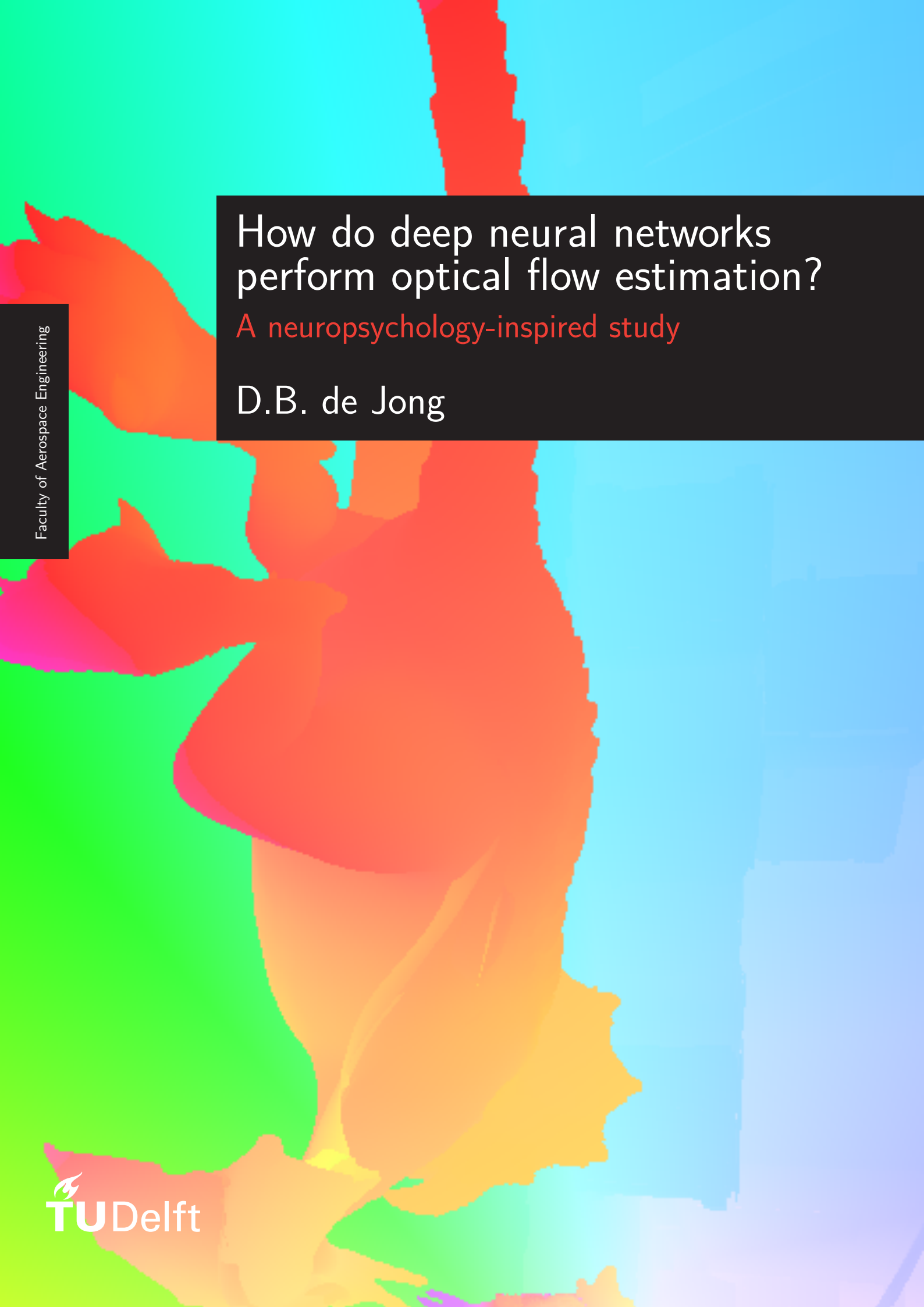


How do deep neural networks perform optical flow estimation?

A neuropsychology-inspired study

D.B. de Jong

Faculty of Aerospace Engineering

TU Delft

How do deep neural networks perform optical flow estimation?

A neuropsychology-inspired study

by

D.B. de Jong

to obtain the degree of Master of Science

at the Delft University of Technology,

Student number:	4291506	
Readers:	Prof. Dr G. C. H. E. de Croon,	TU Delft, supervisor
	Dr J. van Gemert,	TU Delft
	C. de Wagter MSc,	TU Delft
	F. Paredes-Vallés MSc,	TU Delft

An electronic version of this thesis is available at `http://repository.tudelft.nl/`.

TU Delft

Acknowledgements

First and foremost, I would like to thank my supervisors for their valuable guidance and support throughout my thesis. Specifically, I would like to thank Guido de Croon for teaching me how to think critically and not take things at face value. I would like to thank Federico Paredes Vallés for our many insightful discussions and valuable input into my work.

A word of appreciation goes out to my friends, roommates, and fellow graduate students for keeping up my spirit during the thesis phase.

Lastly, and most importantly, I would like to thank my family for their continued emotional support even during in difficult times which this thesis brought along as well. In fact, the following quote summarises some of my experiences during this time perfectly:

> *"All models are wrong, but some are useful"*
> — George E.P. Box

D.B. de Jong
Delft, April 2020

Abstract

End-to-end trained Convolutional Neural Networks have led to a breakthrough in optical flow estimation. The most recent advances focus on improving the optical flow estimation by improving the architecture and setting a new benchmark on the publicly available MPI-Sintel dataset. Instead, in this article, we investigate how deep neural networks estimate optical flow. By obtaining an understanding of how these networks function, more can be said about the behavior of these networks in unexpected scenarios and how the architecture and training data can be improved to obtain a better performance. For our investigation, we use a filter identification method that has played a major role in uncovering the motion filters present in animal brains in neuropsychological research. The method shows that the filters in deep neural networks are sensitive to a variety of motion patterns. Not only do we find translation filters, as demonstrated in animal brains, but thanks to the easier measurements in artificial neural networks, we even unveil dilation, rotation and occlusion filters. Furthermore, we find similarities in the refinement part of the network and the perceptual filling-in process which occurs in the mammal primary visual cortex.

Besides the research on the workings of Convolutional Neural Networks for optical flow estimation, this thesis also includes a literature-review of the main concepts related to this work. Furthermore, an extensive preliminary evaluation of Convolutional Neural Networks for different image sequences with optical ground truth can be found.

Contents

List of Symbols

Math symbols

$*$	Convolution operator
∇	Nabla operator
$\|\cdot\|_1$	L1 norm
$\|\cdot\|_2$	L2 norm

Greek symbols

α	Angle between planar surface and X-axis of observer
β	Angle between planar surface and Y-axis of observer
θ	Angle with respect to positive f_x axis
λ	Weight factor
τ	Time-to-contact
ϵ	Positive constant

Latin symbols

$\mathbf{c}$	RGB color
$\mathbf{c}_b$	RGB color body reflection component
$\mathbf{c}_i$	RGB color interface reflection component
$\mathbf{D}$	Optical flow divergence
d_x, d_y	Spatial shift
E	Energy term
e	Light source intensity
f	Focal length
F	Spatial frequency magnitude
f_t	Temporal frequency

f_x, f_y	Spatial frequencies
h	Distance between camera pinhole and planar surface
I	Image intensity
I_t	Temporal image intensity derivative
I_x, I_y	Spatial image intensity derivatives
$\mathcal{L}$	Cost function
m	Geometrical reflection factor
$\mathbf{n}$	Normal direction
O	Aperture of the camera in the pinhole camera mdoel
p, q, r	Rotational rates around X, Y, Z
s	Normal speed
t	Time
u, v	Optical flow components
U, V, W	Translational velocities along X, Y, Z
u_{GT}, v_{GT}	Ground truth optical flow components
u^R, v^R	Rotational optical flow components
u^T, v^T	Translational optical flow components
$\mathbf{v}$	Optical flow vector
$\mathbf{v}_n$	Component of motion normal to spatial contours
W	Window function
$\mathbf{x}$	Pixel location vector on the image plane
x, y	Pixel coordinates on the image plane
X, Y, Z	coordinates of the Cartesian coordinate system

List of Abbreviations

AAE Average Angular Error.

AE Angular Error.

AEE Average Endpoint Error.

CNN Convolutional Neural Network.

EE Endpoint Error.

FoC Focus of Contraction.

FoE Focus of Expansion.

HOG Histogram of Oriented Gradients.

ICA Independent Component Analysis.

MAV Micro Air Vehicle.

OFH Optic Flow in Harmony.

PCA Principal Component Analysis.

ReLU Rectified Linear Unit.

RNN Recurrent Neural Network.

SSD Sum-of-Squared Differences.

TTC Time-to-Contact.

List of Figures

List of Tables

1

Introduction

In the field of optical flow generally two types of motion fields are discerned. The *apparent motion* and the *motion field*. The former refers to the apparent motion of brightness patterns in the image and the latter to the 2D projection of the 3D motion of surfaces in the world. The apparent motion field is used for frame interpolation to enable video compression. Whereas the motion field has applications such as object tracking, navigation and visual odometry (the estimation of ego-motion of the observer using sequential images) for robotics including Micro Air Vehicles (MAVs). Variational approaches (Chapter 3) have dominated optical flow estimation ever since the pioneering work of B. K. Horn and Schunck, 1981. Many improvements have been introduced since (Brox, Bruhn, Papenberg & Weickert, 2004; Zimmer, Bruhn & Weickert, 2011).

Several machine learning techniques have been applied to optical flow estimation. The method of Sun, Roth, Lewis and Black, 2008 was among the first to end-to-end trained optical flow estimation methods. Due to the lack of training data, it did not fully show the full promise of learning-based optical flow approaches. With the notable exception of Wulff and Black, 2015, machine learning optical flow estimation methods have not been able to achieve the same level of performance as variational methods.

The availability of more processing power, a new deep learning architecture called Convolutional Neural Networks (CNNs), and synthetically generated datasets have inspired Dosovitskiy et al., 2015 to propose two new end-to-end trained encoder-decoder architectures for optical flow estimation. End-to-end means a single feed-forward architecture is used which takes images as input and produces a flow map as output. Note that the availability of large amounts of training data has caused CNNs to become the state-of-the-art method on a variety of computer vision tasks such as stereo, segmentation and object detection. Obtaining sufficient ground-truth data for optical flow has proven to be difficult. Because there are currently no sensors for directly capturing optical flow ground-truth and manual labeling is difficult and time-consuming. Synthetically generated datasets have overcome this problem and allowed deep CNNs to be end-to-end trained. Inspired by the CNN architecture of other pixel-level prediction tasks, such as biomedical image segmentation (Ronneberger, Fischer & Brox, 2015), Dosovitskiy et al., 2015 propose two hourglass-like CNN architectures. One is a correlation-based architecture which first extracts low-level features and then computes a patchwise similarity measure. The other is a spatiotemporal filter-based architecture which takes two stacked images as input.

Dosovitskiy et al., 2015 have provided a brief explanation of the workings of correlation-based architectures and Ilg, Ozgun et al., 2018 have shown that by slightly modifying the correlation-based architecture, the CNN produces high uncertainty in cases where optical flow estimation is difficult. Ilg et al., 2017 showed that the correlation-based architecture is superior in terms of performance. Therefore, recent advances have primarily focused on extending and improving the accuracy of correlation-based architectures and setting a benchmark on public test sets such as MPI-Sintel Butler, Wulff, Stanley and

Black, 2012. Concerning spatiotemporal filter-based architectures, other than a visualization of filters in the first layer by Ranjan and Black, 2017, there has not been any research on *how* this architecture performs optical flow estimation.

1.1 Motivation and research question

Knowing what cues these CNNs exploit for optical flow estimation is useful. Firstly, it advances our current understanding when the network will behave reliably, and what to expect when it deals with scenarios not seen before in a training set. Secondly, the networks can be trained more efficiently by rendering different synthetic datasets or modifying the data augmentation. Thirdly, knowing what the limitations of the architecture are, provides insight for practical applications. This leads to the corresponding main research question:

How do spatiotemporal filter-based convolutional neural networks estimate optical flow?

This main question can be split up into multiple sub-questions which collectively answer the main research question. The structure of the sub-questions is as follows:

1. **Until what scale is the network able to solve the aperture problem?**
2. **How does the network estimate motion?**
3. **What motion patterns are the filters in the network sensitive to?**
4. **How is the network able to overcome the fundamental limit of the uncertainty relation?**

1.2 Structure of this work

This Master of Science thesis constitutes of four parts.The main contributions of this thesis are presented in the scientific paper in Part I. The scientific paper can be read as a stand-alone document. The paper features an introduction to the main concepts relevant to our research and continues into a related work section which discusses relevant literature. After this, the model used in our experiments is specified and the experiment and their results are presented. Lastly, we discuss the findings of our work and draw relevant conclusions. The remaining parts of this thesis provide background information and supporting materials for the scientific paper.

Part II features a review of relevant literature on optical flow modeling and optical flow estimation methods. Chapter 2 deals with the modeling of optical flow, the photometric factors which influence time-varying image intensity and the capturing of optical flow ground-truth. Chapter 3 discusses intensity-based differential methods which consist of global and local methods. Chapter 4 continues with correlation-based methods which define displacement as the shift that gives the best fit between image regions at different times. Chapter 5 discusses image velocity in the frequency domain and frequency-based estimation methods. Chapter 6 contains learning-based methods. Both machine-learning and CNN-based methods are discussed in this Chapter. In Chapter 7 the research gap, based on the findings in literature, is identified.

In Part III a preliminary evaluation of spatiotemporal filter-based CNNs is presented. In this part a performance evaluation of the CNNs is performed and the filters of the first layer of these networks is

visualized. Chapter 8 provides an outline of the methodology used in the experiments and discusses the specifications of the analyzed models. Next, Chapter 9 present the preliminary results and Chapter 10 contains a discussion of these results.

Lastly, in Part IV individual appendices can be found with supplementary material for the preliminary evaluation in Part III. Details about the analyzed models can be found in Appendix A and the flow field color coding used to visualize flow maps can be found in Appendix B.

