the VNMF algorithm and are ordered by size.

these interactions, like a lateral competition between concurring activations as enforced in the parts-based energy term. The second method to calculate the simple cell response is therefore based on the VNMF update rules with a fixed basis vector set with the convolution as initialization. The simple cell response is termed $\mathcal{H}_{\text{refine}}$ and the algorithm is

- Initialize $\mathcal{H}_{\text{refine}} = \text{corr}_2(\mathcal{V}_n, \mathcal{W}_j), \quad \forall j \in [1, ..., J]$.
- Loop for N iterations
  1. Calculate $\mathcal{R}_n = \sum_j \text{conv}_2(\mathcal{H}_{jn}, \bar{\boldsymbol{W}}_j)$,
  2. update $\mathcal{H}_{\text{refine}} \rightarrow \mathcal{H}_{\text{refine}} \circ \frac{(\nabla_{\mathcal{H}_{jn}} E_{\text{VNMF}})^-}{(\nabla_{\mathcal{H}_{jn}} E_{\text{VNMF}})^+}, \quad \forall j \in [1, ..., J]$.

The energy parameters are set to $\lambda_{\text{part}} = 0.2$ and $\lambda_{\text{h}} = 0.1$ and the maximal number of iterations is set to $N = 10$. The resulting activations patterns are depicted in fig. 5.20. Compared to the simple correlation, the activations are increasingly topologically sparse, but not as sparse as the activations during learning. This is due to the low number of iterations. Increasing the number of iterations leads to more topologically sparse activations as depicted in fig. 5.18, but has linearly increasing computational costs.

To avoid the calculation of a large number of iterations, a simple non-linear post processing (nl) is performed. *I.e.* the activations are thresholded in a way that small valued activations are set to zero. There are two kinds of threshold values. First, an absolute value τ_{abs} which ensures that noisy parts of the input do not result in an activation, and a relative value τ_{res}, which depends on the highest activation in a predefined local neighborhood. The local neighborhood is defined by the spatial pooling block cells $A(\boldsymbol{x})$ discussed in the next subsection. The mathematical operations are

$$h_{jn}(\boldsymbol{x}) := 0, \quad \forall h_{jn}(\boldsymbol{x}) < \tau_{\text{abs}}, \tag{5.5}$$

$$h_{jn}(\boldsymbol{x}) := 0, \quad \forall h_{jn}(\boldsymbol{x}) < \tau_{\text{rel}} \cdot \max_{\boldsymbol{y} \in A(\boldsymbol{x})} (h_{jn}(\boldsymbol{y})). \tag{5.6}$$

The threshold values are set to $\tau_{\text{abs}} = 0.05$ and $\tau_{\text{abs}} = 0.1$. The post-processed activations for the correlation $\mathcal{H}_{\text{corr+nl}}$ and the refined activations $\mathcal{H}_{\text{refine+nl}}$ are shown in fig. 5.20. They have an increased topological sparsity, because small activations are clipped to zero by the thresholding. However, they are still far from being as sparse as the refined activations using $N = 150$ iterations shown in fig. 5.18.

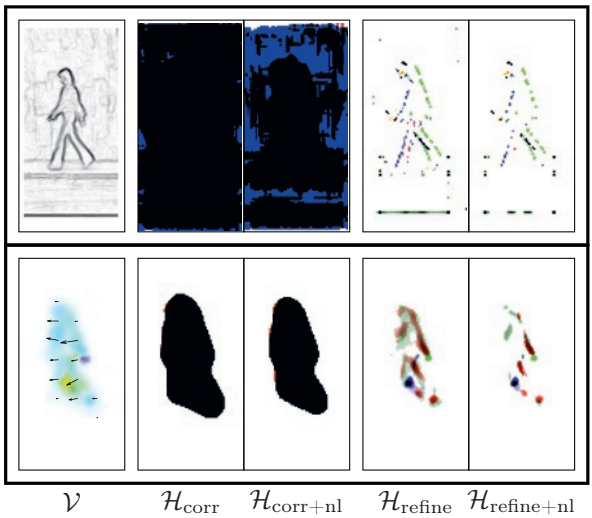

Figure 5.20: Different simple cell responses for two kinds of inputs, gradient amplitudes (upper row) and optical flow fields (lower row). From left to right: The input $\mathcal{V}$, the correlation response $\mathcal{H}_{\text{corr}}$, the correlation response with the non-linear post processing $\mathcal{H}_{\text{corr+nl}}$, the correlation response refined with $N = 10$ VNMF updates without $\mathcal{H}_{\text{refine}}$ and with the non-linear post processing $\mathcal{H}_{\text{refine+nl}}$. Different colors correspond to different activations. Multiple activations on the same spatial position result in an addition of the colors, *i.e.* a black spot indicates the presence of multiple activations of different basis vectors.

5.3.2 Complex Cell Response

The complex cell response used in the proposed FFNN is a *spatial pooling* operation with 50% overlapping blocks. The basic idea of the spatial pooling is, that the features should be invariant to small shifts. In addition the invariance reduces the spatial resolution and thus the feature dimension significantly. The pooling operation is illustrated in fig. 5.21. In the example the number of pooling blocks is set to $5 \times 2 = 10$ blocks, as shown in the right image of fig. 5.21. Since the blocks have a 50% overlap in both directions, there exist $6 \times 3 = 18$ cells as illustrated in the middle image of fig. 5.21. Except for the cells at the border of the image, each cell is part of four overlapping blocks. The simple cell activations in the example

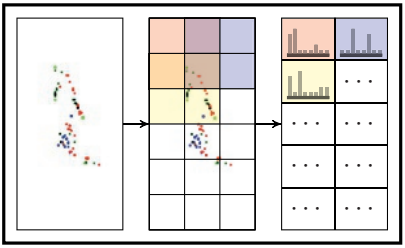

Figure 5.21: The spatial pooling process of a simple cell response. From left to right: The simple cell response, the pooling grid with the overlapping pooling blocks and the resulting descriptors. The background colors mark the pooling area for three example pooling blocks.

have a dimensionality of $Y \times X \times J = 128 \times 64 \times 8 = 65536$. The pooled activations have a dimensionality of $5 \times 2 \times 8 = 80$.

Pooling is further necessary, because even though the image is centered around the person, the exact position of key elements of poses, such as the head or the arms, vary strongly depending on different factors, such as view points or the person performing the action. Due to the spatial pooling, the exact position is lost and the complex cell response for different position is identical, thus invariant to the shifts. However, it is important that the invariance is locally bound, since the relative position of certain features is a significant information to differentiate different poses. *E.g.* , during waving the arms reach the area above the head, while during walking they stay below the shoulders. *I.e.* the pooling range has to be a compromise between the *generality* of the feature and its *class-specific* properties. More specific features with smaller pooling blocks can be used when the figure centering is robust and accurate and the number of variations can be captured by the training data. The fewer training data that is available, the more invariances have to be build in the system.

The overlap of the blocks smoothes the complex cell responses. Without overlapping blocks, the invariance to the shifts would end at the exact border of the pooling block. As a consequence a shift of one pixel at the border would result in an uncorrelated complex cell response. Due to the overlap a simple cell activation in an overlapping cell activates two feature responses, one for each of the overlapping blocks. If the activation is shifted outside the overlapping cell it will still remain in one of the overlapping blocks, even though it left the receptive field of the other block.

In mathematical terms, the complex cell response $\mathcal{C}_n$ of an simple cell activation $\mathcal{H}_n$ is

$$c_{jn}(\boldsymbol{y}) = \sum_{\boldsymbol{x} \in A(\boldsymbol{y})} h_{jn}(\boldsymbol{x}), \quad \forall j \in [1, ..., J], \tag{5.7}$$

with $A(\boldsymbol{y})$ defining all pixels inside the block at block position $\boldsymbol{y}$.

The pooling block size during the classification was choosen to 32×32 pixels which leads to $7 \cdot 7 = 49$ blocks. This size resulted from a compromise between the preserved locality information and the desired invariance to shifts. For $J = 8$ basis vectors the feature dimension is thus 392 for the gradient and the optical flow. The combined features have a dimensionality of 784. Increasing the number of basis vectors J further increases the feature dimension.

5.3.3 Relation to HOG/HOF Descriptor

The VNMF patterns resemble explicit local structures and the complex cell responses retain a coarse representation of the location of these local structures in the input. Other types of image descriptors also describe the local structures, but not by locating explicit structural elements, but rather by representing the local statistics of the gradient or optical flow field, *e.g.* by using histograms of discrete gradient directions as feature descriptors.

One popular method is the *Histogram of Oriented Gradients* (HOG) introduced by Dalal and Triggs [21] as descriptors for pedestrian detection, or the feature descriptors used in the SIFT descriptors introduced by Loewe [70] for object recognition. In the following the similarities and differences between the HOG/HOF descriptors and the VNMF descriptors are discussed.[4)] In related work, Le et al. [64] compare the classification performance of patterns learned with ISA, a two layered extension of the well known ICA [54], to extract spatio-temporal features for human action recognition. They show that the learned pattern features outperform the classic HOG/HOF and 3D HOG descriptors on multiple human action recognition datasets.

The HOG descriptor is a hand-designed statistical description of gradient structures via a histogram, where each entry corresponds to a discrete gradient direction, a so called bin. The descriptors are able to represent class discriminative structures and are computationally cheap. However,

[4)] The discussion is based on the publication [46].

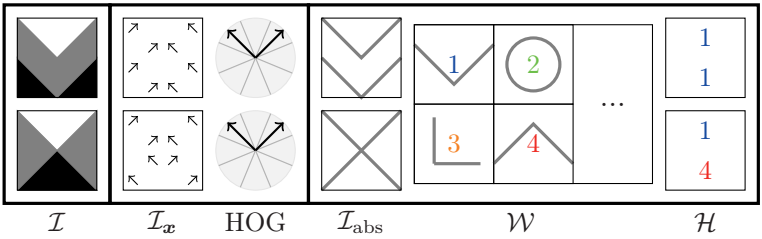

Figure 5.22: The HOG and VNMF descriptor for two 16×16 images (upper and lower row). From left to right: Artificial input image ($\mathcal{I}$), spatial gradients ($\mathcal{I}_{\boldsymbol{x}}$), HOG descriptor (HOG) with $b = 8$ bins, gradient amplitudes ($\mathcal{I}_{\text{abs}}$), basis vector set ($\mathcal{W}$) with $J = 8$ basis vectors, activations ($\mathcal{H}$). The activations 1 and 4 mark the position, where the corresponding basis vectors $\mathcal{W}_1$ and $\mathcal{W}_4$ are placed for the reconstruction of the input.

the simplistic histogram description has natural limitations. It discards local spatial relations between structural elements, *i.e.* the topology of the gradients is neglected, because the explicit spatial occurrence of the gradients is lost in the histogram representation. Furthermore, the number of elements in each descriptor block is limited by the number of bins.

The HOG descriptor consists of two parts: First, a grid of 50% overlapping blocks. The cell/block structure is identical to the cell/block structure of the spatial pooling cells used during the calculation of the complex cell responses described in section 5.3.2. The second step is the calculation of a normalized histogram of the oriented gradients in each of the blocks. The block descriptor is build in three steps: First, each gradient vector (in case of HOF, optical flow vectors) is binned into one of *e.g.* $b = 8$ distinct directions. Second, for each block the gradient vectors are summed up for each bin, resulting in a histogram with b elements. To achieve invariance to contrast changes, the histograms are normalized using the Euclidean norm.