I

Scientific Paper

How do deep neural networks perform optical flow estimation? A neuropsychology-inspired study

D. B. de Jong*‡, F. Paredes-Vallés†‡, G. C. H. E. de Croon†‡
‡*Micro Air Vehicle Laboratory*
Delft University of Technology
Delft, The Netherlands

***Abstract*—End-to-end trained Convolutional Neural Networks have led to a breakthrough in optical flow estimation. The most recent advances focus on improving the optical flow estimation by improving the architecture and setting a new benchmark on the publicly available MPI-Sintel dataset. Instead, in this article, we investigate how deep neural networks estimate optical flow. A better understanding of how these networks function is important for (i) assessing their generalization capabilities to unseen inputs, and (ii) suggesting changes to improve their performance. For our investigation, we use a filter identification method that has played a major role in uncovering the motion filters present in animal brains in neuropsychological research. The method shows that the filters in the deepest layer of the encoder-decoder neural network are sensitive to a variety of motion patterns. Not only do we find translation filters, as demonstrated in animal brains, but thanks to the easier measurements in artificial neural networks, we even unveil dilation, rotation, and occlusion filters. Furthermore, we find similarities in the refinement part of the network and the perceptual filling-in process which occurs in the mammal primary visual cortex.**

***Index Terms*—Optical flow, convolutional neural networks, Gabor filters, neuropsychology**

I. Introduction

Optical flow is a visual cue defined as the appearance of spatiotemporally varying brightness patterns [1], which can be perceived by both biological vision systems and cameras. This cue is important for the behavior of animals of varying size [2], ranging from small flying insects [3] to humans [4], as it allows these animals to estimate their ego-motion. Optical flow is also important in computer vision and robotics applications, e.g. Micro Air Vehicles (MAVs), for tasks such as object tracking [5], navigation [6], [7], and image interpolation [8].

Many algorithms have been introduced to determine optical flow, including, correlation-based matching methods [9], [10], frequency-based methods [11], [12], and differential methods [13], [14]. Correlation-based matching methods try to maximize the similarity between different intensity regions across multiple frames. Finding the best match then corresponds to finding the shift which maximizes the similarity score. Frequency-based methods exploit either the amplitude or phase component of the complex valued response of a Gabor quadrature filter pair [15] convolved with an image sequence. Differential methods compute optical flow based on a Taylor expansion of the brightness constancy assumption.

*MSc student, †supervisor

Correlation-based, frequency-based, and differential-based methods all compute flow based on the assumption that the brightness of a moving pixel remains constant over time and when applied locally are subject to the aperture problem [16]; the true motion of a one-dimensional structure (such as a bar or an edge) cannot be estimated unambiguously. Instead, only the motion component that is normal to this structure can be perceived. In functional form, this corresponds to one equation with two unkowns (the horizontal and vertical flow component) and thus additional constraints are needed to solve for this equation.

For example, for differential methods a global smoothness constraint has been added [13], which assumes that neighboring pixels undergo a similar motion. Then, the global differential method can be formulated as a global energy function consisting of a data term based on the brightness constancy assumption and a global smoothness term. The global energy term can be minimized using the Euler-Lagrange equations [17], which belong to the mathematical field *calculus of variations*. Methods based on the minimization of a global energy term using this numerical scheme are called *variational* methods and have played a dominant role for many years due to their performance. However, variational methods have two significant drawbacks. First, the iterative minimization of the energy function leads to long computation times. Second, the brightness constancy assumption is a coarse approximation to reality and this limits the performance. Deviations like illumination changes and occlusion violate this assumption [18], [19]. Research has focused on incorporating extra energy terms to deal with deviations from the brightness constancy assumption and improving the robustness of global smoothness constraints, leading to slow but steady progress.

As in many other computer vision areas, currently, the best-performing algorithms are trained deep neural networks. A major challenge that had to be overcome to be able to train such networks was obtaining ground-truth training data. Obtaining this data for the task of optical flow is difficult due to the lack of ground-truth sensing and the excessive human effort required for manual optical flow labeling. *Dosovitskiy et al.* [20] were the first to successfully train deep neural networks to estimate optical flow by using a synthetically generated dataset with optical flow ground truth. Their networks, FlowNetS and FlowNetC, initially performed slightly worse than the state-of-the-art variational methods [21]. However, trained deep neural

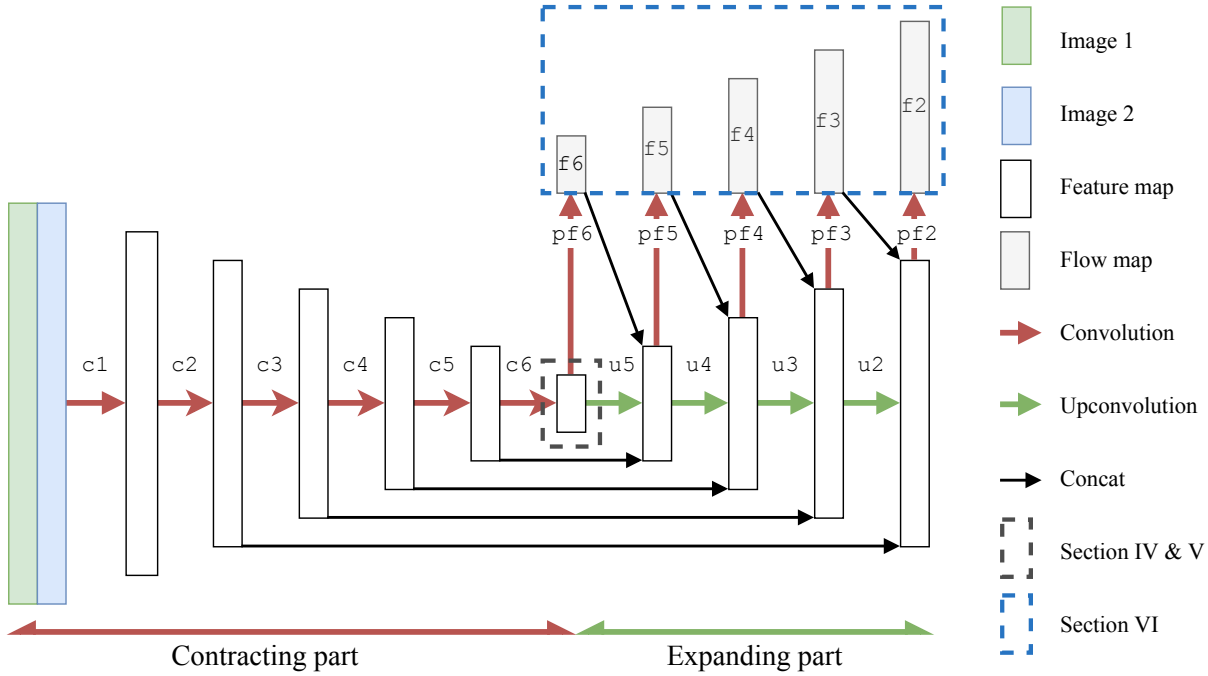


Figure 1: Schematic representation of the FlowNetS architecture. The network consists of a contracting and expanding part. The contracting part compresses spatial information through the use of strided convolutions, and the expanding part uses upconvolutions to refine the flow maps. The predict-flow (`pf`) layers transform the activations in the feature maps to a horizontal and vertical flow component. The final flow map `f2` is bilinearly upsampled to achieve the same resolution as the input. The feature map corresponding to the output of the `c6` layer will be studied in Section IV & V. The flow refinement process will be discussed in Section VI.

networks became the new state-of-the-art method for optical flow estimation by subsequent researchers who focused on improving the architecture and training data [22]–[24].

Until now, the functioning of these networks is poorly understood. In this article we investigate *how* deep neural networks perform optical flow estimation. There are two main reasons why this is important. First, it is difficult to guarantee correct behavior outside of the publicly available testsets without knowing what the network does. Second, a better understanding of what the architecture does may lead to valuable recommendations for improving the performance, for instance, by changing properties of the architecture or training data. To the best of our knowledge this is the first work which provides extensive insight into the workings of deep neural networks for optical flow estimation.

In our analysis of deep optical flow networks, we make use of a method that has helped unveiling the workings of motion-sensitive brain areas in neuropsychology [25]. Specifically, we measure the response of neurons in the deepest layer of the contracting part of FlowNetS [20] to stimuli with varying spatio-temporal frequencies, and construct a spectral response profile. The input stimuli used are translating plane waves, as this input type proved to be more selective in the frequency domain than moving bars [26]. Based on the earlier findings of Gabor filters [15] in biological vision systems [27], [28] and other learning-based methods, such as independent component analysis [29] and learned basis functions for sparse image representation [30], we expect to find these filters in FlowNetS as well. Therefore, we fit a Gabor function to the spectral response profile and study the residual error patterns. We find that the Gabor translational motion filter model is suitable for the majority of the filters. Additionally, we find filters sensitive to motion patterns such as dilation, rotation, and occlusion. Interestingly, neurons sensitive to these motion patterns have not been mentioned in neuropsychology literature. Furthermore, our analysis strongly suggests that the resolution in the temporal frequency domain can be significantly improved if more than two frames would be used as input to the neural network. Lastly, we find that the optical flow refinement process in the decoder part of the network behaves similarly in function to flow refinement in biological vision systems.

The structure of this paper is as follows. Section II provides an account of the related work regarding neural response identification methods used in neurosychology and deep-learning. In Section III an explanation of the architecture of our version of FlowNetS [20], which can be seen in Figure 1, is given. Also, the motivation behind the minor changes in training and architecture is discussed here. In Section IV, translating plane

waves are used as input to FlowNetS and the fit of the Gabor filter model [15] to the response of filters in the `c6` layer is analysed. Next, the limitations of the methodology of Section IV is discussed in Section V. Furthermore, Section V discusses the response of filters in the `c6` layer of FlowNetS to dilating and rotating waves. Section VI outlines until what scale the neural network is able to resolve the aperture problem and highlights the filling-in effect which occurs in the expanding part of the network. A discussion based on the findings takes place in Section VII and outlines potential future work. Lastly, the conclusion drawn from the experiments can be found in Section VIII.

II. Related Work

A. End-to-end trained neural networks for optical flow estimation

Every since the pioneering work of *Horn et al.* [13], variational optical flow methods [31] have played a dominant role in optical flow estimation due to their performance. Most modern variational optical flow estimation pipelines consist of four stages: matching, filtering, interpolation, and variational refinement. Various improvements have been proposed over time to deal with issues such as long-range matching [32] and occlusion [33]. Furthermore, improvements such as dense correspondence matching based on convolution response maps of the reference image with the target image [34], and supervised data-driven interpolation of a sparse optical flow map [35] were also proposed. These last two improvements introduced elements of deep learning into the variational optical flow estimation pipeline.

Dosovitskiy et al. [20], however, were the first to introduce a supervised end-to-end trained Convolutional Neural Network (CNN). CNNs have three major advantages when it comes to estimating optical flow. First, as shown by subsequent researchers [22]–[24], CNNs outperform variational optical flow estimation methods in terms of accuracy, thus establishing a new state-of-the-art in this problem. Second, the runtime of CNN-based optical flow algorithms is significantly lower than variational methods [22]. Third, CNN-based methods can learn from data and can exploit statistical patterns not realized by a human designer. This is an advantage over variational methods which require explicit assumptions on the input which are coarse approximations to reality. However, CNN-based methods also have three disadvantages. First, the results obtained depend on the quality and size of the training data used. Second, CNN-based methods face the risk of overfitting, which is especially the case for optical flow estimation because it is difficult to obtain ground truth [21]. Third, there is no guarantee that the trained models will generalize to scenarios which are not encountered in the training dataset. Due to the black-box nature of these methods there is also no insight into the limitations of the networks and the workings of the learned solution. There are also two difficulties which arise when using CNN-based methods. First, due to the large amount of parameters, the memory footprint of these models is typically large. Second, the learning process is significantly affected by the setting of hyperparameters [23] and the loss function used [36].

Dosovitskiy et al. [20] introduced two architectures, i.e. FlowNetS and FlowNetC, based on the U-net architecture [37], which consists of a contracting and an expanding part. In the contracting part information is spatially compressed and in the expanding part information is refined. While FlowNetS is a rather generic network consisting of simple convolutions, FlowNetC creates two separate processing streams and combines these streams in a *correlation-layer*. This layer performs a multiplicative patch comparison between feature maps. Due to the explicit use of a correlation-layer, it is more straightforward to understand the workings of FlowNetC. Subsequent researchers have focused on improving the correlation-based architecture by using an image pyramid with warping in between pyramid levels [23], and a flow regularization method based on variational energy minimization principles [24]. However, not much is known about the workings of FlowNetS. *Ranjan et al.* [38] introduced SpyNet, a spatial image pyramid with simple convolutional layers at each pyramid level and a warping operation between pyramid levels. They visualized the weights of the first layer of their network and claim that these filters resemble Gabor filters [15]. This provided a glimpse into the working principle of SpyNet. Finally, *Teney et al.* [39] built a shallow CNN-architecture by integrating domain knowledge, such as invariance to brightness and in-plane rotations, and using signal processing principles. On small motion, their architecture performs well, but their shallow CNN performs poorly on large motion near occlusions. They conclude good occlusion performance requires reasoning over a larger spatiotemporal extent, which their shallow architecture is not able to do.

Ilg et al. [40] tried to quantify the uncertainty of CNN-based methods to handle the black-box nature of deep neural networks. They used a modified FlowNetC which produces multiple hypotheses per forward pass, which are then merged to a single distributional flow output. They showed that their network produces flow estimations with high uncertainty in cases where optical flow estimation is difficult (shadows, transparant motion, etc.). Lastly, *Ranjan et al.* [41] highlighted another downside of deep neural networks, which is the ability of adversarial examples to fool neural networks and produce erroneous results. They showed that especially networks using an encoder-decoder architecture are affected, while networks using a spatial pyramid framework are less vulnerable. None of the works above, however, provide an explanation of how their architecture performs optical flow estimation.

B. Receptive field mapping

In order to understand what neural networks have learned, two threads of research in neural network interprability can be discerned: attribution and feature visualization. Attribution methods [42], [43] are used to *attribute* filter outputs, like optical flow, to parts of the input by visualizing the gradient. However, it is hard to see where an optical flow estimate comes from. On the other hand, feature visualization is concerned

with understanding what neurons, filters, or layers in a neural network are sensitive to by optimizing the input [44]. When optimizing the input, the result is usually an image with noisy and visually difficult to interpret high-frequency patterns [45]. Three methods of regularization can be applied to cope with this phenomenon. First, frequency penalization discourages the forming of these patterns. The downside is that this approach also discourages the forming of legitimate high-frequency patterns which are of interest for optical flow estimation. Second, small transformations like scaling, rotation, or translation can be applied in between optimization steps [46]. This approach is also not viable because transformation affects the ground truth of optical flow. Third, priors can be used which can keep the optimized input interpretable. Such approaches typically involve learning a generative model [47] or enforcing priors based on statistics from the training data [48]. Also, this approach is often very complex and it may be unclear what can be attributed to the prior and what can be attributed to what the network has learned.