The main similarity is that for both descriptor types, the input image is fragmented in a grid of overlapping blocks. The block grid captures the *global* spatial relations between the block descriptors, *e.g.* the upper blocks are more likely to describe head shapes, while the lower blocks reflect features corresponding to leg poses and movements. However, the features differ in the way they describe what is inside the blocks, *i.e.* the *local* topological information in the image.

Fig. 5.22 shows the HOG descriptors and the VNMF activations for two 16×16 artificial input images. The HOG descriptors for both blocks are identical, because the blocks differ only in the spatial structure of the gradients, and not in the amount of gradient vectors, which is captured by the histograms. In contrast, the pooled activations are different, because different basis vectors are used for the reconstruction. This artificial example illustrates why in principle pattern-based descriptors are able to preserve the *local* topological information in cases where the histogram descriptor discards this information.

Another difference is *how* the image structure is described. The binning approach is simple and computationally cheap. Nevertheless, the number of bins b is limited, because a finer binning makes the HOG descriptor less invariant and may not increase its discriminative properties. On the contrary, the sparsity constraints in the VNMF algorithm allow the learning of an overcomplete basis, so the number of basis vectors J is not as limited as the number of bins, because the more basis vectors are learned, the more image structures can be explicitly represented. Besides, the basis vectors are learned and not hand-crafted as the HOG, so they are easier to adapt to different kinds of input data.

In summary, the VNMF descriptors should outperform the HOG descriptors if the *local topological information* is important for modeling discriminative image descriptors. In the following chapter, this hypothesis is evaluated in classification experiments.

6 Human Action Recognition

The goal of the proposed FFNN is the classification of human movements such as gestures, facial expressions or human full body movements, *e.g.* actions. The classification performance is evaluated on two benchmarks for *Human Action Recognition* (HAR), the Weizmann [8] and UCF-Sports dataset [88] and a benchmark for *Facial Expression Recognition* (FER) of Dollar et al. [25]. The evaluation focuses on the classification performance of the FFNN as proposed in chapter 2. The influence of the two feature types, *i.e.* the *static* (gradient amplitude) and *dynamic* (optical flow) form patterns is analyzed for different basic vector sets learned with varying energy (λ_{h} and λ_{part}) and basis vector (J and $mRFS$) parameters. The classification performance of the patterns learned with the VNMF algorithm is compared to patterns learned with PCA and sNMF as well as with HOG/HOF descriptors. The evaluation includes the two types of how to calculate the simple cell response as discussed in section 5.3.1. The overall system is further compared to related results of state-of-the-art algorithms.

6.1 Support Vector Machine (SVM)

The final supervised learning layer of the proposed FFNN contains a multiclass *Support Vector Machine* (SVM) [14, 105], a classifier often applied in computer vision. SVMs are still a vivid research topic, details about the mathematical definitions and properties can be found *e.g.* in [6]. For details on the implementation see the related work on the LIBSVM library [14].

Since the Weizmann and UCF-Sports datasets have no defined training and testing sets, the classifiers are trained and evaluated in leave-one-out experiments. The persons used for the learning of the basis vectors on the Weizmann dataset are discarded in the evaluation. The FER dataset [25] has only two different persons, so one person is used for training and one for testing. This procedure is applied to both persons.

The SVM is learned with an RBF Kernel and a soft-margin parameter to increase robustness. The corresponding parameters are obtained using

Figure 6.1: Example images for the ten classes, *bending*, *jumping jack*, *hopping*, *jumping*, *running*, *skipping*, *jump on one leg*, *walking*, *waving with one arm* and *waving with two arms* of the Weizmann dataset [8].

5-fold cross validation on the training data. Once the SVM classifiers are trained, each frame of each video is classified individually. The final classification result for each video is the weighted average of all its frame results. The weights are the class probabilities provided by the SVM.

6.2 Results for Different Basis Vector Sets

The classification performance for the different basis vector sets is evaluated on the two HAR datasets. The 10-class Weizmann dataset [8] shown in fig. 6.1 and the 9-class UCF-Sports dataset [88] as depicted in fig. 6.2 differ strongly in the complexity and variations. The Weizmann dataset is filmed with a static camera and no view-point variations and the actions are performed by the different persons in the dataset in a similar manner. The difficulty encountered in the dataset are the high similarities between the different actions. While *e.g. bending* can be easily differentiated to *walking*, there are multiple classes, *e.g. hopping*, *running*, *skipping*, *jump on one leg*, *walking* with very similar poses. To the contrary, the actions in the UCF Sports dataset have strong variations in the viewpoints and in the individual performance of the actions, as discussed in section 1.1.1.

The default parameters for the learned simple cell patterns are

$$J = 16, \qquad \lambda_{\mathrm{h}} = 0.1,$$
$$mRFS = 16 \times 16, \qquad \lambda_{\mathrm{part}} = 0.2.$$

And the default parameters for calculating simple cell/complex cell response are

$$\lambda_{\mathrm{part}} = 0.2, \qquad \lambda_{\mathrm{h}} = 0.1,$$

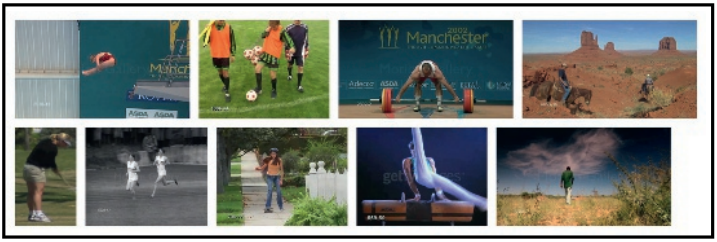

Figure 6.2: Example images for the nine classes, *diving*, *kicking*, *weight lifting*, *horse riding*, *golfing*, *running*, *skateboarding*, *gymnastics* and *walking* of the UCF Sports dataset [88].

calculated with the correlation method eq. (5.4) including the non-linear refinement eq. (5.6) and a pooling block size of 32×32. The results for the optical flow are marked with $\mathcal{V}_{\boldsymbol{x}}$ and the gradient patterns are marked with $\mathcal{I}_{\boldsymbol{x}}$. The optical flow has been calculated using the method introduces in chapter 4.

6.2.1 Varying Basis Vector Parameters

Table 6.1 shows the classification results based on the optical flow patterns. There is no strong parameter dependency for both datasets. Surprisingly, a higher number of basis vectors does not enhance the classification performance significantly.

Table 6.1: Classification results for the optical flow ($\mathcal{V}_{\boldsymbol{x}}$) patterns for different J and $mRFS$.

mRFS	8×8			16×16			24×24		
J	8	16	24	8	16	24	8	16	24
Weiz.	0.98	0.99	0.99	0.99	0.99	0.97	0.97	0.99	**1.00**
UCF	0.89	0.88	0.87	0.89	0.89	0.90	0.89	**0.91**	0.89

The results are similar for the gradient patterns (table 6.2) and the combined patterns (table 6.3). While an increased number of basis vectors does not seem to have an effect on the classification performance, a larger $mRFS$ improves the results in all three cases.

Table 6.2: Classification results for the gradient patterns ($\mathcal{I}_{\boldsymbol{x}}$) patterns for different J and $mRFS$.

mRFS	8×8			16×16			24×24		
J	8	16	24	8	16	24	8	16	24
Weiz.	0.60	0.58	0.58	0.63	0.70	0.64	**0.79**	0.69	0.69
UCF	0.81	0.83	0.81	0.85	**0.87**	0.87	0.85	**0.87**	0.87

The dynamic optical flow patterns outperform the static gradient patterns for both datasets. While the combined patterns give the best results for the UCF-Sports dataset, the optical flow patterns outperform the combined patterns for the Weizmann dataset.

Table 6.3: Classification results for the combined use of optical flow ($\mathcal{V}_{\boldsymbol{x}}$) and gradient patterns ($\mathcal{I}_{\boldsymbol{x}}$) for different J and $mRFS$.

mRFS	8×8			16×16			24×24		
J	8	16	24	8	16	24	8	16	24
Weiz.	0.92	0.90	0.90	0.92	0.93	0.90	0.92	0.92	**0.94**
UCF	0.91	0.91	0.91	0.92	**0.93**	0.92	**0.93**	0.91	0.91

6.2.2 Varying Energy Parameters

Table 6.4 shows the results for the VNMF patterns learned with different energy parameters. For both datasets, the patterns learned with the parts-based parameter set to $\lambda_{\text{part}} = 0.2$ significantly outperform the patterns learned without the parts-based energy. These results indicate that the topological sparsity learns prototypical patterns that are class discriminative. Like for the varying basis vector parameters, the dynamic optical flow patterns outperform the static gradient patterns.

6.2.3 Comparison to PCA and sNMF Patterns

Table 6.5 and table 6.6 show the classification results for the VNMF patterns compared patterns extracted with PCA and sNMF. The VNMF patterns clearly outperform the PCA and sNMF patterns on both datasets. This

Table 6.4: Classification results for different $\lambda_{\text{part}} \in \{0, 0.2\}$ for the optical flow ($\mathcal{V}_{\boldsymbol{x}}$), gradient ($\mathcal{I}_{\boldsymbol{x}}$) and combined ($\mathcal{V}_{\boldsymbol{x}}$+$\mathcal{I}_{\boldsymbol{x}}$) patterns.

	$\mathcal{V}_{\boldsymbol{x}}$		$\mathcal{I}_{\boldsymbol{x}}$		$\mathcal{V}_{\boldsymbol{x}}$+$\mathcal{I}_{\boldsymbol{x}}$	
λ_{part}	0	0.2	0	0.2	0	0.2
Weiz.	0.94	**0.99**	0.67	0.70	0.95	0.93
UCF	0.77	0.89	0.47	0.87	0.77	**0.93**

further underlines the benefits of the topological sparse and translation invariant learning procedure.

Table 6.5: Classification results for optical flow ($\mathcal{V}_{\boldsymbol{x}}$) and gradient ($\mathcal{I}_{\boldsymbol{x}}$) patterns learned with PCA, sNMF and VNMF.

	$\mathcal{V}_{\boldsymbol{x}}$			$\mathcal{I}_{\boldsymbol{x}}$		
	PCA	sNMF	VNMF	PCA	sNMF	VNMF
Weiz.	0.94	0.94	**0.99**	0.65	0.66	0.70
UCF	0.71	0.80	0.89	0.65	0.70	0.87

Table 6.6: Classification results for combined ($\mathcal{V}_{\boldsymbol{x}}$+$\mathcal{I}_{\boldsymbol{x}}$) patterns learned with PCA, sNMF and VNMF.

	$\mathcal{V}_{\boldsymbol{x}} + \mathcal{I}_{\boldsymbol{x}}$		
	PCA	sNMF	VNMF
Weiz.	0.94	0.94	0.93
UCF	0.83	0.80	**0.93**

6.2.4 Varying Simple Cell Response

Next, the results for the two types of simple cell responses as discussed in section 5.3.1 are compared. Table 6.7 shows the results for the correlation response and the correlation refined with the VNMF update rules. The correlation response outperforms the refined simple cell response for both datasets. This result indicates that the SVM profits from the non-sparse representation of the correlation. This is somehow contradictive to the

Figure 6.3: Example images for the six classes, *anger*, *disgust*, *surprise*, *fear*, *happiness* and *sadness* of the FER dataset [25].

idea of a sparse decomposition, but computationally beneficial, because the correlation method is the fastest way to calculate the simple cell response.

Table 6.7: Results for two types of calculated simple cell responses: The correlation response (Corr) explained in eq. (5.4) and the correlation response with the VNMF refinement (Corr+Ref) for the optical flow ($\mathcal{V}_{\boldsymbol{x}}$), gradient ($\mathcal{I}_{\boldsymbol{x}}$) and combined ($\mathcal{V}_{\boldsymbol{x}}$+$\mathcal{I}_{\boldsymbol{x}}$) patterns.