Due to these reasons, we look at the field of neuropsychology and specifically study what methods researchers have used to determine what stimuli activate neurons in mammalian vision systems and what functions best describe the responses. It was shown that Gabor functions [15] best modeled the spatial response of simple cells in the mammal visual cortex [27], [49], [50]. It can be shown that Gabor filters are optimal for simultaneously localizing a signal in the spatial and frequency domain [51], making them ideal for motion estimation. Later, *DeAngelis et al.* [52] examined the spatiotemporal response of cells and their space-time separability. If a cell is space-time separable, it can be described as the multiplication of a function of space and a function of time. If the response of a cell is not space-time separable, a spatial description of the receptive field profile does not suffice. In functional form, space-time separable Gabor filters are frequency-tuned with a stationary Gaussian envelope and space-time inseperable Gabor filters are velocity-tuned with a moving Gaussian envelope [53]. In this work we only considered fitting frequency-tuned Gabor filters, due to their simplicity and the low number of frames used by FlowNetS and FlowNetC.

Two approaches to receptive field mapping in neuropsychology can be discerned: the reverse-correlation based approach and the spectral response profile approach. The reverse-correlation-based approach presents a rapid random sequence of flashing bars at various imaging locations to the mammal. The spike train emitted by the neuron in the subject is correlated to the sequence in which the stimuli were presented. This approach allows for a rapid measurement of the receptive field profile in the spatiotemporal domain [28]. On the other hand, the spectral response profile approach presents translating plane waves to the mammal at varying orientations and spatiotemporal frequencies [54], [55]. *Jones et al.* used both the reverse-correlation approach to construct a spatial receptive field profile [56] and measured the response to plane waves to construct a spectral response profile [25]. Subsequently, the spatial and spectral responses obtained were compared to the Gabor filter model in the spatial and frequency domain and the filter parameters obtained from both methods proved to be highly correlated [27]. *Deangelis et al.* [52] used the reverse-correlation approach to measure the spatiotemporal receptive field profiles in visual cortex of cats. In a follow-up work, they examine the linearity in the spatial and temporal responses [55]. Therefore, they compared the Fourier-transformed responses obtained using the reverse-correlation procedure to the spectral responses obtained using translating plane waves.

In this work we extend the approach of *Jones et al.* [27] to the spatiotemporal domain and measure spectral responses of the network to translating plane waves to which frequency-tuned spatiotemporal Gabor filters will be fitted. A benefit of measuring the spatiotemporal spectral responses for optical flow is that translation is more easily described in the frequency domain [53]. Although there has been research into non-Fourier motion, such as theta motion [57], translucency [58], and occlusion [59], an analytical description of dilation and rotation in the frequency domain is, to the best of our knowledge, missing. Therefore we simulate the response of dilation and rotation filters to translating plane waves, which can be found in Section V.

C. Aperture problem

Optical flow estimations methods are only able to resolve motion components normal to the orientation of an edge in the intensity pattern. If motion takes place tangent to an edge, then we are not able to resolve it locally. This is known as the aperture problem [16]. In CNNs the size of the aperture of a neuron is referred to as the receptive field. The receptive field is defined as the region in the input which affects the activation of the neuron. For a neuron in a given layer, it can be calculated what the receptive field size is using simple arithmetic [60]. In this work we show that the receptive field size is related to the aperture problem by training different versions of FlowNetS with varying receptive field sizes.

In neuropsychology, *Komatsu* [61] has shown the existence of a perceptual filling-in mechanism in the mammalian visual cortex for cues such as colour, brightness, texture, or motion. While the precise neural workings are still under discussion, edge structure [62] and the interaction between neighboring neurons play an important role in this process [63].

In neural networks attempts have been made to implement such a mechanism as well. To allow for the interaction between neurons, a recurrent model can be used [64]. *Zweig et al.* [35], however, used an unfolded feed-forward version of a recurrent network and a multi-layer loss to allow for interaction between neurons. Their CNN-based motion interpolation architecture takes a sparse flow map and edge structure as input. They showed their motion interpolation method refines motion estimates similarly to the human visual cortex by demonstrating the filling-in effect of the network on a Kanizsa illusion [65]. FlowNetS also features a multi-layer loss, and in Section VI the ability of the expanding part of FlowNetS to interpolate and refine flow maps is highlighted.

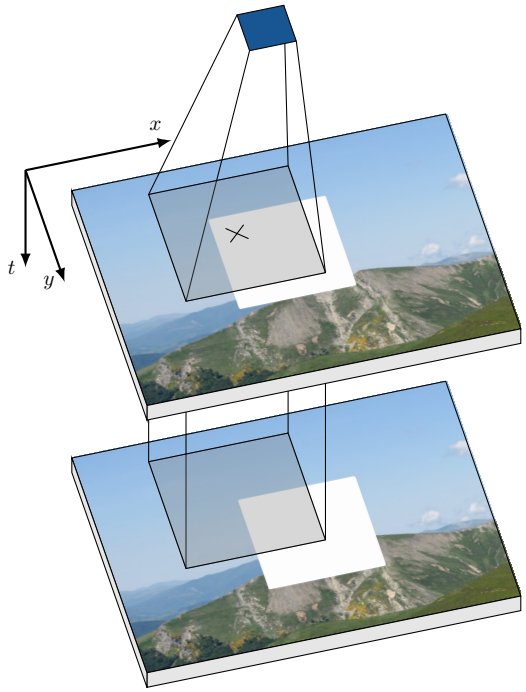


Figure 2: The receptive field size corresponding to the activation of a filter in the `c6` layer of our FlowNetS projected onto two 1024x768 input frames. The cross marks the center pixel and corresponds to the origin of the coordinate system. Image is drawn to scale in the xy-plane.

III. Model details

In this section we specify the model which was used during the experiments. In Figure 1 a schematic representation of the architecture of FlowNetS can be seen, which takes two consecutive images as input. The network consists of a contrating part which uses strided convolutions to compress spatial information, and an expanding part that uses upconvolutions and a multi-layer loss function. The flow map `f2` is bilinearly upsampled to achieve an output flowmap of the same resolution as the input.

We slightly modify the original FlowNetS in order to improve the interpretability of the motion filter analysis. First, we use a ReLU activation function as opposed to a leakyReLU[1] activation function to simplify the spectral Gabor fitting process discussed in Section IV. Furthermore, in the predict-flow (`pf`) layers the bias terms are removed because the flow is assumed to be zero-centered. Also, the kernel size in the `pf` layers is reduced from 3x3 to 1x1, meaning that the activations in the `c6` layer are converted to motion in u and v image coordinates[2] by means of a simple multiplication, resulting in the coarsest flow map `f2`. This brings the total receptive field size in the `c6` layer to 383 pixels as opposed to the original size of 511 pixels. The size of the receptive field is depicted in Figure 2. The full details of our version of FlowNetS can be found in Table I in Appendix A.

Regarding training, as in [20] we use the same data augmentation *on both* frames, but we do not use incremental flow and color augmentation *between* frames since the authors do not specify the parameters of the latter data augmentation scheme. Furthermore, the network is trained for fewer iterations (300K iterations versus 600K iterations) due to limited availability of computational resources. Evaluation on the MPI-Sintel dataset and FlyingChairs dataset shows comparable performance between our FlowNetS and the original version, as can be seen in Table II in Appendix A.

The synthetic dataset FlyingChairs, which was used to train the original and our slightly modified FlowNetS, consists of approximately 22k image pairs. The image pairs are composed of a varying numbers of chairs and background images from natural scenes. Between image pairs, a composition of translation, rotation, and scaling motion is applied. The size of the chairs is sampled from a Gaussian with a mean and standard deviation of 200 pixels, clamped between 50 and 640 pixels. Note that the synthetic scenes also contain occlusion. Details about the composition of affine motion can be found in [20].

IV. Gabor spectral response profile fitting for translation

In this section we investigate what motion patterns the filters in the `c6` layer of our FlowNetS are sensitive to. Instead of analyzing the selectivity of all FlowNetS filters, we focus our study on the filters of the `c6` layer. As shown in Figure 1, the activations of the feature maps of these layers are directly transformed by two multiplicative values (i.e. `pf6`) into a horizontal and vertical motion estimate (i.e. `f6`). Subsequently, this initial flow estimate is also used for refinement. For these reasons, we believe that the most compressed representation of what optical flow is and how to estimate it, is encoded in this layer.

In this section, first the theory behind Gabor filters and the Gabor spectral response fitting method is discussed. Second, the result obtained are presented. Third, we discuss the resolution in the temporal frequency domain of the fitted filters.

A. Methodology

As in [11], [15], [53], the spatiotemporal frequency-tuned Gabor filter g in Cartesian coordinates centered in the origin $(0, 0, 0)$ can be written as the product of a spherical Gaussian w and a translating plane wave s:

$$g(x, y, t) = s(x, y, t)w(x, y, t) \tag{1}$$

where the spherical Gaussian w is defined by:

$$w(x, y, t) = \exp\left(-\pi\left(\frac{x_r^2}{\sigma_x^2} + \frac{y_r^2}{\sigma_y^2} + \frac{t^2}{\sigma_t^2}\right)\right) \tag{2}$$

where σ_x, σ_y, and σ_t control the spread of the spatiotemporal Gaussian window. The spherical Gaussian w can be centered at any spatial location using an offset. To decrease the number

[1] *Dosovitskiy et al.* [20] mention the use of the ReLU activation function in their work. The release of their pre-trained models, however, uses a leakyReLU activation function.

[2] u and v correspond to motion in x and y direction respectively.

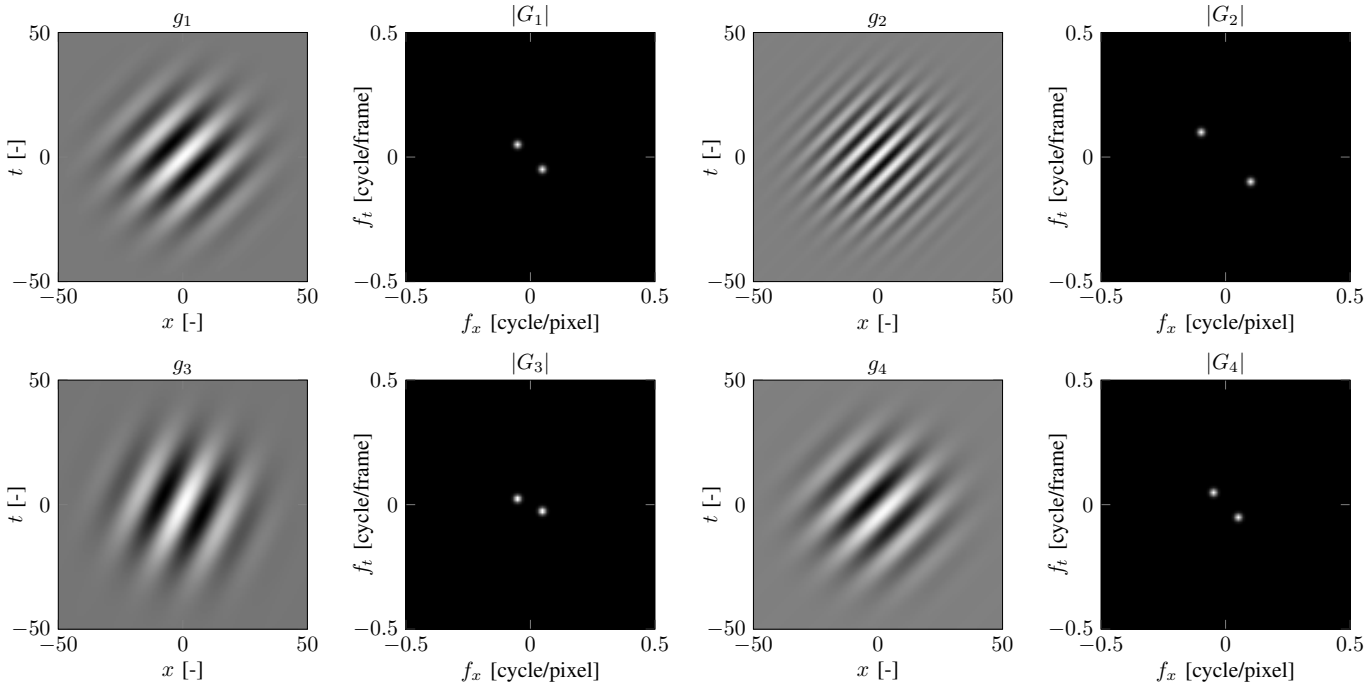


Figure 3: Gabor filters g_1, g_2, g_3, and g_4 in the xt-domain and the power spectrum of the corresponding Fourier-transformations G_1, G_2, G_3, and G_4. g_1 and g_2 are sensitive to the same velocity v_0. g_3 is tuned to the same spatial frequency f_x as g_1 but to a lower temporal frequency f_t and is thus sensitive to a lower velocity v_0. g_4 is tuned to the same frequencies as g_1 but at phase φ_0 of 90 degrees.

of parameters in the fitting process, it is assumed that the center of the Gaussian coincides with the center pixel of the receptive field. Furthermore, the subscript r denotes a rotation operation which allows the spherical Gaussian to be aligned along orientation θ_0 and is defined as:

$$\begin{aligned} x_r &= x\cos(\theta_0) + y\sin(\theta_0) \\ y_r &= -x\sin(\theta_0) + y\cos(\theta_0) \end{aligned} \tag{3}$$

where a positive value of θ corresponds to a clockwise rotation with respect to the positive x-axis. Note the use of a clockwise convention due to the use of the pixel-coordinate system which uses a downward positive y-axis as can be seen in Figure 2. The subscript 0 indicates the parameter value corresponding to the peak response of the filter. The center of the coordinate system corresponds to the center of the receptive field as indicated by the cross in Figure 2.

Furthermore, a translating plane wave s in the Cartesian coordinate system can be written as:

$$s(x, y, t) = \cos\left(2\pi\left(F_0 x_r - f_{t_0} t\right) + \varphi_0\right) \tag{4}$$

where the spatial frequency magnitude F_0 in cycles per pixel is related to the spatial frequency in x and y direction via $F_0 = \sqrt{f_{x_0}^2 + f_{y_0}^2}$, and the preferred direction of motion θ_0 to the spatial frequencies via $\theta_0 = \tan^{-1}\left(f_{y_0}/f_{x_0}\right)$. A higher spatial frequency F_0 allows tracking of motion of thinner image structures. Note that velocity v_0 is defined as pixels per frame and is related to spatial frequency F_0 and temporal frequency f_{t_0} via $v_0 = f_{t_0}/F_0$ [11]. When a signal is sampled in time or space, frequency components which are larger than or equal to 0.5 cycle per frame (the Nyquist frequency) become undersampled and aliasing occurs. Thus, if we limit ourselves to signals which do not suffer from aliasing, the maximum velocity a signal can have is limited by its spatial frequency F_0.

Now consider two Gabor filters in the xt-domain, g_1 and g_2, as depicted in Figure 3. The spatial frequency F_0 and temporal frequency f_{t_0} of g_2 are twice as large as for g_1 and thus both filters are sensitive to the same velocity v_0. In fact, it can be seen that all spatiotemporal frequency components with velocity v_0 lie on a straight line passing through the origin, and the slope corresponds to the velocity magnitude $|v_0|$. In Figure 4a the 3D frequency space with the half-magnitude profile of a Gabor filter g is visualized and the slope of the velocity magnitude $|v_0|$ can be seen. Furthermore, g_3 in Figure 3 depicts a filter tuned to the same spatial frequency f_x as g_1 and a lower temporal frequency f_t , resulting in a filter tuned to a different velocity v_0.

Lastly, φ_0 denotes the phase, and the filter is even when $\varphi_0 = 0$ and odd when $\varphi_0 = \pm\pi$. An example of this can be seen in g_4 and g_1 which depict a sine and cosine respectively.

Because we will fit the response of phase-sensitive filters, we highlight three phase-dependent convolution phenoma. Note that a valid convolution of two tensors with equal size corresponds to the sum of the dot product of two tensors. First,

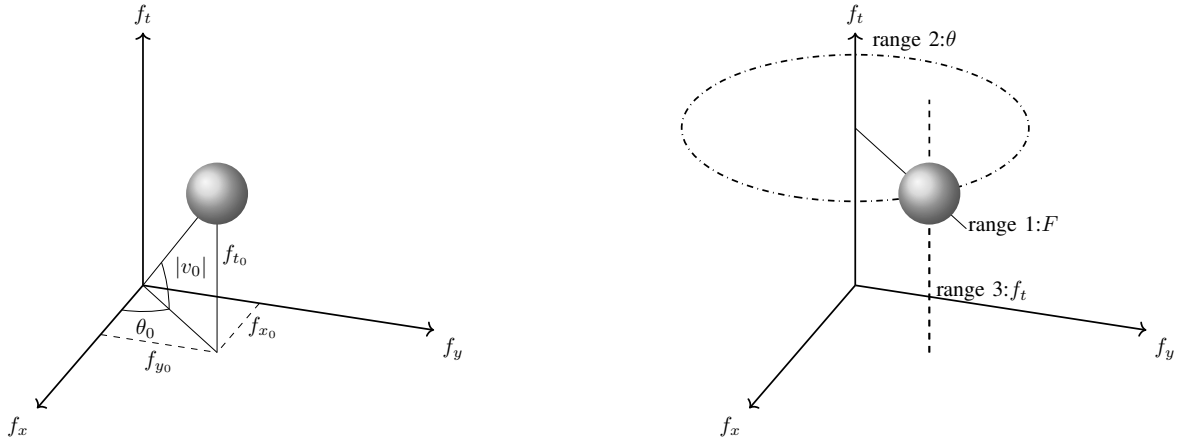


Figure 4: (a) Illustration of the half-magnitude profile in the 3D frequency domain of a spatiotemporal Gabor filter. (b) The three ranges along which the responses of the Gabor half-magnitude profile will be evaluated for the fitting process.

because a sine is an odd signal, the dot product of two sines at opposite frequencies $-f$ and $+f$ is negative as can be seen in the top plot in Figure 5. Second, the dot product of a cosine (an even signal) at opposite frequencies will be positive due to the even nature of the function as can be seen in the middle plot in Figure 5. Third, sine and cosine are *decorrelated* and thus the dot product will be zero between these two signals which is illustrated in the bottom plot in Figure 5.

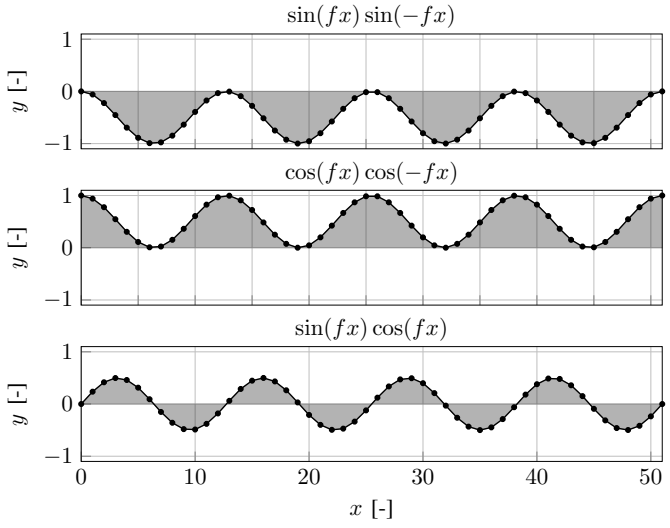


Figure 5: Dot product of waves with the same frequencies of different phase. *Top:* Dot product of two sines with opposite frequencies resulting in a negative activation. This demonstrates the odd nature of a sine. *Middle:* Dot product of two cosines with opposite frequencies resulting in a positive activation. This demonstrates the even nature of a cosine. *Bottom:* Dot product of a sine and cosine at the same frequency. Because sine and cosine are decorrelated this results in zero activation.

Gabor spectral response profile fitting

In the Gabor spectral response fitting process, translating plane waves s are used as input and we try to minimize the difference in response between filters in the $\texttt{c6}$ layer of our FlowNetS and fitted Gabor filters g. To better approximate the response of the filters in the $\texttt{c6}$ layer, we enhance the Gabor filter output with a gain term K, bias term b, and pass the response through a ReLU non-linearity. Then the response r to a convolution with a translating plane wave s and a Gabor filter g is given by:

$$r = \text{ReLU}\left(K(s(x,y,t) * g(x,y,t) + b)\right) \tag{5}$$

where the response r is a function of nine parameters. These parameters are estimated in a two-step process.

First, a gridsearch is performed to determine the location in the spatiotemporal frequency domain with the highest response per filter in the $\texttt{c6}$ layer. We denote the response of the filters in the network by $\hat{r}$ and the peak response value by $\hat{r}_0$. Because the fitted Gabor filters are phase sensitive, this amounts to estimating four parameters (F_0, θ_0, f_{t_0}, φ_0). Therefore, a four-dimensional grid is constructed of all combinations of these parameters within a given range and step size. The range and step size per parameter used for a translating plane wave s can be seen in Table III in Appendix B. The range for the value of half spatial wavelength $\lambda_0/2 = 1/(2F_0)$ is chosen so that it captures the sizes of the chairs present in the training dataset (as explained in Section III).

Second, the spatiotemporal spread of the Gaussian, determined by $(\sigma_x, \sigma_y, \sigma_t)$, and the non-linear transformation parameters (K, b) are estimated. This is done by varying the F_0, θ_0, and f_t parameters along three separate ranges. A depiction of the dimensions along which the response $\hat{r}$ is evaluated in the spatiotemporal frequency space can be seen in Figure 4b. The range per parameter along which the responses are evaluated can be found in Table IV in Appendix B. Then, we define the cost function $\mathcal{L}$ as the squared difference in

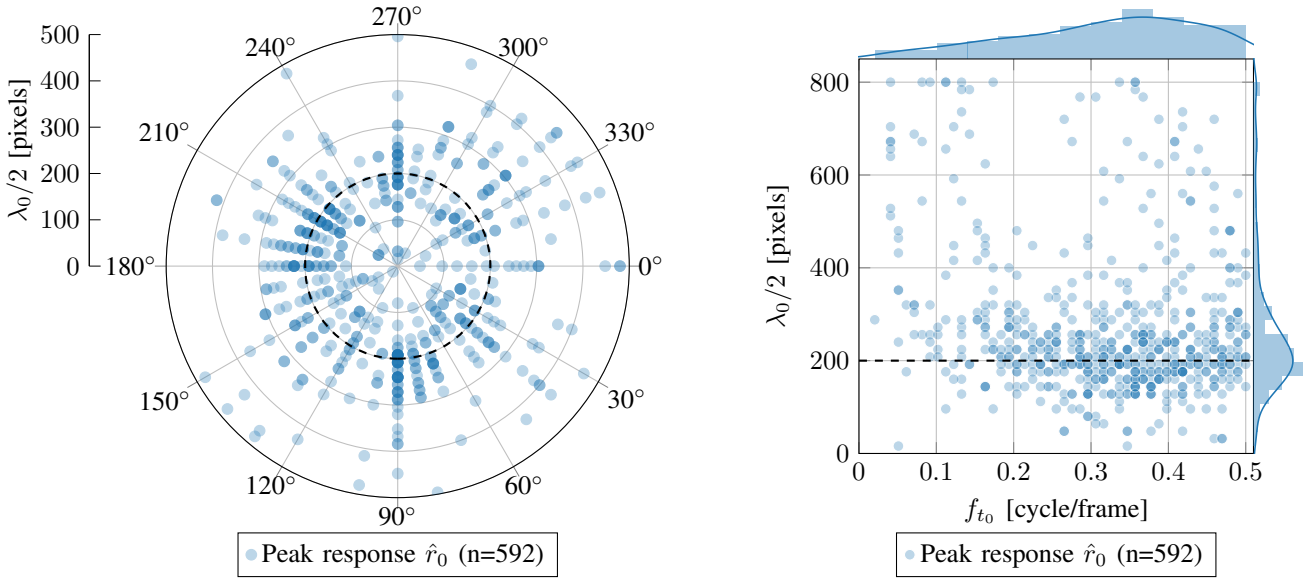


Figure 6: Location of peak response $\hat{r}_0$ per filter ($n = 592$) in the spatiotemporal frequency domain in response to translating plane waves. *Left:* Half spatial wavelength $\lambda_0/2$ and orientation θ_0 corresponding to peak response $\hat{r}_0$ per filter. The radial limit of the axis is set to 500 pixels to improve readability. *Right:* Half spatial wavelength $\lambda_0/2$ and temporal frequency f_{t_0} corresponding to peak response $\hat{r}_0$ per filter. The distribution of half spatial wavelength exhibits a peak around 200 pixels, indicated by the black dashed line, which is to be expected due to the nature of the training data.

response between the fitted Gabor filter per datapoint r, and the filter $\hat{r}$ in the $\texttt{c6}$ layer per datapoint i along three ranges of varying parameters in response to a convolution with a translating plane wave s:

$$\begin{aligned}\mathcal{L} &= \sum_i \|r_i - \hat{r}_i\|_F + \sum_j \|r_j - \hat{r}_j\|_\theta + \sum_k \|r_k - \hat{r}_j\|_{f_t} \\ &= \mathcal{L}_F + \mathcal{L}_\theta + \mathcal{L}_{f_t}\end{aligned} \tag{6}$$

where $\mathcal{L}_F$, $\mathcal{L}_\theta$, and $\mathcal{L}_{f_t}$ denote the Sum of Squared Errors (SSE) along their respective intervals. In order to compare the obtained cost values between filters, we construct a normalized cost value $\mathcal{L}_{norm}$ by dividing the cost by the squared peak response of the filter:

$$\mathcal{L}_{norm} = \mathcal{L}/\hat{r}_0^2 \tag{7}$$

We constrain the bounds of the Gabor filter parameters to obtain reasonable values. This leads to a non-linear bounded convex optimization problem which can be solved using the robust trust-region-reflective algorithm [66], [67]

B. Results

We found 592 of the 1024 filters in the $\texttt{c6}$ layer of FlowNetS to have an activation larger than 0. This indicates that our network has a lot of 'dead neurons', a problem which can be attributed to the ReLU activation function [68]. The location of the peak response of the active filters in terms of half spatial wavelength $\lambda_0/2$, orientation θ_0, and temporal frequency f_{t_0} can be seen in Figure 6. In the left plot of Figure 6 it can be seen that the locations of the peak responses of the filters are well distributed over all angles. Radially, there is a concentration around a half spatial wavelength of 200 pixels (indicated by the red dotted line), which is to be expected based on the nature of the training data as the average size of the chairs in the training dataset is 200 pixels. The concentration of the peak responses becomes even more apparent in the right plot of Figure 6 which shows the distribution along the temporal and half spatial wavelength axes. Furthermore, we note that the distribution of the temporal frequencies is skewed toward the Nyquist limit of 0.5 cycle per frame. A possible reason for this is the low resolution in the temporal frequency due to the low number frames used as input. This will be further discussed in Section IV-C.

The fitted modified Gabor functions (Equation 5) seem to capture the selectivity of the $\texttt{c6}$ filters of FlowNetS accurately. In order to give insight into the goodness of fits for all neural responses in the $\texttt{c6}$ layer, we show three example responses corresponding to different normalized cost values $\mathcal{L}_{norm}$ in Figure 7. Note that the red and green filters fit the data reasonably well, but the red filter shows a systematic deviation from the fitted Gabor filter near $\theta = 0$. For this reason, all the fits and error patterns above the 75% percent threshold, corresponding to the green filter, were visually inspected for systematic deviations. A visual inspection is performed because an auto-correlation procedure is not possible due to a non-uniformly spaced polar 3D frequency grid[27].