	$\mathcal{V}_{\boldsymbol{x}}$		$\mathcal{I}_{\boldsymbol{x}}$		$\mathcal{V}_{\boldsymbol{x}}$+$\mathcal{I}_{\boldsymbol{x}}$	
	Corr	Corr+Ref	Corr	Corr+Ref	Corr	Corr+Ref
Weiz.	**0.99**	0.93	0.70	0.68	0.93	0.92
UCF	0.89	0.89	0.87	0.68	**0.93**	0.89

6.3 Facial Expression Recognition

To show that the proposed FFNN is not restricted to the recognition of human actions, the algorithm is evaluated on a dataset for *Facial Expression Recognition* (FER) as depicted in fig. 6.3. The goal of the classification is to differentiate the six basic emotions introduced by Ekman [29][1)].

Table 6.8 and table 6.9 show the classification results for the gradient, optical flow and combined patterns learned with PCA, sNMF and the VNMF algorithm on the FER dataset. For all input types, the VNMF outperforms the other learning algorithms. The dynamic form patterns learned on the optical flow outperform the static form patterns learned on the gradient amplitudes, while the combined patterns perform best. This results indicate that the low-scale movements are captured by the

[1)] For details on FER see discussion in section 4.5.1.

VNMF-OFE algorithm and that the motion related to the facial action units (see table 4.1) is indeed very class-discriminative.

Table 6.8: Results for optical flow ($\mathcal{V}_{\boldsymbol{x}}$) and gradient ($\mathcal{I}_{\boldsymbol{x}}$) patterns learned with PCA, sNMF and VNMF on the FER dataset.

$\mathcal{V}_{\boldsymbol{x}}$			$\mathcal{I}_{\boldsymbol{x}}$		
PCA	sNMF	VNMF	PCA	sNMF	VNMF
0.72	0.70	0.75	0.67	0.66	0.71

Table 6.9: Results for the combined ($\mathcal{V}_{\boldsymbol{x}}$+$\mathcal{I}_{\boldsymbol{x}}$) patterns learned with PCA, sNMF and VNMF on the FER dataset.

$\mathcal{V}_{\boldsymbol{x}}$ + $\mathcal{I}_{\boldsymbol{x}}$		
PCA	sNMF	VNMF
0.69	0.72	**0.82**

6.4 Comparison to Related Work

In the following the classification results of the VNMF descriptors are compared to the state-of-the-art HOG/HOF descriptors [21] and the overall system is compared to other HAR systems.

6.4.1 HOG/HOF Results

To make the comparison of the learned basis vectors to state-of-the-art features extractors independent of the figure-centering and optical flow estimation in the preprocessing, the HOG/HOF features are calculated on the same data as used for the learned basis vectors. The cell/block building of the HOG features is identical to the overlapping summation pooling blocks used for the complex cell response. The same pooling sizes is used for both features types. To make the feature dimension identical the number of basis vectors (J) is set equal to the number of bins (b) typically used for the HOG descriptor: $b = J = 8$.

The results are depicted in table 6.10. Throughout all datasets, the dynamic form patterns (optical flow) outperform the static form patterns

(gradient), while the combined features (optical flow + gradient) perform best on the UCF-Sports and Facial Expression dataset. This result is of particular interest, because it shows that the dynamic information contributes more to the recognition of biological motion than the static information. However, each stream on its own is able to recognize some of the actions, and the information from both streams is complementary, since the results improve considerably when combining form and motion.

Table 6.10: Results for the VNMF and HOG/HOF descriptors for the optical flow ($\mathcal{V}_{\boldsymbol{x}}$), gradient ($\mathcal{I}_{\boldsymbol{x}}$) and combined ($\mathcal{V}_{\boldsymbol{x}}$+$\mathcal{I}_{\boldsymbol{x}}$) patterns.

	$\mathcal{V}_{\boldsymbol{x}}$		$\mathcal{I}_{\boldsymbol{x}}$		$\mathcal{V}_{\boldsymbol{x}}$+$\mathcal{I}_{\boldsymbol{x}}$	
	VNMF	HOF	VNMF	HOG	VNMF	HOG/HOF
Weiz.	**0.99**	0.86	0.63	0.80	0.92	0.87
UCF	0.89	0.78	0.85	0.77	**0.92**	0.80
FER	0.31	0.39	0.75	0.63	**0.82**	0.71

The learned pattern features outperform the designed state-of-the-art HOG/HOF descriptors significantly for all three datasets. As discussed in section 5.3.3, the fact that the VNMF descriptors outperform the HOG descriptors shows that the *local topological information* is important for modeling discriminative image descriptors.

6.4.2 Benchmark Results

The experimental results of the proposed FFNN are now compared to other HAR systems on four benchmarks for HAR and FER, *i.e.* the Weizmann [8], KTH [92], visualized in fig. 6.4, UCF-Sports [88] and FER dataset by Dollar [25].

Table 6.11 shows the results for the proposed FFNN using the VNMF and the HOG/HOF descriptors as well as related work. The systems most similar to the proposed FFNN are the biologically inspired multilayer network of Jhuang et al. [56] as well as the system proposed by Dean et al. [22]. [56] make use of example-based patterns, no learning is applied during the feature extraction. In [22] the features are learned with a sparse coding algorithm applied to space-time-volumes. Both networks, as well as the HOG/HOF descriptors are outperformed by the proposed FFNN.

Most of the other algorithms [2, 15, 60, 107] are Bag-of-Words approaches as discussed in appendix A. They differ *e.g.* in the way the codebook is

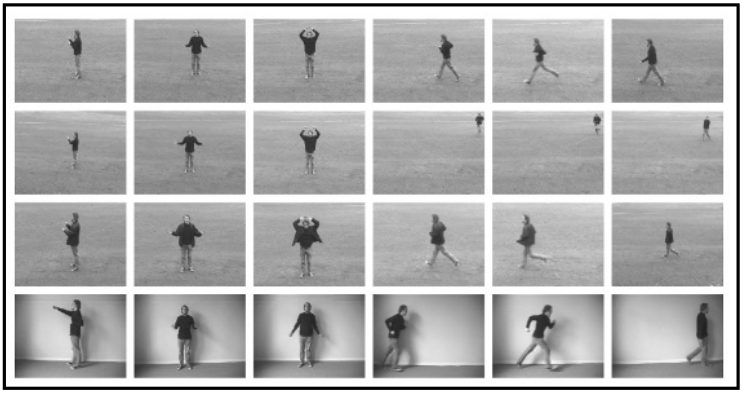

Figure 6.4: From left to right: Example images for the six classes, *boxing*, *clapping*, *waving*, *jogging*, *running* and *walking* of the KTH dataset [92]. Each row depicts one of four sets, the second set includes view-point and scale variations, the third set varying clothing and the fourth set a different background.

learned. While the classical approaches [60] train the codebook with k-means clustering, others use NMF [2], sparse coding [41] or sNMF [15], with improved results.

In addition, the descriptors can be learned, *e.g.* with ISA; the classification results [64] are slighty better than the results of the proposed FFNN on the KTH dataset, but significantly worse on the more challenging UCF-Sports dataset. The only algorithm that outperforms the proposed FFNN on the UCF-Sports dataset is introduced in [15]. Besides a codebook that is learned with sNMF, they added a novel vorticity-based feature-point detector to the Bag-of-Words approach.

In summary, the proposed FFNN outperforms other biological inspired approaches, but does not achieve the best classification performance on the challenging KTH and UCF-Sports dataset, compared to highly optimized state-of-the-art algorithms in computer vision. In the FFNN the VNMF descriptors outperform the HOG/HOF descriptors. Since the high performing algorithms make use of extended HOG/HOF descriptors, the Bag-of-Words approach might benefit from the VNMF descriptors as well.

Table 6.11: Classification results for optical flow ($\mathcal{V}_{\boldsymbol{x}}$), gradient ($\mathcal{I}_{\boldsymbol{x}}$) and combined ($\mathcal{V}_{\boldsymbol{x}}$+$\mathcal{I}_{\boldsymbol{x}}$) patterns on the Weizmann [8], KTH [92], UCF-Sports [88] and FER by Dollar [25] for the VNMF algorithm ($J = 16$, $mRFS = 16\times16$) compared to state-of-the-art HOG/HOF features and related work.

		KTH	Weiz.	UCF	FER
VNMF	$\mathcal{I}_{\boldsymbol{x}}$	0.71	0.80	0.87	0.31
	$\mathcal{V}_{\boldsymbol{x}}$	0.90	**0.99**	0.89	0.75
	$\mathcal{V}_{\boldsymbol{x}}$+$\mathcal{I}_{\boldsymbol{x}}$	0.93	**0.99**	0.93	**0.82**
HOG/HOF	$\mathcal{I}_{\boldsymbol{x}}$	0.67	0.74	0.77	0.39
	$\mathcal{V}_{\boldsymbol{x}}$	0.80	0.86	0.78	0.63
	$\mathcal{V}_{\boldsymbol{x}}$+$\mathcal{I}_{\boldsymbol{x}}$	0.82	0.87	0.80	0.71
Related Work	Jhuang et al. [56]	0.92	0.96	-	-
	Dean et al. [22]	0.86	-	-	-
	Guha et al. [41]	-	**0.99**	-	**0.82**
	Klaser et al. [60]	-	-	0.90	-
	Amiri et al. [2]	**1.00**	-	-	-
	Le et al. [64]	0.94	-	0.87	-
	Wang et al. [107]	-	-	0.89	-
	Chen et al. [15]	0.97	-	**0.99**	-

7 Conclusion

In the following the experimental results for the proposed *Feed-Forward Neural Network* (FFNN) for biological motion recognition, the *optical flow estimation* (VNMF-OFE) and the *learned features* (VNMF) are summarized and critically discussed. Finally, the shortcomings of the proposed algorithms are analyzed and an outlook on future work based on the discussion is given.

7.1 Summary & Discussion

The basis for the different layers in the proposed FFNN is the novel VNMF algorithm introduced in chapter 3, with its focus on a direction selective, strict non-negative representation, sparse activations and inhibition. The basic idea, that the *non-negativity* constraints results in *meaningful*, thus interpretable patterns, which are more class-specific than holistic patterns, is confirmed by the experimental results for all layers. The inability to subtract patterns from a model, whether during the optical flow estimation or for learning gradient or optical flow patterns, guides the underlying optimization problem to local minima related to basis vectors that resemble prototypical input parts. This behavior is further emphasized by the local competition of overlapping patterns introduced with the novel inhibition functions.

Besides the benefit of learning prototypical patterns, the VNMF learning framework is highly robust, easy to parametrize and consists of linear operations which can be easily implemented for parallel computing. Unlike most unsupervised learning algorithms, all energy functions that are optimized with the VNMF algorithm scale relative to the reconstruction and are thus directly coupled. This makes the parametrization independent of the type of the input data as well as from the amount of input data. The algorithm directly scales with the number of inputs and can thus be extended to larger datasets without any re-parametrization. This claim is underlined by the fact that the algorithms were applied with identical parameters to varying input types and different datasets.

7.1.1 Optical Flow Estimation (VNMF-OFE)

The motion analysis starts on the small scale with the estimation of the optical flow, *i.e.* the motion of each pixel between two consecutive images. The VNMF approach changes the OFE problem to an model parameter search problem. The result is a sparse but detailed optical flow. For the VNMF-OFE the local receptive fields are learned and allow for the preservation of small but class-specific movements, like the motion of face parts during facial expressions. The major benefit of the VNMF-OFE method is its robustness. The optical flow model is estimated for the entire input image and not to a local restricted neighborhood, which makes it less vulnerable to the aperture effect.