Figure 9A shows a filter in the $\texttt{c6}$ layer whose response fits nicely in the Gabor filter framework. Note that the measured data, fit, and error are visualized in the 2D polar spatial frequency domain at peak response temporal frequency f_{t_0}

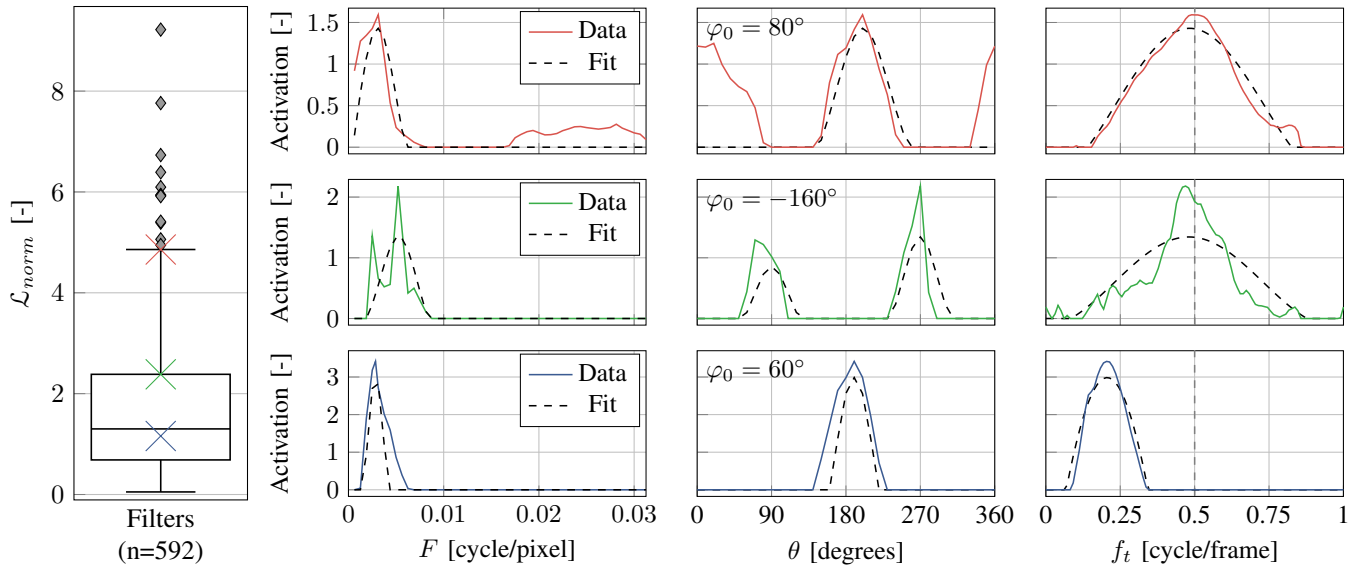


Figure 7: Quantitative results of the spectral Gabor filter fitting process. *Left:* Boxplot containing the total normalized cost $\mathcal{L}_{\text{norm}}$ per filter ($n = 592$). The blue, green and red cross correspond to a filter at the median, near the 75th percentile and near the upper whisker limit, respectively. The grey diamonds denote the outliers. *Right 3x3 plots:* Row-wise the measured responses of three different filters in $\mathtt{c6}$. The top, middle and bottom row correspond to the cost value as indicated in the boxplot. The dotted line per row denotes the response of the fitted Gabor filter.

and phase φ_0. Furthermore, we find three types of systematic deviations (Figure 9B, 9C, 9E) from the Gabor model and conclude that some patterns are too complex for interpretation, such as Figure 9D, partly due to the limitations of the methodology, which will be further discussed in Section V.

First, note the red filter in Figure 7 which shows a systematic deviation from the fitted Gabor filter 180 degrees away from θ_0. The red filter is responsive to edge structure ($|\varphi_0| \approx 90°$) and is thus approximately odd. Remember that the dot product of two odd signals at opposite frequencies results in a negative value (see Figure 5). However, this filter is sensitive to edge structure and still produces a positive activation at the opposite spatial frequency $-F_0$, corresponding to 180 degrees away from θ_0. In Figure 8 the distribution of the phase values φ_0 versus orientation cost $\mathcal{L}_\theta$ for all filters is depicted. It can be seen that there are multiple filters responsive to edge structure which have a high orientation cost $\mathcal{L}_\theta$. One possible reason for this systematic deviation from the Gabor response is that the network is able to learn a successful flow filter that is invariant to polarity (meaning white-black or black-white transitions). This mechanism can be seen as an improvement over a phase-sensitive Gabor filter, and merits further investigation in future work. In Figure 9B the 2D spatial frequency response is visualized of such a filter.

Second, we find two filters which exhibit weak directional bias. It should be noted that these types of filters are also found in the Lateral Geniculate Nucles (LGN) of mammals [28]. An example of such a filter can be found in Figure 9C.

Third, we also find filters which exhibit two or more Gaussian peaks with similar peak response magnitudes tuned to

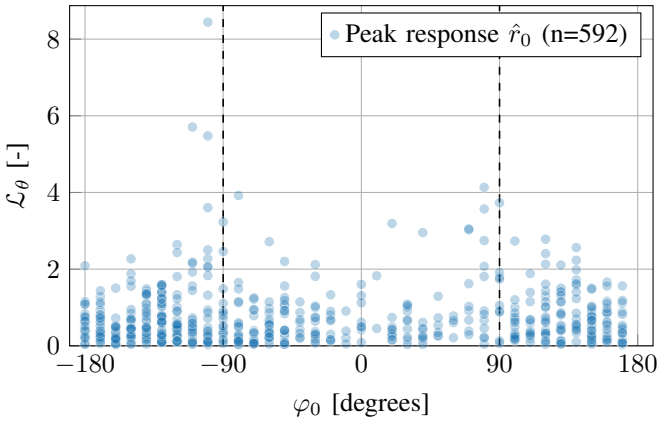


Figure 8: Orientation cost $\mathcal{L}_\theta$ per filter as a function of peak response phase φ_0. It can be seen that a number of filters activating on edge structure near ± 90 degrees phase have a orientation cost.

different spatial frequencies F_0, orientations θ_0, and temporal frequencies f_{t_0}. An example of such a filter can be found in Figure 9E and the 2D spatiotemporal representation corresponding to this filter can be found in Figure 10. A possible explanation is that these filters are sensitive to occlusion which will be further discussed in Section V.

Lastly, we also find filters which appear noisy and are hard to interpret given the limitations of our methodology. Such an example can be seen in Figure 9D and the limitations of our methodology will be discussed in Section V.

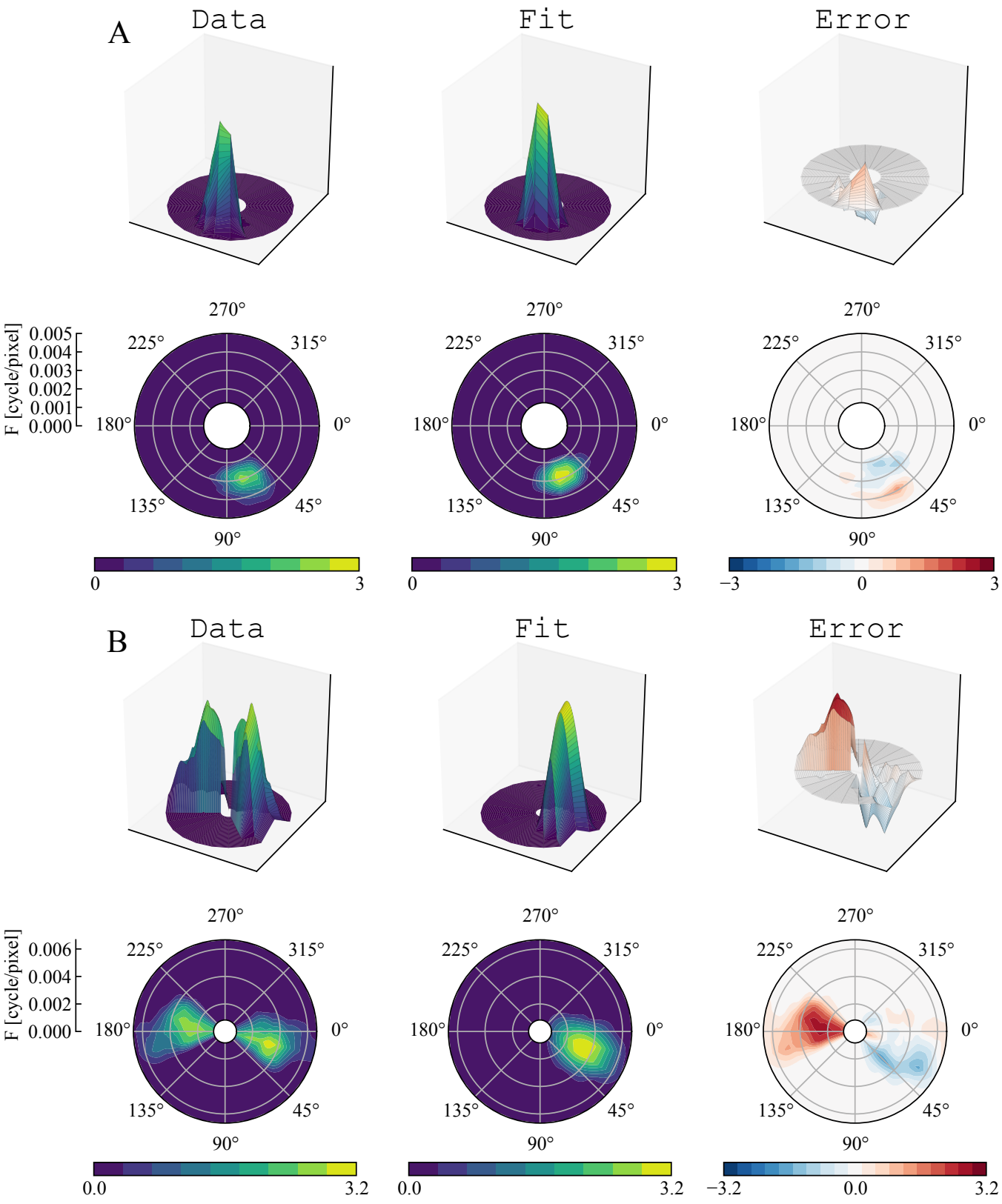


Figure 9: Qualitative analysis of the error patterns of the spectral Gabor fitting process. The spectral response profiles are shown as a function of spatial frequency F and orientation θ. `Data` corresponds to the measured response of a single filter, `Fit` is the response of the fitted Gabor filter, and `Error` shows the difference between these responses. Evaluations are with respect to temporal frequency f_{t_0} and phase φ_0 corresponding to peak filter response. These dimensions are omitted for brevity. (A) The spectral response profile of a filter which does not show any systematic deviation. (B) A filter which activates on opposite spatial Frequency $-F_0$ although the peak response phase φ_0 is around 90 degrees.

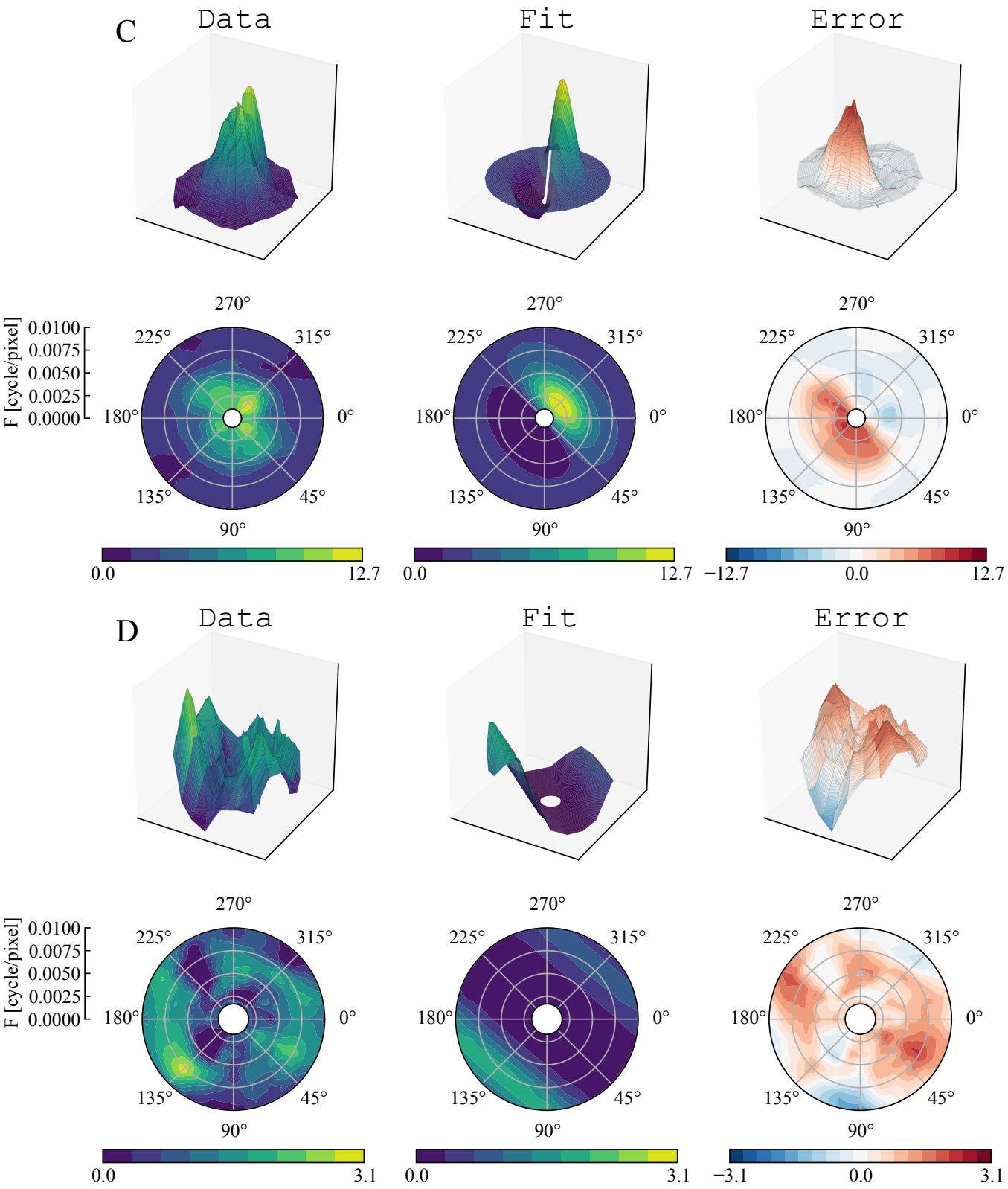


Figure 9: (continued) Qualitative analysis of the error patterns of the spectral Gabor fitting process. (C) Filter with a very weak directional bias. Filters with a weak directional bias are also found in the Lateral Geniculate Nucles (LGN) of mammals[28]. (D) Noisy filter pattern which is difficult to interpret given the current limitations of our approach, which will be discussed in Section V.