If any necessary conditions for an successful OFE are violated, *e.g.* in case of really fast movements, the VNMF-OFE focuses on the borders of the moving object and the sparsity suppresses the motion for the unreliable areas. The computational costs of the VNMF-OFE algorithm scale linearly with the number of basis vectors (J) and are independent of the maximum receptive field size ($mRFS$). The most time consuming operations are the two dimensional correlations and convolutions, which need to be performed per basis vector. Since all operations are linear they can be calculated in parallel and independently for each basis vector. An optimized and parallel implementation is thus independent of (J). During learning the algorithm scales with the number of input images, which can also be implemented for parallel computing and during the detection the computational cost are only depending on the size of the input, thus the x- and y-dimension.

However, the proposed method has several drawbacks. The current version does not include a multi-scale approach to care for fast movements. Further on, the euclidean energy function is not very robust towards outliers and thus the estimated optical flow is often not accurate concerning the speed of the movement. The estimated optical flow is not fully dense and not evaluated on standard optical flow benchmarks.

Nevertheless, due to the robustness and the simple parametrization, the VNMF-OFE is a great tool for OFE on datasets for biological motion recognition. And above all, the VNMF-OFE directly links unsupervised pattern learning to the spatio-temporal domain.

7.1.2 Feature Extraction (VNMF)

The goal of the proposed unsupervised learning algorithm is to learn prototypical parts of the input, *i.e.* to give a puzzle-like decomposition.

Following the idea of a *non-negative*, *sparse*, *direction-selective*, *translation-invariance* representation that includes *inhibition* of overlapping parts, the VNMF algorithm achieves the proposed goal. The new inhibition function has a strong effect on the activations, *i.e.* it leads to topological sparse activations, without any additional non-linear function. As a direct consequence the patterns resemble prototypical input parts.

Since all energy terms of the optimization objective function scale relative to the input space, the VNMF has identical properties, whether the inputs are optical flow fields or gradient amplitudes. If given optical flow fields as inputs the VNMF patterns represents body parts. For the gradient amplitudes the VNMF patterns resemble edge structures.

The structure preserving ability of the VNMF patterns is what makes them useful during the classification of human actions. The simple cell/complex cell feature extraction outperforms the state-of-the-art HOG/HOF descriptors throughout all experiments. The parts-based VNMF patterns outperform the patterns learned with sNMF and the holistic PCA, which highlights the benefits of parts-based over holistic representations.

The computational costs of the VNMF algorithm scale linear with the number of basis vectors (J) and are independent of the maximum receptive field size ($mRFS$). The most time consuming operations are the two dimensional correlations and convolutions, which need to be performed per basis vector. Since all operations are linear they can be calculated in parallel and independent for each basis vector. An optimized and parallel implementation is thus independent of (J). During learning the algorithm scales with the number of input images, which can also be implemented for parallel computing and during the detection the computational cost are only depending on the size of the input, thus the x-, y- and feature-dimension.

Learning algorithms, including the proposed VNMF algorithm often suffer from over-fitting to the training dataset. However, in the experiments the patterns learned on the Weizman dataset are directly applied to the UCF-Sports dataset without reduced classification performance, even though the UCF-Sports dataset is in no way similar to the Weizman dataset. *E.g.* the Weizman dataset has no camera movement and a rather homogenous background, while the UCF-Sports dataset has strong camera motion as well as different cluttered backgrounds. The learned patterns can even be applied to a face movement dataset and still achieve good results for FER. This suggests that the VNMF algorithm is capable of learning general purpose, rather than dataset specific patterns.

7.1.3 Biological Motion Recognition Model (FFNN)

The proposed FFNN for biological motion recognition consists of two streams, one for static (*i.e.* gradient amplitudes) and one for dynamic (*i.e.* optical flow) form representations. The results on multiple computer vision benchmarks (Weizman, UCF-Sports and FER) show that the biological inspired model is competitive with state of the art approaches.

It is shown that the static as well as the dynamic patterns contribute to the classification of human actions. While both, the static as well as the dynamic patterns, can achieve reasonable results when applied unpaired, the best results are achieved when both patterns are used in parallel. These results indicate that low-level motion analysis contributes to the recognition of biological motion.

A major difference of the proposed FFNN compared to other hierarchical convolutional models [65] is that the feature stages are trained in a purely unsupervised fashion and no supervised learning, *e.g.* in form of a back-propagation algorithm, is applied to refine the features. As discussed in section 1.1.2, using unsupervised feature learning is beneficial when different classes share identical poses or pose-sequence, which is true for human actions, such as walking, running and kicking.

Another specialty of the proposed FFNN is the multiplicity of how the VNMF algorithm is applied to solve different tasks throughout the stages of the classification hierarchy. By adapting the objective function, but keeping the sparse, non-negative model, the VNMF approach can solve diverse tasks such as OFE, optical flow and gradient feature extraction as well as modeling linear dynamic systems (discussed in appendix D).

All these results strongly suggest that *sparsity*, *non-negative* representations and *inhibition* are coding principles that are beneficial, not only for the proposed FFNN for biological motion recognition, but for multiple kinds of neural networks.

7.2 Outlook

The results reported in this thesis indicate the usefulness of the proposed FFNN and the novel unsupervised learning algorithms. However, the properties of the proposed system need to be further analyzed and extended. In the following, some strait forward extensions for the different stages of the FFNN are listed.

The current version of the VNMF-OFE does not include a multi-scale approach which is typically used to deal with larger displacements. The algorithm should be extended to include a multi-scale optical flow estimation. In addition, the VNMF-OFE could be combined with a robust segmentation (see [99]) to achieve an increasingly dense OF-field. Including temporal relations into the VNMF-OFE as proposed in [112] would further increase the robustness and accuracy of the OFE. The modified VNMF-OFE should be evaluated on state of the art OFE benchmarks [36].

The VNMF-OFE could be modified to solve the correspondence problem in stereo vision. Like OFE, extracting a depth map out of a stereo camera setting is a correspondence problem, which can be solved with the proposed VNMF-OFE algorithm. The additional depth information could provide additional features for the FFNN as well as a segmentation that could improve the quality of the OFE.

The model of the VNMF algorithm is restricted by the fixed number of basis vectors J. To overcome this restriction, the VNMF could be extended by *e.g.* iterative NMF as proposed in [85]. Furthermore, the euclidean error function used for the reconstruction energy term is not robust to outliers in the input data. Hence, more robust energy functionals (for an overview see [19]) could be applied. Most importantly, a faster feed forward simple cell computation during detection should be considered. In [65] an additional feed-forward filter learning is proposed in combination with a sparse coding algorithm. The VNMF could be extended to include this filter learning. During detection only the feed-forward filters with an additional non-linearity could be used to calculate the activations, thus speeding up the detection process significantly.

It should be further investigated how to new inhibition term can be related to the *dropout* [97] technique that is currently used in all hierarchical convolutional networks for object classification. Dropout, *i.e.* the random cancellation of activations during learning, is said to increase the networks sparsity. Unlike the proposed inhibition function, which has a similar effect, dropout is a random heuristic approach whose usefulness is hard to motivate.

To make the proposed biological motion recognition setting applicable in a real world scenario, a robust action detection step [60] would be required. To better understand how the proposed approach scales for a larger amount of classes, the FFNN should be tested on larger datasets like the UCF-101 [96] and HMDB [61]. The scene complexity of these two datasets is comparable to the UCF-Sports datasets, however, they have over 50 different action categories and over 24 hours of video material.

In the current setup, all layers of the FFNN are learned in uncoupled subsequent steps. If the hierarchical NMF approach introduced in [84] would be extended to include the pooling step, all layers could be learned in a unified procedure. To push the idea of the unified algorithm setting even further, the classification layer should be changed from the SVM to a non-negative and sparse algorithm. This could be either done by exploiting the similarities between SVM and NMF [80] or by using a non-negative multi-layer perceptron instead of the SVM. This would require a modification of the back propagation algorithm to care for the non-negative representation. Such a back propagation algorithm could be trained throughout all layers of the FFNN, similar to other convolutional recognition hierarchies [65, 97]. Though, this would contradict the idea of learning the first layers in a purely unsupervised fashion as discussed in section 1.1.2.

And last but not least, the FFNN should be included into a larger system for dynamic scene understanding as discussed in appendix B, including feed-forward recognition as well as context and task driven feed-back loops.

A Bag of Words

The Bag of Words (BOW) method for human action recognition as introduced in [94] is widely applied, because it is robust and has a good classification performance without the need of any pre-processing, such as figure centering or an exact temporal segmentation of subactions. Each video is seen as a 3D space-time-volume, which is represented by a single, high-dimensional feature descriptor. The classification is divided into four steps:

1. 3D *feature point detection*,
2. for each feature point, calculation of a *local spatio-temporal descriptor*,
3. which is then projected onto a prelearned *codebook*,
4. the resulting codebook *histogram* for all the feature points is then classified using a multi-class SVM.

For each step there are different algorithms. A comparison of the classification performance for the feature point detectors and local descriptors is given in [108]. Common local descriptors are 3D extensions of the HOG/HOF descriptors introduced in [21], *e.g.* the dense trajectory descriptors in [107]. The codebook can either consist of a set of randomly selected descriptors [60], can be learned on training data using k-means [60], sparse coding [41], NMF [2] or sNMF [15]. The classification is mostly done with a SVM using radial basis functions or linear kernels.

The major drawback of the BOW method is that the spatio-temporal relations between the local descriptors are lost, because they are treated independently of their relative location in the video. In [60] it is shown that a figure centric representation with a spatial grid improves the results of their BOW method. In related work [5, 109] the spatio-temporal context between the local descriptors is described by designed features.

B Visual Cortex

Processing of visual information starts as early as in the eye, where different cell types on the retina are specialized *e.g.* to differentiate between colors or to be motion sensitive. From the retina, the continuous signals are send to the visual cortex that consists of different types of spiking neurons. As a simplification, in this thesis, we make the assumption that the input is always a discrete time series of images. Furthermore, neurons are not modeled as spiking devices, but by using non-negative continuous signals. This highly simplified approach is *biologically inspired* rather than a realistic description of the actual biological processes. It focuses the analysis on the *functional level* rather than on the *anatomical level*.

The tasks solved by the early visual system include the control of the *eye movement* for object *tracking* and finding relevant *focus points* in an input image. In the visual cortex the tasks have an increasing complexity and include *e.g.* object/action *segmentation*, *detection* and *recognition* and *self localization*. To successfully solve these tasks, information coming from the visual input as well as information from other sensors or higher cognitive areas are merged. If we *e.g.* search a specific person and hear his/her voice, we already know where to look at (focus point) as well as what to look for (prior knowledge about the person). The recognition of visual actions might be to some extent guided by the motoric system and self localization is accomplished by fusing information from the sensors in the inner ears with input from the eye. Sensor fusion as well as the influence of higher cortical areas are not discussed in this thesis. The focus is on primary visual functions [23], *i.e.* feed-forward driven recognition processes. Finding the focus of attention or object detection are also not analyzed. The analyzed persons are always given in a *figure-centric* representation.