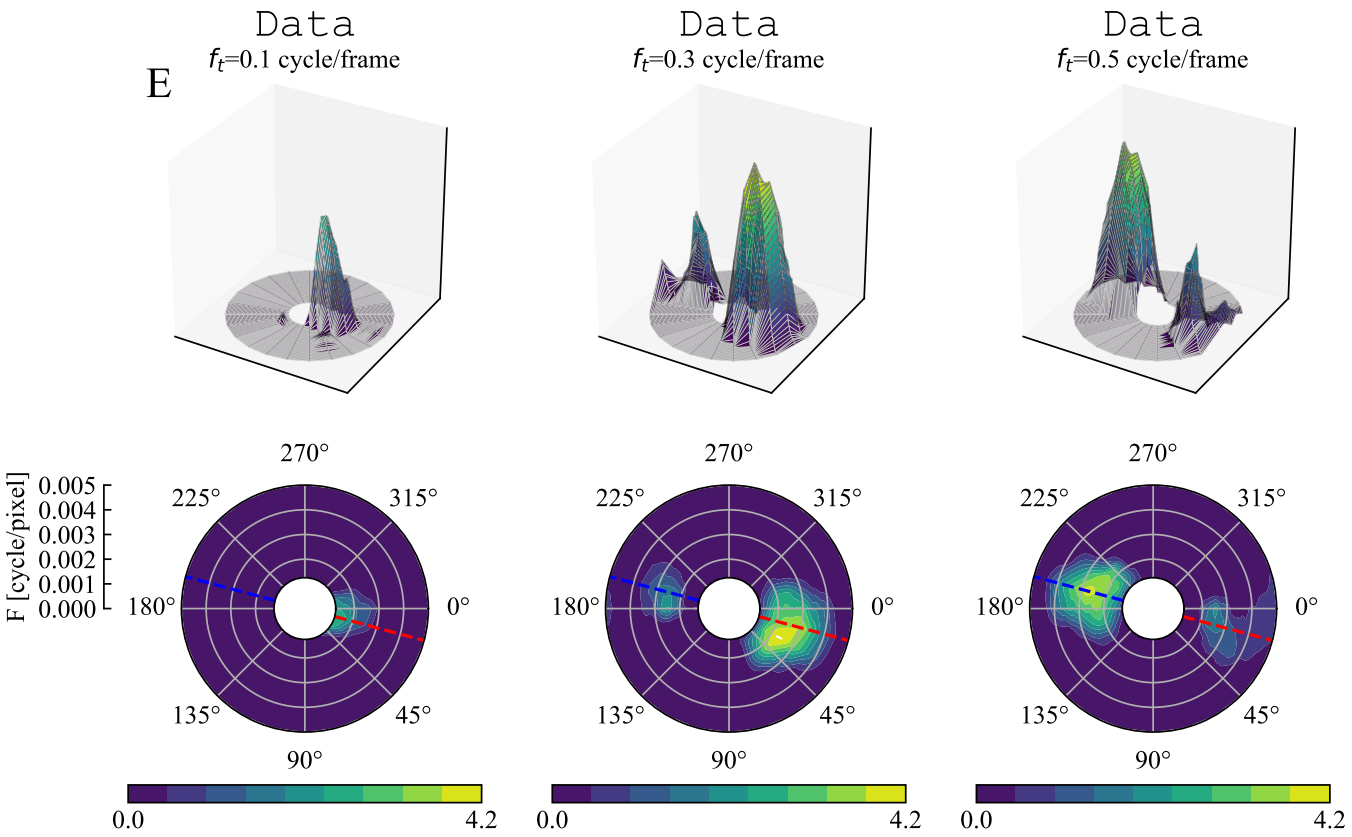


Figure 9: (continued) Qualitative analysis of the error patterns of the spectral Gabor fitting process. (E) For this filter the spectral response profile for three different temporal frequency f_t values is visualized. Two different Gaussian peak responses at opposite orientation at $f_t = 0.3$ and $f_t = 0.5$ cycle per frame can be seen. The blue and red lines correspond to the axes of the 2D representation which can be seen in Figure 10.

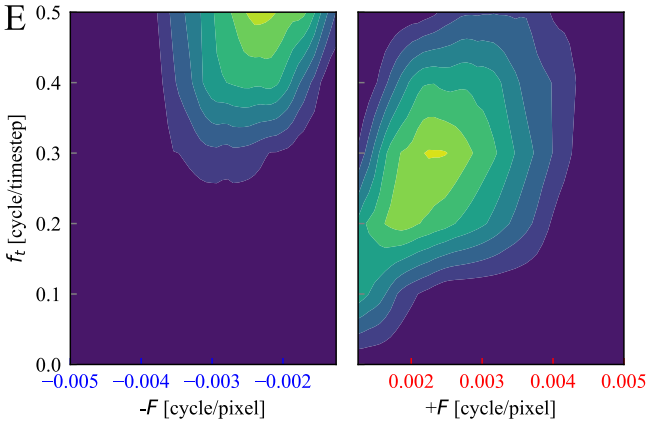


Figure 10: 2D representation of the measured filter response corresponding to Figure 9E. The positive and negative spatial frequency (F) axis can be seen in blue and red which correspond to the blue and red lines in Figure 9E. Two different Gaussian lobes can be identified tuned to different spatiotemporal frequencies.

C. Temporal bandwidth

For orientation θ and temporal frequency f_t, the bandwidth is defined as the width of the filter which provides an output above half the maximum filter output $\hat{r}_0$. This leads to a bandwidth in degrees $\Delta\theta_{1/2}$ and cycles per frame $\Delta f_{t_{1/2}}$ for orientation and temporal frequency respectively:

$$\Delta f_{t_{1/2}} = f_{t_{\max}} - f_{t_{\min}} \tag{8}$$

$$\Delta\theta_{1/2} = \theta_{\max} - \theta_{\min} \tag{9}$$

For spatial frequency F, the bandwidth is defined in terms of octaves as follows:

$$\Delta F_{1/2} = \log_2 \left(F_{\max} / F_{\min} \right) \tag{10}$$

Although we estimate the true parameters of the filters in the fitting process, due to the non-linear transform in Equation 5 the apparent bandwidth of the filter differs. The bandwidth is therefore measured based on the fitted Gabor filter response. In Figure 11 the bandwidth of 75% of the filters with the lowest normalized cost $\mathcal{L}_{norm}$ can be seen as we deem the fit of these filters sufficient to analyze. In this figure it can be seen that the

Inter Quartile Range (IQR) for spatial frequency bandwidth is between 1 and 2 octaves and the median orientation bandwidth is approximately $50°$. Lastly, this figure illustrates that the temporal frequency bandwidth is of large extent with a median of approximately 0.27 cycle per frame. We note that the network is able to narrow the extent of the filter response in the temporal domain using the non-linear transformation K, b, and the ReLU activation function. An illustration of this mechanism can be seen in Figure 12. In the top plot, the fitted filter and the measured response $\hat{r}$ over the temporal frequency range can be seen. In the middle plot the response without the gain K, bias b, and ReLU activation function can be seen. The extent of the half-magnitude profile is wider in the middle plot. Hence, the nonlinear transformation allows to reduce the filter's temporal extent, so that more precise motion can be measured. The bottom plot indicates what happens when more frames are added to the input and the other parameters are kept the same. The dotted line in the bottom plot corresponds to the dotted line in the top plot. Figure 11 suggests that an even narrower extent could be reached by feeding the network with more images over time than just the two subsequent images in FlowNetS. Note that a higher resolution in the frequency domain is beneficial as this allows for a more precise measurement of the flow.

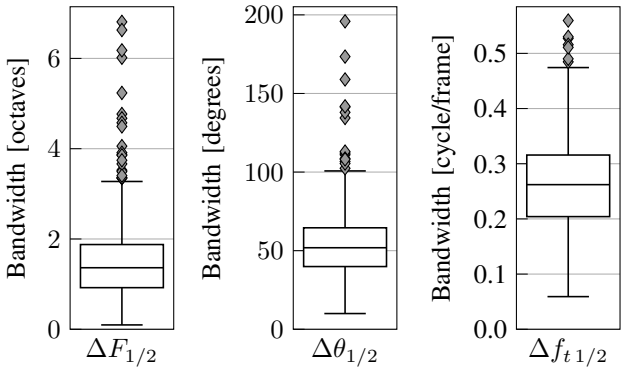


Figure 11: Bandwidth of spatial frequency F *left* orientation θ (*middle*) and temporal frequency f_t (*right*) for the 75% filters with the lowest normalized cost $\mathcal{L}_{norm}$ ($n = 397$). This corresponds to all filters upto and including the green filter in Figure 7. Bandwidth values are determined using the responses from the fitted Gabor filters.

V. Network response to dilation & rotation

In this section, the sensitivity of the filters in the $\mathrm{c}6$ layer of our FlowNetS to dilation and rotation is analysed. First, we explain the limitations of the spectral Gabor response profile fitting process and why we are not able to discern filters activating on translation, dilation, and rotation with this methodology. Also, we simulate the response of an occlusion filter to this methodology. Second, the methodology used to identify filters sensitive to dilation and rotation will be presented. Lastly, our results will be discussed.

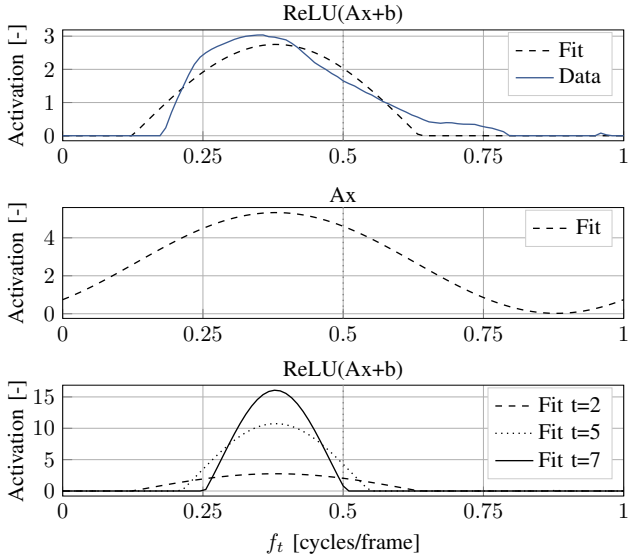


Figure 12: Illustration of how the network is able to decrease the extent of the filter response in the temporal domain. *Top:* Fit and measured data for the median filter of the FlowNetS network, corresponding to the bottom-right plot in Figure 7. *Middle:* The response of the fitted Gabor filter without the bias term and ReLU non-linearity. Note that the bandwidth of the signal before this transformation is wider. *Bottom:* Response of the fitted Gabor filter when the number of frames are increased. Note that filter is able to achieve a higher temporal resolution when more input frames are used.

It should be noted that Gabor translation filters [15] and occlusion filters [59] already have an analytical description in both the space-time and frequency domain. Such a description of dilation and rotation is, to the best of our knowledge, missing. Therefore, fitting responses of a filter to a dilation and rotation motion model requires a novel mathematical foundation which is outside of the scope of this work. In the following section the analytical description of dilation and rotation in the space-time is simply multiplied with a Gaussian to simulate a response.

A. Limitations of Gabor spectral response profile fitting

During the first part of the spectral response profile fitting process, a gridsearch is performed to find the peak response. In the subsequent fitting process three response lines are generated by varying either spatial frequency F_0, temporal frequency f_{t_0}, or orientation θ_0, whilst keeping phase φ_0 constant. This method only allows the measurement of the relative attenuation in amplitude with respect to the peak response $\hat{r}_0$. This procedure is sufficient for translation which can be defined as a single constant phase Gaussian in the 3D frequency spectrum and thus produces a Gaussian in response. In this section, we convolve translating plane waves with dilation, rotation, and occlusion filters to simulate their response. Due to the ReLU activation function the convolution of two translating plane waves at the same frequency, which

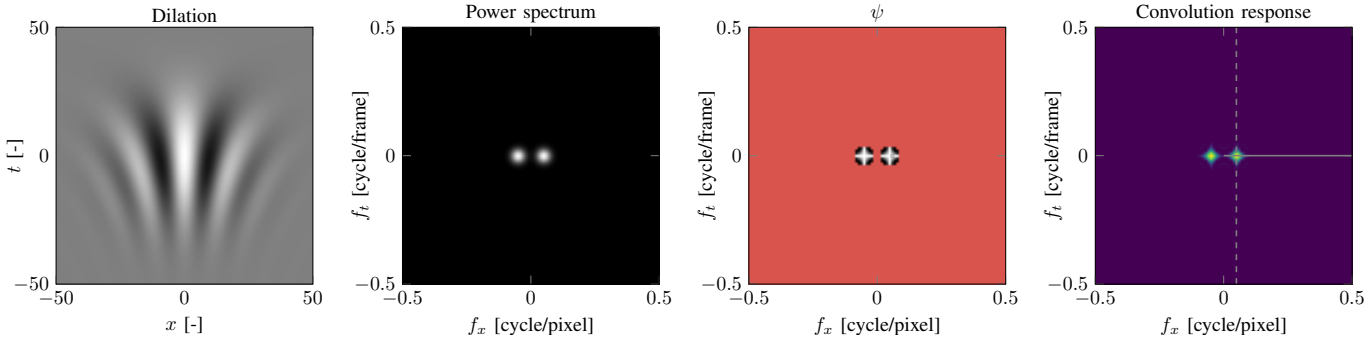


Figure 13: Simulation of convolution response of a dilation filter dw with a translating plane wave s evaluated with spatiotemporal frequencies at k integer multiples of the fundamental frequency. *Left:* Dilating wave d multiplied with a Gaussian w centered at the origin. *Middle left:* The power spectrum of the Fourier-transformed dilation filter. *Middle right:* The angle ψ indicating the phase difference between the Fourier components of the dilation filter and the translating plane wave s at the k integer multiples of the fundamental frequency. A larger phase difference corresponds to a darker color with black being equal to or greater than $\pi/2$. Also, a red mask is applied to frequency components with low power. *Right:* Convolution response between dilation filter dw and translating plane waves s. The lines indicate the Gaussian pattern perceived by our methodology.

are more than or equal to 90 degrees out-of-phase, will be zero (see Figure 5).

In order to determine which frequency components of dilation and rotation are more than 90 degrees out of phase, the Discrete Fourier Transform (DFT) [51] is used to transform a simulated space-time signal to a representation in the frequency domain. The DFT is defined as:

$$X[k] = \sum_{n=0}^{N-1} x[n] e^{-jk\frac{2\pi}{N}n} \tag{11}$$

where $k\frac{2\pi}{N}$ is the k-th discrete frequency and N the total length of the discrete signal. When the discrete signal is real, the DFT of the signal will result in a complex number:

$$X[k] = Ae^{i\varphi} \tag{12}$$

where the magnitude is denoted by A and the phase value by φ. Remember that a valid convolution of two tensors with equal size corresponds to the dot product of these two tensors. Furthermore, note that a convolution in the space-time domain equals to multiplication in the frequency domain according to the convolution theorem[51]. Because we evaluate the convolution response only at discrete frequencies of k integer multiples along the f_x, f_y, and f_t, axis only a single frequency component of S will contain power[3]. Then, if we define the k-th frequency component of the Fourier-transformed translating plane wave S as the complex vector $\mathbf{p}$, and the k-th frequency component of the Fourier transformation of the filter to be analysed (rotation, dilation, or occlusion) as $\mathbf{q}$, the phase difference between these two complex vectors is defined as the angle ψ and given by:

$$\psi = \cos^{-1}\left(\frac{\mathbf{p}\cdot\mathbf{q}}{|\mathbf{p}||\mathbf{q}|}\right) \tag{13}$$

[3]Not taking into account the complex conjugate frequency component.

where the maximum value of ψ is π, and values of ψ larger than or equal than $\pi/2$ will result in a zero response due to the ReLU non-linearity in Equation 5. Thus, ψ is a measure for how much out-of-phase the frequency components of the two signals are.