The visual information coming from the two eyes can be categorized into four channels: *color*, *gradient*, *depth* and *motion* information. Unlike the color information, the information from three other channels needs to be estimated in the visual cortex. The information in the different channels can either be redundant or complementary, so a visual recognition system that makes use of all four channels might give the best results, but a reduced system can still reach a high performance. This redundancy makes

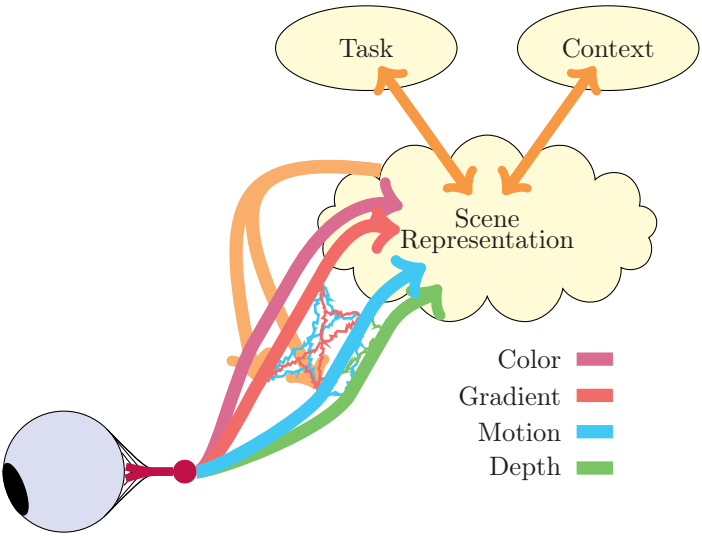

Figure B.1: Simplified overview of the visual system: The information coming from the eye can be roughly divided into four types of information: color, gradient, depth and motion. Based on these four channels a scene representation is build in a feed-forward process. The visual information contained in the scene representation, such as positions and attributes of objects is fed back into the feed-forward process to stabilize and enhance the detection. Task and context information further guide the recognition process. In this thesis only the motion and gradient channels and their interactions are analysed, while the feed-back loops are neglected.

visual recognition very *robust* to partial failures, as *e.g.* the loss of stereo vision due to a loss of an eye. An example for this redundancy and the partially complementary information is the extraction of form. Forms can be defined by similarities in color, depth, motion or by its borders defined in image gradients. Finding distinct forms can then help in segmentation and recognition. In addition, segmentation can help to calculate depth and motion information, *i.e.* due to feedback loops from higher visual processes the different information types can interact and their estimation can be improved. A simplified model of such a visual system is illustrated in fig. B.1.

C Gradient Derivations

C.1 Translation Invariant Learning

$$E_{\text{rec}} = \frac{1}{2}\sum_{n} \|\boldsymbol{V}_n - \boldsymbol{R}_n\|_2^2 \tag{C.1}$$

$$= \frac{1}{2}\sum_{n,\boldsymbol{x}} \Big(v_n(\boldsymbol{x}) - \sum_{j,\boldsymbol{m}} h_{jn}(\boldsymbol{m})\bar{w}_j(\boldsymbol{x}-\boldsymbol{m})\Big)^2 \tag{C.2}$$

Gradients for the activities:

$$\nabla_{h_{jn}(\boldsymbol{m})} E_{\text{rec}} = \sum_{\boldsymbol{x}} \Big(- \bar{w}_j(\boldsymbol{x}-\boldsymbol{m})(v_n(\boldsymbol{x}) - r_n(\boldsymbol{x}))\Big) \tag{C.3}$$

$$= \sum_{\boldsymbol{x}} r_n(\boldsymbol{x})\bar{w}_j(\boldsymbol{x}-\boldsymbol{m}) - \sum_{\boldsymbol{x}} v_n(\boldsymbol{x})\bar{w}_j(\boldsymbol{x}-\boldsymbol{m}) \tag{C.4}$$

$$\nabla_{\boldsymbol{H}_{jn}} E_{\text{rec}} = \underbrace{\text{corr}_2(\boldsymbol{R}_n, \bar{\boldsymbol{W}}_j)}_{:=\left(\nabla_{\boldsymbol{H}_{jn}} E_{\text{rec}}\right)^+} - \underbrace{\text{corr}_2(\boldsymbol{V}_n, \bar{\boldsymbol{W}}_j)}_{:=\left(\nabla_{\boldsymbol{H}_{jn}} E_{\text{rec}}\right)^-} \tag{C.5}$$

Gradients for the basis vectors, with the substitution $\boldsymbol{x}' = \boldsymbol{x} - \boldsymbol{m}$:

$$\nabla_{\bar{w}_j(\boldsymbol{x}')} E_{\text{rec}} = \sum_{n,\boldsymbol{x}} \Big(- h_{jn}(\boldsymbol{x}-\boldsymbol{x}')(v_n(\boldsymbol{x}) - r_n(\boldsymbol{x}))\Big) \tag{C.6}$$

$$= \sum_{n,\boldsymbol{x}} r_n(\boldsymbol{x})h_{jn}(\boldsymbol{x}-\boldsymbol{x}') - \sum_{n,\boldsymbol{x}} v_n(\boldsymbol{x})h_{jn}(\boldsymbol{x}-\boldsymbol{x}') \tag{C.7}$$

$$\nabla_{\bar{\boldsymbol{W}}_j} E_{\text{rec}} = \underbrace{\sum_{n} \text{corr}_2(\boldsymbol{R}_n, \boldsymbol{H}_{jn})}_{:=\left(\nabla_{\bar{\boldsymbol{W}}_j} E_{\text{rec}}\right)^+} - \underbrace{\sum_{i} \text{corr}_2(\boldsymbol{V}_n, \boldsymbol{H}_{jn})}_{:=\left(\nabla_{\bar{\boldsymbol{W}}_j} E_{\text{rec}}\right)^-} \tag{C.8}$$

C.2 Topological Sparsity

$$E_p = \frac{1}{2} \sum_{n,j,\boldsymbol{m}} \boldsymbol{R}_{njm}^\top \Big(\boldsymbol{R}_n - \boldsymbol{R}_{njm} \Big) \tag{C.9}$$

$$= \underbrace{\frac{1}{2} \sum_{n,j,\boldsymbol{m}} \boldsymbol{R}_{njm}^\top \boldsymbol{R}_n}_{:=E_{p1}} - \underbrace{\frac{1}{2} \sum_{n,j,\boldsymbol{m}} \boldsymbol{R}_{njm}^\top \boldsymbol{R}_{njm}}_{:=E_{p2}} \tag{C.10}$$

$$E_{p1} = \frac{1}{2} \sum_{n,\boldsymbol{x}} \Big(\sum_{j,\boldsymbol{m}} \bar{w}_j(\boldsymbol{x}-\boldsymbol{m}) h_{jn}(\boldsymbol{m}) \Big)^2 \tag{C.11}$$

$$E_{p2} = \frac{1}{2} \sum_{n,\boldsymbol{x},j,\boldsymbol{m}} \bar{w}_j^2(\boldsymbol{x}-\boldsymbol{m}) h_{jn}^2(\boldsymbol{m}) \tag{C.12}$$

Gradients for the basis vectors, with the substitution $\boldsymbol{x}' = \boldsymbol{x} - \boldsymbol{m}$:

$$\nabla_{\bar{w}_j(\boldsymbol{x}')} E_{p1} = \sum_{n,\boldsymbol{x}} h_{jn}(\boldsymbol{x}-\boldsymbol{x}') r_n(\boldsymbol{x}) \tag{C.13}$$

$$\nabla_{\bar{\boldsymbol{W}}_j} E_{p1} = \sum_n \operatorname{corr}_2(\boldsymbol{R}_n, \boldsymbol{H}_{jn}) \tag{C.14}$$

$$\nabla_{\bar{w}_j(\boldsymbol{x}')} E_{p2} = \sum_n \bar{w}_j(\boldsymbol{x}') \sum_{\boldsymbol{x}} h_{jn}^2(\boldsymbol{x}-\boldsymbol{x}') \tag{C.15}$$

$$\nabla_{\bar{\boldsymbol{W}}_j} E_{p2} = \bar{\boldsymbol{W}}_j \sum_n \boldsymbol{H}_{jn}^\top \boldsymbol{H}_{jn} \tag{C.16}$$

$$\nabla_{\bar{\boldsymbol{W}}_j} E_p = \nabla_{\bar{\boldsymbol{W}}_j} E_{p1} - \nabla_{\bar{\boldsymbol{W}}_j} E_{p2} \tag{C.17}$$

$$= \sum_n \operatorname{corr}_2(\boldsymbol{R}_n, \boldsymbol{H}_{jn}) - \bar{\boldsymbol{W}}_j \sum_n \boldsymbol{H}_{jn}^\top \boldsymbol{H}_{jn} \tag{C.18}$$

Gradient for the activities:

$$\nabla_{h_{jn}(\boldsymbol{m})} E_{p1} = \sum_{\boldsymbol{x}} \bar{w}_j(\boldsymbol{x}-\boldsymbol{m}) r_n(\boldsymbol{x}) \tag{C.19}$$

$$\nabla_{\boldsymbol{H}_{jn}(\boldsymbol{m})} E_{p1} = \operatorname{corr}_2(\boldsymbol{R}_n, \bar{\boldsymbol{W}}_j) \tag{C.20}$$

$$\nabla_{h_{jn}(\boldsymbol{m})} E_{p2} = h_{jn}(\boldsymbol{m}) \sum_{\boldsymbol{x}} \bar{w}_j^2(\boldsymbol{x}-\boldsymbol{m}) \tag{C.21}$$

$$\nabla_{\boldsymbol{H}_{jn}} E_{p2} = \boldsymbol{H}_{jn} \bar{\boldsymbol{W}}_j^\top \bar{\boldsymbol{W}}_j \tag{C.22}$$

$$\nabla_{\boldsymbol{H}_{jn}} E_p = \nabla_{\boldsymbol{H}_{jn}} E_{p1} - \nabla_{\boldsymbol{H}_{jn}} E_{p2} \tag{C.23}$$

$$= \operatorname{corr}_2(\boldsymbol{R}_n, \bar{\boldsymbol{W}}_j) - \boldsymbol{H}_{jn} \bar{\boldsymbol{W}}_j^\top \bar{\boldsymbol{W}}_j \tag{C.24}$$

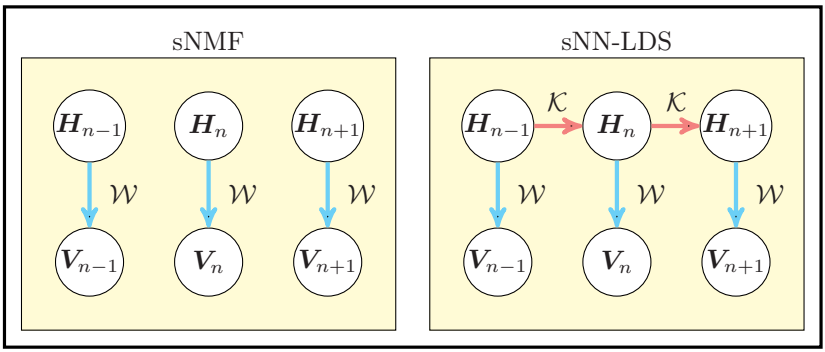

Figure D.1: The activations of a sNMF depicted on the left share a common set of basis vectors $\mathcal{W}$ for all inputs $\mathcal{V}$, but are conditionally independent. The activations of a sNN-LDS depicted on the right share a common set of basis vectors $\mathcal{W}$ and the temporal transitions between the activations are defined by the common transition matrix $\mathcal{K}$.