Convolution response: Dilation & rotation filters

In Figure 13, in the left-most plot, the space-time representation of a dilating wave d (see Equation 14) multiplied with a Gaussian window w can be seen. In the middle-left plot the power spectrum of this dilation filter can be seen. In the middle-right plot, the angle ψ indicates how much out of phase each frequency component of the occlusion filter is with the corresponding translating plane wave frequency component. The right-most plot indicates the convolution response of the dilation filter dw to translating plane waves s in which a diamond-like pattern emerges. Because we evaluate the responses along lines orthogonal to the peak response, the pattern perceived by our methodology is indicated by the dashed and solid gray line. These line patterns correspond to the line patterns in the 3D frequency space in Figure 4b. Thus, along the varying spatial frequency F range (solid line) and the varying temporal frequency range f_t (dashed line) a Gaussian will be perceived. Hence, given the limitations of our methodology we are not able to discern between dilation and translation filters.

Similarly, in Figure 14 in the two left-most columns the representation of a rotating wave c (see Equation 18) multiplied with a Gaussian window w in the space-time and frequency domain can be seen. Note that the 3D power spectrum in the second column is different from a spherical Gaussian. In the third column the angle ψ is depicted, and at high temporal frequencies (± 0.2 cycle per frame) the frequency components of the rotation filter cw and translating plane waves s are out-of-phase. Thus, as can be seen in the fourth

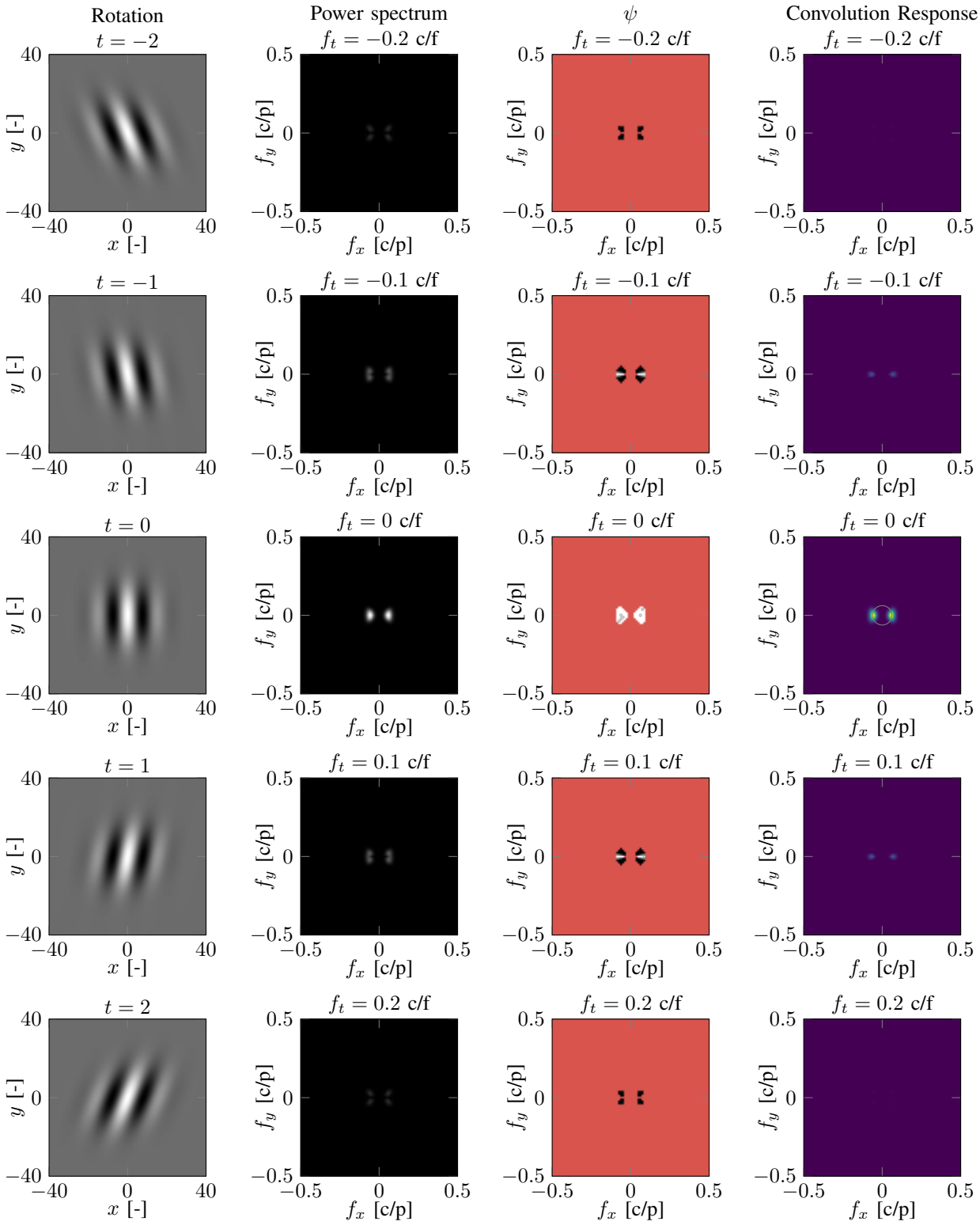


Figure 14: Simulation of convolution response of a rotation filter cw with a translating plane wave s evaluated with spatiotemporal frequencies at k integer multiples of the fundamental frequency. *Left column:* A cosine rotating wave c multiplied with a Gaussian w at five timesteps. *Middle left column:* The 3D power spectrum of the Fourier-transformed rotation filter. *Middle right column:* The angle ψ indicating the phase difference between the Fourier components of the rotation filter and the translating plane wave s at k integer multiples of the fundamental frequency. A larger phase difference corresponds to a darker color with black being equal to or greater than $\pi/2$. Also, a red mask is applied to frequency components with low power. *Right column:* Convolution response between rotation filter cw and translating plane waves s. The circle indicates the double lobe Gaussian pattern which will be perceived by our methodology by varying orientation θ_0.

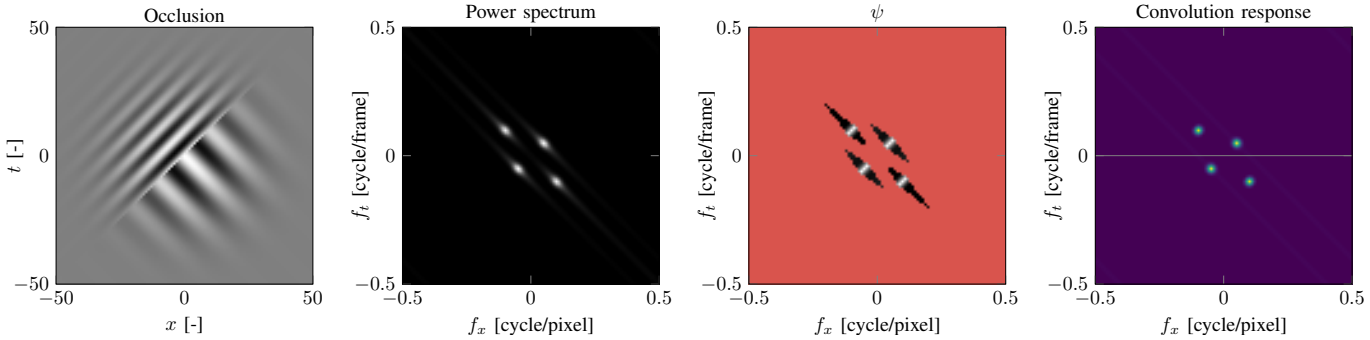


Figure 15: Simulation of convolution response of an occlusion filter with a translating plane wave s evaluated with spatiotemporal frequencies at k integer multiples of the fundamental frequency. *Left:* Example occlusion signal following the description of *Beauchemin et al.*[59] concerning occlusion in the frequency domain. The signal is composed of a Gaussian, two cosine translating plane waves s with different spatiotemporal frequencies f_x and f_t, and a Heaviside step function. *Middle left:* The power spectrum of the Fourier-transformed occlusion filter. In the power spectrum two pairs of Gaussian lobes can be seen with long 'tails' due to the Heaviside step function. *Middle right:* The angle ψ indicating the phase difference between the Fourier components of the rotation filter and the translating plane wave at k integer multiples of the fundamental frequency. A larger phase difference corresponds to a darker color with black being equal to or greater than $\pi/2$. Also, a red mask is applied to frequency components with low power. *Right:* Convolution response between the occlusion filter and translating plane waves s. The pattern above the gray line is perceived in Figure 10 as well.

column, these frequency components will not be detected by our methodology. Note that the pattern perceived along the varying θ_0, as indicated by the circle in the third row and the fourth column (which corresponds to the circle in 3D frequency space in Figure 4b), is two Gaussian lobes at opposite frequency. This pattern is similar to the convolution response of a cosine Gabor filter tuned to stationary patterns (zero temporal frequency). Therefore, our methodology is also not able to detect rotation filters.

Convolution response: Occlusion filters

Furthermore, we convolve an occlusion filter, using the description of *Beauchemin et al.* [59], with translating plane waves s. Occlusion in the spatiotemporal domain can be described as the combination of a Gaussian g, a Heaviside step function, and two translating plane waves translating with different frequencies (f_{x_0}, f_{y_0}, f_{t_0}) as can be seen in the right-most plot in Figure 15. The power spectrum of the Fourier-transformed filter can be described as two Gaussian filter pairs with 'tails' due to the Heaviside step function and can be seen in the middle-left plot. In the middle-left plot the angle ψ is depicted which demonstrates that the 'tails' have a large phase difference. Consequently, in the right-most plot, only the two pairs of Gaussian lobes will be detected using our methodology. The pattern above the solid gray line thus corresponds to two different Gaussian lobes tuned to different frequencies. This pattern can also be seen in Figure 10, thus making it likely that this filter is responsive to occlusion. However, it should be noted that we are not able to discern such a pattern from the superposition of two regular Gabor filter pairs tuned to different frequencies.

B. Methodology

In order to assess the sensitivity of the filters to dilation and rotation, two gridsearches will be performed. We assess the locations of the peak responses for filters which have a higher response to dilation or rotation than to translation. We do not classify a filter as either a rotation or dilation filter, since a filter can be sensitive to a composition of these respective motions. This is to be expected given the nature of the training dataset.

Dilation parametrization

As in [12], a dilating wave d is given by:

$$d(x, y, t) = \cos\left(2\pi F_0(x_r - \alpha x_r t) + \varphi_0\right) \tag{14}$$

where α denotes the dilation factor. The training dataset used to train FlowNetS defines scaling motion in terms of scaling factor h. Because FlowNetS only takes two frames as input, we define the relation between affine scaling factor h and dilation factor α between $t = 0$ and $t = 1$ as follows:

$$h = \frac{1}{1 - \alpha} \tag{15}$$

The gridsearch is performed for a scale factor h range of 0.5 till 2.0, as this range encapsulates the scaling factor h encountered during the training process. The scaling factor encountered is a combination of scaling motion present in the dataset and scaling factors applied by the online data augmentation process. The search space used for the dilation gridsearch can be found in Table V in Appendix B. In order to mitigate the effect of temporal aliasing (see Section IV), the search space is constrained so that the velocity of a point in the motion field is not more than half its spatial wavelength $\lambda_0/2$. It can be shown that the velocity of a point in the motion

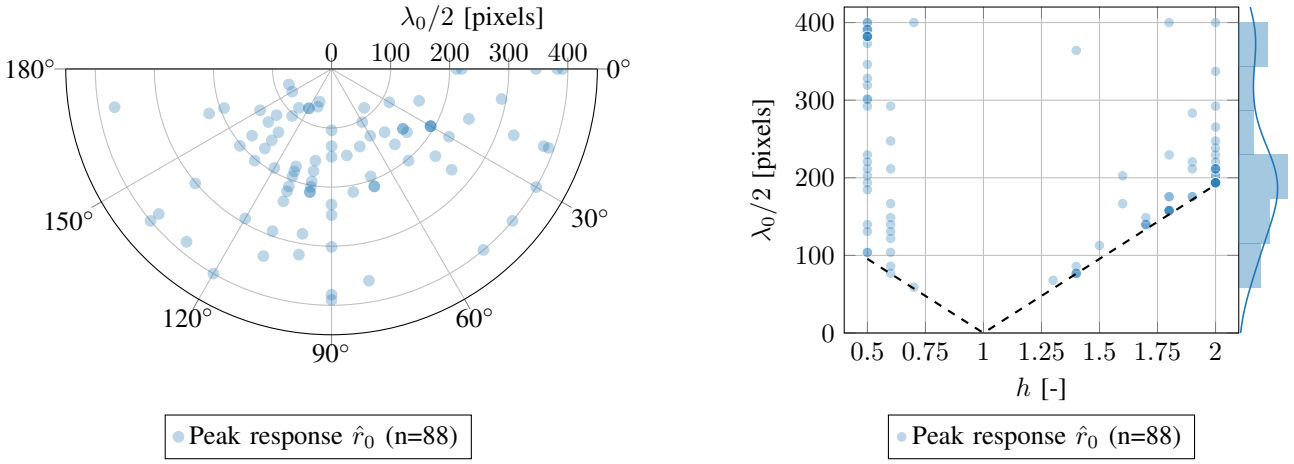


Figure 16: Location of peak response $\hat{r}_0$ per filter ($n = 88$) in the spatiotemporal frequency domain in response to dilating waves. Note that only filters are shown whose peak response $\hat{r}_0$ was higher than the maximum found in the translation gridsearch. *Left:* Half spatial wavelength $\lambda_0/2$ and initial orientation θ_0. *Right:* Half spatial wavelength $\lambda_0/2$ and scale factor h. The black dashed line indicates the temporal aliasing constraint given by Equation 17.

field for a dilating wave in Equation 14 at $t = 0$ is given by the following relation:

$$v = (\frac{1}{1-\alpha} - 1)x = (h-1)x \tag{16}$$

Then the temporal aliasing constraint for dilating waves is given by:

$$(h-1)x \leq \frac{1}{2}\lambda_0 \tag{17}$$

Rotation parametrization

For rotation the following equation is used to define the input:

$$c(x, y, t) = \cos\left(2\pi F_0 x_r(t) + \varphi_0\right) \tag{18}$$

where $x_r(t)$ varies with time and is defined as:

$$x_r(t) = x\cos(\theta_0 + \omega t) + y\sin(\theta_0 + \omega t) \tag{19}$$

where ω denotes the angular velocity in radians per frame. The search space for the rotation gridsearch can be found in Table VI in Appendix B. A constraint was added to the rotation gridsearch as well to limit the effect of temporal aliasing. The angular velocity ω can be related to a point at distance m from the center of rotation by $v = \omega m$. The maximum distance from the center of rotation to the edge is equal to half the receptive field size, which is 383 pixels in the $\mathrm{c}6$ layer of our FlowNetS. As the wave rotates around the center pixel and the distance from the center pixel to the outside pixel is 191.5 pixels, the velocity at this point should thus be lower than half the spatial wavelength. The constraint is given by the following relation:

$$\omega m_{\max} \leq \frac{1}{2}\lambda_0 \tag{20}$$

C. Results

Dilation results

The peak responses of filters which have a higher activation to dilation than to translation can be seen in Figure 16. We find that approximately 15% of the filters (i.e. 88 filters) respond more strongly to dilation than to translation. Furthermore, the filters show a radially dispersed pattern along the orientation axis (θ) as can be seen in the left plot in Figure 16. In the right plot in Figure 16 a peak in the distribution of half spatial wavelengths $\lambda_0/2$ can be seen near 200 pixels which is to be expected due to the fact that the average size of a chair in the training data is 200 pixels. Lastly, the peak responses are often close to the temporal aliasing limit and the maximum scaling value of the gridsearch. This is similar to the temporal peak response location for the translation gridsearch.