D Sparse Non-Negative Linear Dynamic Systems

D.1 Temporal Extension of sNMF

As introduced in section 3.3.5, the sNMF model does not contain any specific relations between consecutive input data $\boldsymbol{V}_n$ and $\boldsymbol{V}_{n+1}$. The underlying assumption is that they are conditionally independent and only coupled due to a common basis vector set $\bar{\mathcal{W}}$. However, this assumption might not hold, *e.g.* for human actions, where the actions can be defined by consecutive poses.

By adding transitions between the activations, the *static* sNMF model can be extended to a *dynamic* model. The relation between each consecutive inputs can be modeled by a linear dependency between the corresponding

activations $\boldsymbol{H}_n$ and $\boldsymbol{H}_{n+1}$. The result is a *sparse non-negative linear dynamic system* (sNN-LDS). The two systems are illustrated in fig. D.1. The temporal relation between the single activations are

$$h_{jn+1} = \sum_l k_{jl} h_{ln}, \tag{D.1}$$

with a transition parameter k_{jl}. Written in matrix form

$$\mathcal{H}\mathcal{Q} = \mathcal{K}\mathcal{H}\mathcal{S}, \tag{D.2}$$

with the shift matrix

$$\mathcal{S} = \begin{pmatrix} 0 & \mathcal{I} \\ 0 & 0 \end{pmatrix}, \tag{D.3}$$

$\mathcal{S} \in \mathbb{R}^{N \times N}$ and the matrix

$$\mathcal{Q} = \begin{pmatrix} 0 & 0 \\ 0 & \mathcal{I} \end{pmatrix}, \tag{D.4}$$

$\mathcal{Q} \in \mathbb{R}^{N \times N}$, that masks out the first input $\boldsymbol{V}_1$. $\mathcal{K} \in \mathbb{R}^{J \times J}$ is the transition matrix, that contains only non-negative elements, *i.e.*

$$k_{jl} \geq 0, \quad \forall j, l \in [1, ..., J]. \tag{D.5}$$

If multiple videos are used to learn the parameters of the sNN-LDS, the matrices $\mathcal{Q}$ and $\mathcal{S}$ are modified accordingly. The activation $\boldsymbol{H}_{n+1}$ is now influenced by the corresponding input $\boldsymbol{V}_{n+1}$ and the previous activation $\boldsymbol{H}_n$. The sNN-LDS is defined by the transition matrix $\mathcal{K}$ and the observation matrix or basis vectors $\bar{\mathcal{W}}$.

It is noteworthy, that since both the observation and transition models are linear, the model can be be consider as a discrete *linear dynamical system* (LDS) with non-negativity constraints. Additional sparsity constraints make it a sNN-LDS as illustrated in fig. D.2.

D.2 Related Work

There already exist several temporal extensions for NMF or SC algorithms, *e.g.* the *fussed lasso* [102] or [12]. However, these algorithms do not learn the transitions between the activations, but rather enforce a strict temporal consistancy by penalizing differences between consecutive activations.

In speech recognition, there exist hidden markov models learned with NMF-like algorithms [20, 74]. Again, the transitions are not learned, but rather given via prior knowledge into the model.

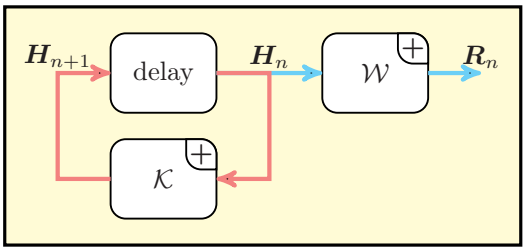

Figure D.2: The sNN-LDS depicted as a discrete linear dynamic system. The components marked with a (+) are strictly non-negative.

D.3 Transition Energy

In addition to the reconstruction energy that is used to learn the basis vectors $\bar{\mathcal{W}}$ of a sNMF model, the sNN-LDS requires an energy function that reflects the influence of the transition matrix $\mathcal{K}$. The reconstruction energy scales relative to the input, so it is beneficial for the parameter settings if the transition energy scales in the same manner. Thus, the estimated activation $\mathcal{KHS}$ is projected onto the input space by the basis vectors $\bar{\mathcal{W}}$. The transition energy function is

$$E_\mathrm{t} = \lambda_\mathrm{t} \frac{1}{2} \|\mathcal{VQ} - \bar{\mathcal{W}}\mathcal{KHS}\|_F^2. \tag{D.6}$$

The gradients for the activations are

$$(\nabla_{\mathcal{H}} E_\mathrm{t})^+ = \lambda_\mathrm{t} \mathcal{K}^\top \bar{\mathcal{W}}^\top \bar{\mathcal{W}} \mathcal{KHSS}^\top, \tag{D.7}$$
$$(\nabla_{\mathcal{H}} E_\mathrm{t})^- = \lambda_\mathrm{t} \mathcal{K}^\top \bar{\mathcal{W}}^\top \mathcal{VS}^\top, \tag{D.8}$$

and for the basis vectors

$$(\nabla_{\bar{\mathcal{W}}} E_\mathrm{t})^+ = \lambda_\mathrm{t} \bar{\mathcal{W}} \mathcal{KHSS}^\top \mathcal{H}^\top \mathcal{K}^\top, \tag{D.9}$$
$$(\nabla_{\bar{\mathcal{W}}} E_\mathrm{t})^- = \lambda_\mathrm{t} \mathcal{VQS}^\top \mathcal{H}^\top \mathcal{K}^\top, \tag{D.10}$$

and the transition matrix

$$(\nabla_{\mathcal{K}} E)^+ = \lambda_\mathrm{t} \bar{\mathcal{W}}^\top \bar{\mathcal{W}} \mathcal{KHSS}^\top \mathcal{H}^\top, \tag{D.11}$$
$$(\nabla_{\mathcal{K}} E)^- = \lambda_\mathrm{t} \bar{\mathcal{W}}^\top \mathcal{VQS}^\top \mathcal{H}^\top. \tag{D.12}$$

D.3.1 Sparsity in the Transitions

To reduce the amount of transitions an additional energy function that enforces sparse transitions is added. The energy function is

$$E_{\mathrm{k}} = \lambda_{\mathrm{k}} \sum_{j,l} k_{jl}, \tag{D.13}$$

with the gradient for the transitions

$$(\nabla_{k_{jl}} E_{\mathrm{k}})^{+} = \lambda_{\mathrm{k}}. \tag{D.14}$$

D.3.2 sNN-LDS Learning Algorithm

The overall energy function for learning the sNN-LDS is

$$\begin{aligned} E_{\text{sNN-LDS}} &= E_{\mathrm{rec}} + \lambda_{\mathrm{h}} E_{\mathrm{h}} + \lambda_p E_p + \lambda_{\mathrm{t}} E_{\mathrm{t}} + \lambda_{\mathrm{k}} E_{\mathrm{k}} \\ &= \frac{1}{2}\|\mathcal{V} - \mathcal{R}\|_F^2 + \lambda_{\mathrm{h}} \sum_{n,j} h_{jn} + \frac{1}{2}\lambda_{\mathrm{part}} \sum_{n,j} \left(\boldsymbol{R}_{jn}^{\top} \sum_{k \neq j} \boldsymbol{R}_{kn}\right) \\ &\quad + \lambda_{\mathrm{t}} \frac{1}{2}\|\mathcal{V}\mathcal{Q} - \bar{\mathcal{W}}\mathcal{K}\mathcal{H}\mathcal{S}\|_F^2 + \lambda_{\mathrm{k}} \sum_{j,l} k_{jl}. \end{aligned} \tag{D.15}$$

The gradients can be derived from the equations (3.34), (3.35), (3.36), (3.37), (3.48), (3.54), (D.7), (D.8), (D.9), (D.10), (D.11), (D.12) and (D.14). The algorithm to learn the model parameters $\mathcal{H}$, $\mathcal{W}$ and $\mathcal{K}$ of a sNN-LDS is:

- Preprocessing
  - Normalize $\mathcal{V} = \frac{\mathcal{V}}{\max(\mathcal{V})}$,
  - initialize $\mathcal{H}$, $\mathcal{W}$ and $\mathcal{K}$ randomly.
- Loop for i iterations
  1. Calculate $\mathcal{R} = \bar{\mathcal{W}}\mathcal{H}$,
  2. update $\mathcal{H} \rightarrow \mathcal{H} \circ \frac{(\nabla_{\mathcal{H}} E_{\text{sNN-LDS}})^{-}}{(\nabla_{\mathcal{H}} E_{\text{sNN-LDS}})^{+}}$,
  3. calculate $\mathcal{R} = \bar{\mathcal{W}}\mathcal{H}$,
  4. update $\mathcal{W} \rightarrow \mathcal{W} \circ \frac{(\nabla_{\mathcal{W}} E_{\text{sNN-LDS}})^{-}}{(\nabla_{\mathcal{W}} E_{\text{sNN-LDS}})^{+}}$,
  5. normalize $\bar{\boldsymbol{W}}_j = \frac{\boldsymbol{W}_j}{\sqrt{\sum_p w_{pj}^2}}, \quad \forall j \in [1, ..., J]$,

6. update $\mathcal{K} \rightarrow \mathcal{K} \circ \frac{(\nabla_{\mathcal{K}} E_{\text{sNN-LDS}})^-}{(\nabla_{\mathcal{K}} E_{\text{sNN-LDS}})^+}$.

The default settings for the energy parameters are $\lambda_{\text{h}} = 0.1$, $\lambda_p = 0.2$, $\lambda_{\text{t}} = 0.2$ and $\lambda_{\text{k}} = 0.1$.

D.4 Results

Classification results of the sNN-LDS algorithm for varying number of basis vectors J for human action recognition are discussed in [45]. Here the sNN-LDS is applied as an additional layer on top of the pooled features and it is compared to a static sNMF algorithm with the same amount of basis vectors. The VNMF layer consists of 8 basis vectors for the optical flow and 8 basis vectors for the gradients. The overall system is illustrated in fig. D.3.

Table D.1: Classification results for leave-one-out experiments.

	sNMF		sNN-LDS	
J	50	100	50	100
Weizmann	0.98	0.99	0.99	**1.00**
UCF-Sports	0.88	0.87	0.90	**0.92**

Table D.1 shows the results for the different experiments on the UCF-Sports and Weizmann dataset. The sNN-LDS slighty outperforms the sNMF on the Weizmann datasets and by 2% on the UCF-Sports datasets, which shows that modeling the temporal relations improves the classification performance. Increasing the number of basis vectors J from 50 to 100 improves the results, but not as significantly as adding the temporal relations.

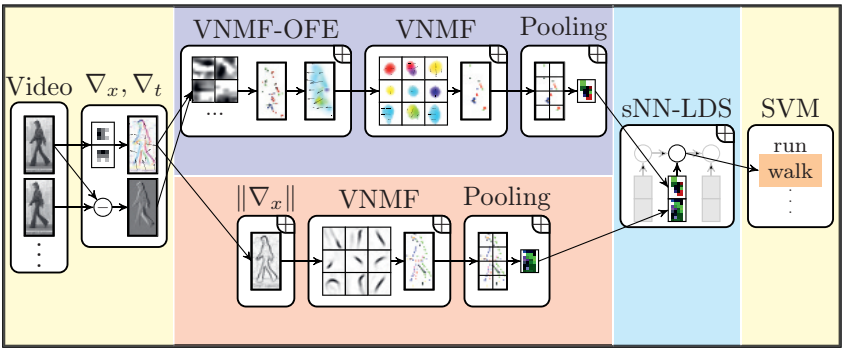

Figure D.3: Overview of the extended two stream hierarchical biological motion recognition system. At first, the spatial and temporal gradients of the incoming video data are calculated. In the motion processing stream (coloured blue), the spatial and temporal gradients are used to estimate the optical flow (VNMF-OFE) which is thereafter matched onto a set of prelearned optical flow patterns via the VNMF algorithm. In the gradient processing stream (coloured red) the spatial gradient amplitudes are calculated and matched onto a set of prelearned gradient patterns with the VNMF algorithm. The spatially pooled pattern responses are then given as input for a sNN-LDS layer (coloured cyan). The activations are classified in the final layer that consists of a Support Vector Machine (SVM). Each box with a (+) has a strictly non-negative representation.

Bibliography

[1] Aggarwal, J. and Ryoo, M. S. (2011). Human activity analysis: A review. *ACM Computing Surveys (CSUR)*, 43(3):16.

[2] Amiri, S. M., Nasiopoulos, P., and Leung, V. (2012). Non-negative sparse coding for human action recognition. In *Image Processing (ICIP), 2012 19th IEEE International Conference on*, pages 1421–1424. IEEE.

[3] Anandan, P. (1989). A computational framework and an algorithm for the measurement of visual motion. *International Journal of Computer Vision*, 2(3):283–310.

[4] Beintema, J. and Lappe, M. (2002). Perception of biological motion without local image motion. *Proceedings of the National Academy of Sciences*, 99(8):5661–5663.

[5] Bilinski, P. and Bremond, F. (2012). Contextual statistics of space-time ordered features for human action recognition. In *Advanced Video and Signal-Based Surveillance (AVSS), 2012 IEEE Ninth International Conference on*, pages 228–233. IEEE.

[6] Bishop, C. M. et al. (2006). *Pattern recognition and machine learning*, volume 1. springer New York.

[7] Blake, R. and Shiffrar, M. (2007). Perception of human motion. *Annu. Rev. Psychol.*, 58:47–73.

[8] Blank, M., Gorelick, L., Shechtman, E., Irani, M., and Basri, R. (2005). Actions as space-time shapes. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, volume 2, pages 1395–1402. IEEE.

[9] Borst, A. and Euler, T. (2011). Seeing things in motion: models, circuits, and mechanisms. *Neuron*, 71(6):974–994.

[10] Bremmer, F., Kubischik, M., Pekel, M., Hoffmann, K.-P., and Lappe, M. (2010). Visual selectivity for heading in monkey area mst. *Experimental brain research*, 200(1):51–60.

[11] Bruhn, A., Weickert, J., and Schnörr, C. (2005). Lucas/kanade meets horn/schunck: Combining local and global optic flow methods. *International Journal of Computer Vision*, 61(3):211–231.

[12] Cadieu, C. F. and Olshausen, B. A. (2012). Learning intermediate-level representations of form and motion from natural movies. *Neural computation*, 24(4):827–866.

[13] Casile, A. and Giese, M. A. (2005). Critical features for the recognition of biological motion. *Journal of vision*, 5(4).

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

[15] Chen, Y., Zhao, Y., and Cai, A. (2013). Recognizing human actions based on sparse coding with non-negative and locality constraints. In *Visual Communications and Image Processing (VCIP), 2013*, pages 1–6. IEEE.

[16] Chessa, M., Solari, F., Sabatini, S. P., and Bisio, G. M. (2008). Motion interpretation using adjustable linear models. In *BMVC*, pages 1–10.

[17] Choi, S. (2008). Algorithms for orthogonal nonnegative matrix factorization. In *Neural Networks, 2008. IJCNN 2008.(IEEE World Congress on Computational Intelligence). IEEE International Joint Conference on*, pages 1828–1832. IEEE.

[18] Chouchourelou, A., Golden, A., Shiffrar, M., and Chouchourelou, A. (2011). What does biological motion really mean? differentiating visual percepts of human, animal, and non-biological motions.

[19] Cichocki, A., Zdunek, R., Phan, A. H., and Amari, S.-i. (2009). *Nonnegative matrix and tensor factorizations: applications to exploratory multi-way data analysis and blind source separation*. Wiley. com.

[20] Cybenko, G. and Crespi, V. (2011). Learning hidden markov models using nonnegative matrix factorization. *Information Theory, IEEE Transactions on*, 57(6):3963–3970.

[21] Dalal, N., Triggs, B., and Schmid, C. (2006). Human detection using oriented histograms of flow and appearance. In *Computer Vision–ECCV 2006*, pages 428–441. Springer.

[22] Dean, T., Washington, R., and Corrado, G. (2010). Sparse spatiotemporal coding for activity recognition.

[23] DiCarlo, J. J., Zoccolan, D., and Rust, N. C. (2012). How does the brain solve visual object recognition? *Neuron*, 73(3):415–434.

[24] Ding, C. H., Li, T., and Jordan, M. I. (2010). Convex and semi-nonnegative matrix factorizations. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(1):45–55.

[25] Dollar, P., Rabaud, V., Cottrell, G., and Belongie, S. (2005). Behavior recognition via sparse spatio-temporal features. In *Visual Surveillance and Performance Evaluation of Tracking and Surveillance, 2005. 2nd Joint IEEE International Workshop on*, pages 65–72. IEEE.

[26] Donoho, D. and Stodden, V. (2003). When does non-negative matrix factorization give a correct decomposition into parts? In *Advances in neural information processing systems*, page None.

[27] Eggert, J. and Korner, E. (2004). Sparse coding and nmf. In *Neural Networks, 2004. Proceedings. 2004 IEEE International Joint Conference on*, volume 4, pages 2529–2533. IEEE.

[28] Eggert, J., Wersing, H., and Korner, E. (2004). Transformation-invariant representation and nmf. In *Neural Networks, 2004. Proceedings. 2004 IEEE International Joint Conference on*, volume 4, pages 2535–2539. IEEE.

[29] Ekman, P. and Rosenberg, E. L. (1997). *What the face reveals: Basic and applied studies of spontaneous expression using the Facial Action Coding System (FACS)*. Oxford University Press.

[30] Exner, S. (1887). Einige beobachtungen Ã¼ber bewegungsnachbilder. *Centralblatt fÃ¼r Physiologie*.

[31] Fleet, D. J., Black, M. J., Yacoob, Y., and Jepson, A. D. (2000). Design and use of linear models for image motion analysis. *International Journal of Computer Vision*, 36(3):171–193.

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

[33] Fukushima, K. (1980). Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological cybernetics*, 36(4):193–202.

[34] Furl, N., Hadj-Bouziane, F., Liu, N., Averbeck, B. B., and Ungerleider, L. G. (2012). Dynamic and static facial expressions decoded from motion-sensitive areas in the macaque monkey. *The Journal of Neuroscience*, 32(45):15952–15962.

[35] 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.

[36] Geiger, A., Lenz, P., and Urtasun, R. (2012). Are we ready for autonomous driving? the kitti vision benchmark suite. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 3354–3361. IEEE.

[37] Giese, M. A. (2014). Biological and body motion perception. *Oxford Handbook of Perceptual Organization*.

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

[39] Grossman, E. D. and Blake, R. (2002). Brain areas active during visual perception of biological motion. *Neuron*, 35(6):1167–1175.

[40] Grossman, E. D., Jardine, N. L., and Pyles, J. A. (2010). fmr-adaptation reveals invariant coding of biological motion on the human sts. *Frontiers in Human Neuroscience*, 4.

[41] Guha, T. and Ward, R. K. (2012). Learning sparse representations for human action recognition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(8):1576–1588.

[42] Guthier, T., Eggert, J., and Willert, V. (2012a). Unsupervised learning of motion patterns. In *European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*, volume 20, pages 323–328, Bruges.

[43] Guthier, T., Gerges, S., Willert, V., and Eggert, J. (2013a). Learning associative spatiotemporal features with non-negative sparse coding.

In *European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*, Bruges.

[44] Guthier, T., Sosic, A., Willert, V., and Eggert, J. (2012b). Finding a tradeoff between compression and loss in motion compensated video coding. In *SIGMAP*, pages 81–84.

[45] Guthier, T., Šošić, A., Willert, V., and Eggert, J. (2014a). snn-lds: Spatio-temporal non-negative sparse coding for human action recognition. In *Artificial Neural Networks, 2014. Proceedings. 2014 IEEE International Conference on*.

[46] Guthier, T., Willert, V., and Eggert, J. (2014b). Beyond histograms: Why learned structure-preserving descriptors outperform hog. In *European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*, volume 22, Bruges.

[47] Guthier, T., Willert, V., and Eggert, J. (2015). Topological sparse learning of dynamic form patterns. *Neural Computation*.

[48] Guthier, T., Willert, V., Schnall, A., Kreuter, K., and Eggert, J. (2013b). Non-negative sparse coding for motion extraction. In *Neural Networks, 2013. Proceedings. 2013 IEEE International Joint Conference on*.

[49] Haag, J. and Borst, A. (2004). Neural mechanism underlying complex receptive field properties of motion-sensitive interneurons. *Nature neuroscience*, 7(6):628–634.

[50] Horn, B. and Schunck, B. (1981). Determining optical flow. *Artificial Intelligence*, 16:185–203.

[51] Hoyer, P. O. (2002). Non-negative sparse coding. In *Neural Networks for Signal Processing, 2002. Proceedings of the 2002 12th IEEE Workshop on*, pages 557–565. IEEE.

[52] Hoyer, P. O. (2004). Non-negative matrix factorization with sparseness constraints. *The Journal of Machine Learning Research*, 5:1457–1469.

[53] Hubel, D. H. and Wiesel, T. N. (1962). Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *The Journal of physiology*, 160(1):106.

[54] Hyvearinen, A., Hurri, J., and Hoyer, P. O. (2009). *Natural Image Statistics*, volume 39. Springer.

[55] Jellema, T. and Perrett, D. I. (2003). Cells in monkey sts responsive to articulated body motions and consequent static posture: a case of implied motion? *Neuropsychologia*, 41(13):1728–1737.

[56] Jhuang, H., Serre, T., Wolf, L., and Poggio, T. (2007). A biologically inspired system for action recognition. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8. IEEE.

[57] Jia, K., Wang, W. X., and Tang, X. (2011). Optical flow estimation using learned sparse model. *IEEE Conf. on Computer Vision (ICCV)*, pages 2391–2398.

[58] Johansson, G. (1973). Visual perception of biological motion and a model for its analysis. *Perception & psychophysics*, 14(2):201–211.

[59] Johnson, K. and Shiffrar, M. (2013). *People Watching: Social, Perceptual, and Neurophysiological Studies of Body Perception*. Oxford University Press.

[60] Klaser, A., Marszałek, M., Laptev, I., Schmid, C., et al. (2010). Will person detection help bag-of-features action recognition?

[61] Kuehne, H., Jhuang, H., Garrote, E., Poggio, T., and Serre, T. (2011). Hmdb: a large video database for human motion recognition. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 2556–2563. IEEE.

[62] Lange, J. and Lappe, M. (2006). A model of biological motion perception from configural form cues. *The Journal of Neuroscience*, 26(11):2894–2906.

[63] Laurberg, H. (2007). Uniqueness of non-negative matrix factorization. In *Statistical Signal Processing, 2007. SSP'07. IEEE/SP 14th Workshop on*, pages 44–48. IEEE.

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

[65] LeCun, Y. (2012). Learning invariant feature hierarchies. In *Computer vision–ECCV 2012. Workshops and demonstrations*, pages 496–505. Springer.

[66] Lee, D. D. and Seung, H. S. (1999). Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791.

[67] Lee, D. D. and Seung, H. S. (2000). Algorithms for non-negative matrix factorization. In *Advances in neural information processing systems*, pages 556–562.

[68] Li, P. S., Givoni, I. E., and Frey, B. J. (2011). Learning better image representations using flobject analysis. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 2721–2728. IEEE.

[69] Li, S. Z., Hou, X. W., Zhang, H. J., and Cheng, Q. S. (2001). Learning spatially localized, parts-based representation. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, volume 1, pages I–207. IEEE.

[70] Lowe, D. G. (1999). Object recognition from local scale-invariant features. In *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*, volume 2, pages 1150–1157. Ieee.

[71] Lucas, B. and Kanade, I. (1981). An iterative image registration technique with an application to stereo vision. *DARPA lmage Understanding Workshop*, pages 121–130.

[72] Mairal, J., Bach, F., Ponce, J., and Sapiro, G. (2010). Online learning for matrix factorization and sparse coding. *The Journal of Machine Learning Research*, 11:19–60.

[73] Minsky, M. and Papert, S. (1969). Perceptrons.

[74] Nakano, M., Le Roux, J., Kameoka, H., Kitano, Y., Ono, N., and Sagayama, S. (2010). Nonnegative matrix factorization with markov-chained bases for modeling time-varying patterns in music spectrograms. In *Latent Variable Analysis and Signal Separation*, pages 149–156. Springer.

[75] Nieuwenhuis, C., Kondermann, D., and Garbe, C. S. (2010). Complex motion models for simple optical flow estimation. In *Proc. of the 32nd DAGM conf. on Pattern recognition*.

[76] Oja, E. (2002). Unsupervised learning in neural computation. *Theoretical computer science*, 287(1):187–207.

[77] Olshausen, B. A. and Field, D. J. (1997). Sparse coding with an overcomplete basis set: A strategy employed by v1? *Vision research*, 37(23):3311–3325.

[78] Paatero, P. and Tapper, U. (1994). Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values. *Environmetrics*, 5(2):111–126.

[79] Pitzalis, S., Sdoia, S., Bultrini, A., Committeri, G., Di Russo, F., Fattori, P., Galletti, C., and Galati, G. (2013). Selectivity to translational egomotion in human brain motion areas. *PloS one*, 8(4):e60241.

[80] Potluru, V. K. (2011). Understanding and exploiting the connections between nmf and svm. In *Data Mining Workshops (ICDMW), 2011 IEEE 11th International Conference on*, pages 1207–1210. IEEE.

[81] Potluru, V. K., Plis, S. M., Morup, M., Calhoun, V. D., and Lane, T. (2009). Efficient multiplicative updates for support vector machines. In *SDM*, pages 1218–1229.

[82] Puce, A. and Perrett, D. (2003). Electrophysiology and brain imaging of biological motion. *Philosophical Transactions of the Royal Society of London. Series B: Biological Sciences*, 358(1431):435–445.

[83] Pyles, J. A., Garcia, J. O., Hoffman, D. D., and Grossman, E. D. (2007). Visual perception and neural correlates of novel biological motion. *Vision Research*, 47(21):2786–2797.

[84] Rebhan, S., Eggert, J., Grossman, E. D., H.-M., and Koerner, E. (2007). Sparse and transformation-invariant hierarchical nmf. In *Artificial Neural Networks–ICANN 2007*, pages 894–903. Springer.

[85] Rebhan, S., Sharif, W., and Eggert, J. (2009). Incremental learning in the non-negative matrix factorization. In *Advances in Neuro-Information Processing*, pages 960–969. Springer.

[86] Reppas, J. B., Niyogi, S., Dale, A. M., Sereno, M. I., and Tootell, R. B. (1997). Representation of motion boundaries in retinotopic human visual cortical areas. *Nature*, 388(6638):175–179.

[87] Riesenhuber, M. and Poggio, T. (1999). Hierarchical models of object recognition in cortex. *Nature neuroscience*, 2(11):1019–1025.

[88] Rodriguez, M., Ahmed, J., and Shah, M. (2008). Action mach a spatio-temporal maximum average correlation height filter for action recognition. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8.

[89] Rosenblatt, F. (1958). The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386.

[90] Rozell, C. J., Johnson, D. H., Baraniuk, R. G., and Olshausen, B. A. (2008). Sparse coding via thresholding and local competition in neural circuits. *Neural computation*, 20(10):2526–2563.

[91] Schachtner, R., Pöppel, G., Tomé, A. M., and Lang, E. W. (2009). Minimum determinant constraint for non-negative matrix factorization. In *Independent Component Analysis and Signal Separation*, pages 106–113. Springer.

[92] Schuldt, C., Laptev, I., and Caputo, B. (2004). Recognizing human actions: a local svm approach. In *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, volume 3, pages 32–36. IEEE.

[93] Servos, P., Osu, R., Santi, A., and Kawato, M. (2002). The neural substrates of biological motion perception: an fmri study. *Cerebral Cortex*, 12(7):772–782.

[94] Sivic, J. and Zisserman, A. (2003). Video google: A text retrieval approach to object matching in videos. In *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, pages 1470–1477. IEEE.

[95] Smith, A. T. and Snowden, R. J. (1994). Motion detection: an overview. *Visual Detection of Motion Academic, London pp3-13*.

[96] Soomro, K., Zamir, A. R., and Shah, M. (2012). Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*.

[97] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958.

[98] Sun, D., Roth, S., and Black, M. J. (2010). Secrets of optical flow estimation and their principles. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 2432–2439. IEEE.

[99] Sun, D., Sudderth, E. B., and Black, M. J. (2012). Layered segmentation and optical flow estimation over time. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 1768–1775. IEEE.

[100] Theusner, S., de Lussanet, M., and Lappe, M. (2014). Action recognition by motion detection in posture space. *The Journal of Neuroscience*, 34(3):909–921.

[101] Thompson, J. C. and Baccus, W. (2012). Form and motion make independent contributions to the response to biological motion in occipitotemporal cortex. *NeuroImage*, 59(1):625–634.

[102] Tibshirani, R., Saunders, M., Rosset, S., Zhu, J., and Knight, K. (2005). Sparsity and smoothness via the fused lasso. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(1):91–108.

[103] Tohyama, K. and Fukushima, K. (2005). Neural network model for extracting optic flow. *Neural Networks*, 18(5):549–556.

[104] Valstar, M. and Pantic, M. (2010). Induced disgust, happiness and surprise: an addition to the mmi facial expression database. In *Proc. Int. Conf. Language Resources and Evaluation, Workshop on EMOTION*, pages 65–70.

[105] Vapnik, V. N. and Chervonenkis, A. J. (1974). Theory of pattern recognition.

[106] Wandell, B., Dumoulin, S. O., and Brewer, A. A. (2008). Visual cortex in humans. *Encyclopedia of neuroscience*, 10:251–257.

[107] Wang, H., Kläser, A., Schmid, C., and Liu, C.-L. (2013). Dense trajectories and motion boundary descriptors for action recognition. *International journal of computer vision*, 103(1):60–79.

[108] Wang, H., Ullah, M. M., Klaser, A., Laptev, I., Schmid, C., et al. (2009). Evaluation of local spatio-temporal features for action recognition. In *BMVC 2009-British Machine Vision Conference*.

[109] Wang, J., Chen, Z., and Wu, Y. (2011). Action recognition with multiscale spatio-temporal contexts. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 3185–3192. IEEE.

[110] Weiner, K. S. and Grill-Spector, K. (2011). Not one extrastriate body area: Using anatomical landmarks, hmt+, and visual field maps to parcellate limb-selective activations in human lateral occipitotemporal cortex. *Neuroimage*, 56(4):2183–2199.

[111] Wersing, H. and Koerner, E. (2003). Learning optimized features for hierarchical models of invariant object recognition. *Neural computation*, 15(7):1559–1588.

[112] Willert, V. and Eggert, J. (2009). A stochastic dynamical system for optical flow estimation. In *Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on*, pages 711–718. IEEE.

[113] Willert, V. and Eggert, J. (2011). Modeling short-term adaptation processes of visual motion detectors. *Neurocomputing*, 74(9):1329–1339.

[114] Willert, V., Toussaint, M., Eggert, J., and Korner, E. (2007). Uncertainty optimization for robust dynamic optical flow estimation. In *Machine Learning and Applications, 2007. ICMLA 2007. Sixth International Conference on*, pages 450–457. IEEE.

Curriculum Vitae

Personal Information

Thomas Guthier

Born June, 16 1983 in Heppenheim, Germany

Academic Studies

02/2010–2015	PhD student, TU-Darmstadt, Control theory and robotics lab
02/2010–2014	Guest scientist, Honda Research Institute Europe
04/2009–10/2009	Diploma Thesis "'Analysis of the communication topology in networked multi-agent systems"'
10/2002–10/2009	Electrical Engineering (Automation), TU Darmstadt Qualifikation: Dipl.-Ing. Electrical Engineering
08/1994–07/2002	Starkenburg Gymnasium Heppenheim

Frankfurt am Main, 31.08.2016

Online-Buchshop für Ingenieure

VDI nachrichten **BUCHSHOP**

Online-Shops

Fachliteratur und mehr - jetzt bequem online recherchieren & bestellen unter: www.vdi-nachrichten.com/Der-Shop-im-Ueberblick

Täglich aktualisiert: Neuerscheinungen VDI-Schriftenreihen

Im Buchshop von vdi-nachrichten.com finden Ingenieure und Techniker ein speziell auf sie zugeschnittenes, umfassendes Literaturangebot.

Mit der komfortablen Schnellsuche werden Sie in den VDI-Schriftenreihen und im Verzeichnis lieferbarer Bücher unter 1.000.000 Titeln garantiert fündig.

Im Buchshop stehen für Sie bereit:

VDI-Berichte und die Reihe **Kunststofftechnik:**

Berichte nationaler und internationaler technischer Fachtagungen der VDI-Fachgliederungen

Fortschritt-Berichte VDI:

Dissertationen, Habilitationen und Forschungsberichte aus sämtlichen ingenieurwissenschaftlichen Fachrichtungen

Newsletter „Neuerscheinungen":

Kostenfreie Infos zu aktuellen Titeln der VDI-Schriftenreihen bequem per E-Mail

Autoren-Service:

Umfassende Betreuung bei der Veröffentlichung Ihrer Arbeit in der Reihe Fortschritt-Berichte VDI

Buch- und Medien-Service:

Beschaffung aller am Markt verfügbaren Zeitschriften, Zeitungen, Fortsetzungsreihen, Handbücher, Technische Regelwerke, elektronische Medien und vieles mehr – einzeln oder im Abo und mit weltweitem Lieferservice

Die Reihen der Fortschritt-Berichte VDI:

1 Konstruktionstechnik/Maschinenelemente
2 Fertigungstechnik
3 Verfahrenstechnik
4 Bauingenieurwesen
5 Grund- und Werkstoffe/Kunststoffe
6 Energietechnik
7 Strömungstechnik
8 Mess-, Steuerungs- und Regelungstechnik
9 Elektronik/Mikro- und Nanotechnik
10 Informatik/Kommunikation
11 Schwingungstechnik
12 Verkehrstechnik/Fahrzeugtechnik
13 Fördertechnik/Logistik
14 Landtechnik/Lebensmitteltechnik
15 Umwelttechnik
16 Technik und Wirtschaft
17 Biotechnik/Medizintechnik
18 Mechanik/Bruchmechanik
19 Wärmetechnik/Kältetechnik
20 Rechnerunterstützte Verfahren (CAD, CAM, CAE CAQ, CIM ...)
21 Elektrotechnik
22 Mensch-Maschine-Systeme
23 Technische Gebäudeausrüstung

ISBN 978-3-18-525108-5