Rotation results

In Figure 17, the peak responses of the filters for the rotation gridsearch can be seen. The left plot in Figure 17 shows an angular dispersion of the peak responses along the orientation axis (θ_0). In the right plot in Figure 17 it can be seen that most filters are active near the temporal translation and temporal rotational aliasing limit. Also, a peak in the distribution of half spatial wavelengths $\lambda_0/2$ can be identified around 250 pixels, which is slightly higher than expected from the training data. A possible explanation for this discrepancy is that rotation is actually a 3D motion and thus the scale should also be limited along its radial axis. Approximately 45% of the filters activate more on rotation than on translation. A possible explanation for this high number of filters is that we do not limit the wavelength along the axis of rotation. The points in the motion field at the far end of the receptive field then move with a very high velocity and therefore the response of the filters in our FlowNetS will be higher.

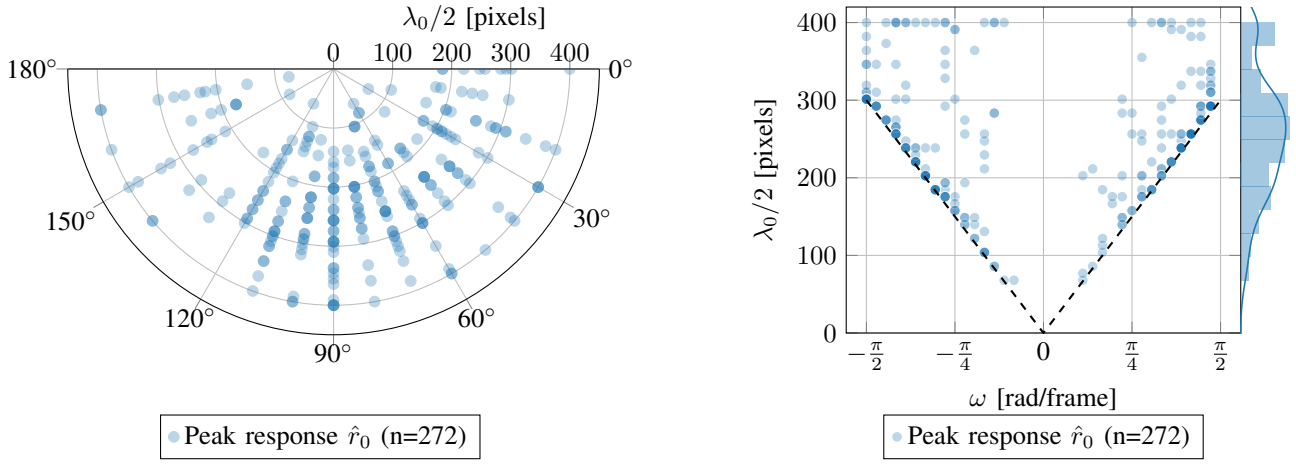


Figure 17: Location of peak response $\hat{r}_0$ per filter ($n = 272$) in the spatiotemporal frequency domain in response to rotating waves. Note that only filters are shown whose peak response $\hat{r}_0$ was higher than the maximum found in the translation gridsearch. *Left:* Half spatial wavelength $\lambda_0/2$ and initial orientation θ_0. *Right:* Half spatial wavelength $\lambda_0/2$ and angular temporal frequency ω. The black dashed line indicates the temporal aliasing constraint given by Equation 20.

VI. Solving the Aperture Problem

The previous section showed that individual neurons in the `c6` layer of FlowNetS act as Gabor-like filters for translation, rotation, and dilation. In this section, we study the aperture problem and the flow refinement process. We first explain our methodology and then present the results.

A. Methodology

In order to determine until what scale of input stimuli FlowNetS can resolve the aperture problem three different versions of FlowNetS are trained under the same circumstances with varying receptive field sizes. The receptive field size is defined as the region in the input images which affects the value of the feature map at a particular layer and feature map location. Therefore, we modify the filter size of the convolutional kernels in layer `c6`. Layer `c6` is composed of two convolutional layers `c6_0` and `c6_1`. FlowNetS has two 3x3 kernels for layers `c6_0` and `c6_1`. We train two models with kernels sizes (1x1, 3x3) (1x1, 1x1) in the last two layers in the contracting part of the network. We name these models FlowNetXS and FlowNetXXS, and the receptive field size of the `f6` flow map is 255 pixels and 191 pixels respectively. Our FlowNetS has a receptive field size of 383 pixels. The expanding part of FlowNetS features upconvolutional layers which also increases the receptive field size. The receptive field size of the `f2` flow map is calculated to be 551, 615 and 743 pixels for FlowNetXXS, FlowNetXS and FlowNetS respectively.

As input, a diagonally translating bar with magnitude $|\mathbf{u}|=$ 64 pixels is used. For an input of 1024x768 pixels, the coarsest flow map `f6` size is 16x12 after 6 convolutions with stride 2. We determine the error of the flow estimate at location $(8, 6)$ of `f6`, which can be seen in Figure18a marked by the red square, and therefore center the bar in the input image accordingly.

B. Results

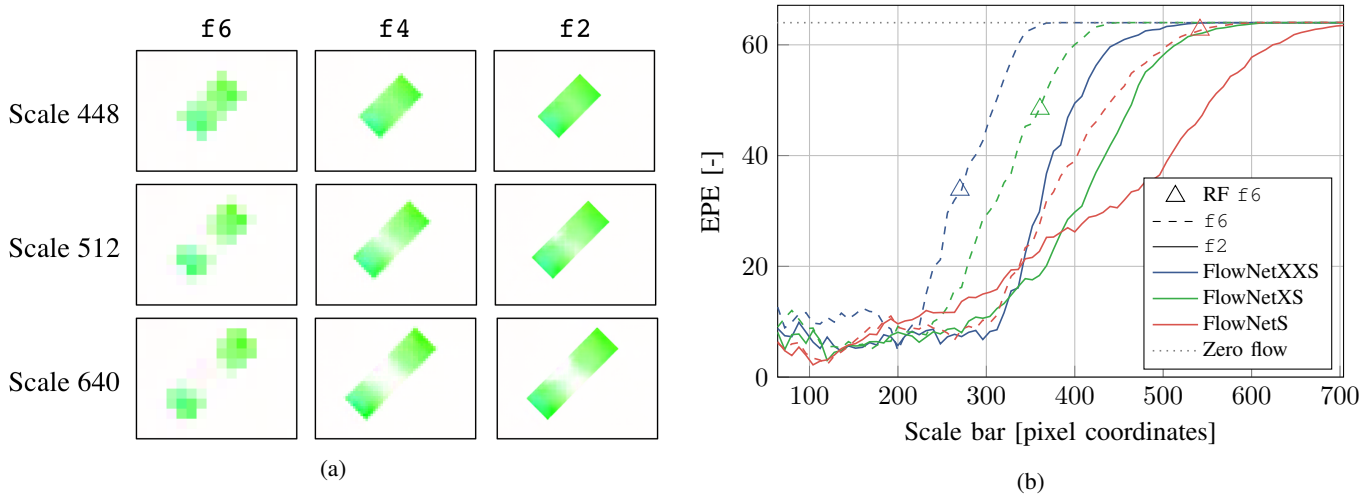
Two translating bar image input pairs are fed into the network, one pair translating upward right and one translating downward left. In Figure 18a, the responses to the upward right translating image pair are shown. The flow field color coding used is similar to *Baker et al.*[69] and can be found in Figure 19 in Appendix C. Column-wise, the `f6`, `f4` and `f2` flow maps can be seen. The first row shows that the flow becomes more refined. Row-wise the scale of the bar is increased. The second and third row indicate that the network is able to extrapolate motion cues from the edges of the bar towards the center. The receptive field size in the expanding part of the network increases due to the size of the upconvolutional kernels.

In Figure 18b, the average End-Point-Error (EPE) of FlowNetS, FlowNetXS, and FlowNetXSS can be seen in response to two diagonally translating input image pairs. The region in which the flow is measured is indicated by the red square outline in Figure 18a. From this figure it can be seen that the ability of the network to resolve the aperture problem is related to its receptive field size, and the networks with larger receptive field sizes are able to resolve the aperture problem at larger scales.

VII. Discussion and Future Work

A. Impact on Computer Vision

Due to the emergence of Gabor-like filters in other learning-based methods our work started out with the expectation of also finding Gabor-like filters. Traditional Gabor filters for optical flow estimation had certain disadvantages. They deal badly with deviations from translation, varying contrast due to changing lighting conditions, and are subject to the uncertainty relation, which corresponds to the balance between localization of the stimuli in the spatial domain and resolution in the frequency domain. FlowNetS copes with all of these



(a) (b)

Figure 18: Response of FlowNetS, FlowNetXS, and FlowNetXXS to diagonally translating bars ($|\mathbf{u}|= 64$). (a) Column-wise, the `f6`, `f4` and `f2` of FlowNetS in response to upward-right diagonally translating bars. Row-wise, the output flow corresponding to bars of different scales. The red square denotes the output region used for evaluating the error. (b) End-Point-Error (EPE) versus scale of the bar in pixel coordinates. The EPE is the average between a bar moving upward to the right and downward to the left with the same motion magnitude. The ground truth is $|\mathbf{u}|$=64 pixels and an EPE at this level corresponds to no motion detected by the network. RF `f6` indicates the diagonal receptive field size in pixel coordinates corresponding to the `6` flow map. The diagonal receptive field sizes in pixel coordinates of the `f2` flow map is omitted because they are greater than the largest evaluated translating bar.

issues. We have shown that deviations from translations are dealt with by additional neurons that are sensitive to deviations from translation. Moreover, *Mayer et al.* [70] showed that FlowNet is able to cope with varying contrast over time due to changing lighting conditions. Lastly, we have demonstrated that FlowNetS is able to achieve a better spatial localization of motion cues in the expanding part of the network thus overcoming the uncertainty relation.

Based on the high similarity of the neurons to the fitted Gabor-like filters, it would be worth studying a hybrid approach, in which there is a fixed Gabor filter bank (extended with rotation and dilation features), followed by a convolutional multi-layer loss flow refinement part. This would reduce training time and greatly increase computational efficiency.

In terms of accuracy, FlowNetS did not reach the levels of state-of-the-art methods. For example, it has poor performance on sub-pixel flow [22]. One reason for this might be the large number of strides utilized before the initial flow prediction is made. Strides reduce the amount of spatial information available. Hence, future work should investigate the effects of strides on the performance of FlowNetS.

Also, our analysis shows that a Gabor filter based on two frames indeed has a large temporal frequency bandwidth, and hence limited performance concerning flow velocity estimation. This is narrowed somewhat by the non-linear transformations due to the ReLU activation function and bias term. However, our analysis indicates that this could be further improved by using more frames and thus providing more temporal information to the network.

B. Impact on biology

We have used and extended methods from neuropsychology for determining the types of filters represented by neurons in FlowNetS' deep `c6` layer. The analysis gave very similar results to those on neurons in the mammalian visual cortex. First, many filter responses fit very accurately with Gabor filters that capture translational motion. Second, the spatial and orientation bandwidth statistics show similarity to bandwidths found in the mammalian visual cortex. Regarding spatial frequency bandwidth, we report a median bandwidth of 1.36 octaves, while *De Valois et al.* [71] found a spatial frequency bandwidth of 1.4 octaves in the macaque visual cortex. Similarly, we find a median orientation bandwidth of 52 degrees, while *De Valois et al.* [72] report a orientation half-magnitude profile width of 65 degrees. This may be due to the network having been subjected to optical flow statistics as also perceived by animals. Third, as in neuropsychological experiments [52], we observed that some neurons respond poorly to translating plane waves. In fact, also the reverse-correlation does not provide an adequate signal-to-noise ratio for the reconstruction of spatiotemporal receptive field profiles of these neurons [52]. Our analysis shows that such poor response may be due to the neurons being sensitive to more complex motions such as dilation and rotation. Indeed, in the human brain, channels sensitive to dilation have been found [73]. However, this did not provide conclusive evidence of *neurons* sensitive to dilation. Our analysis and results suggest that it is worth looking for dilation- and rotation-sensitive neurons in animal brains. In fact, one could even extend the

analysis to also check for shear, as this forms an additional basis for the flow field derivatives [74].

VIII. Conclusion

In this work we have demonstrated that FlowNetS learns a bank of spatiotemporal Gabor filters, tuned to different spatiotemporal frequencies and values of phase, by means of a spectral response fitting approach used in neuropsychology. Moreover, our results indicate that the network also learns a large number of filters that are sensitive to dilation and rotation. Furthermore, we have demonstrated that the receptive field size is linked to the scale at which the network can resolve the aperture problem and that the expanding part of the network performs a filling-in function which is similar in function to the filling-in process which occurs in mammal vision systems.

While we were able to identify the response of filters to translation, dilation, and rotation separately, we were not able to show the exact motion patterns causing the maximum activation of a filter. In reality some filters are most likely sensitive to a combination of these motions. We are not able to quantify this due to limitations in our methodology. Future work should improve our methodology to be able to identify more complex motion patterns like compositions of affine and 3D motion. The latter of which is present in more realistic synthetic training datasets like FlyingThings [75].

The novel insights in FlowNetS provide avenues for improving its performance, such as using smaller strides and providing the network with more input images. Additionally, it provides interesting avenues for neuropsychological research, specifically to use our extended method to investigate if animal brains have dilation- and rotation-sensitive neurons as well.

In this work we studied FlowNetS and we believe this model is prototypical for fully convolutional networks used for optical flow determination due to its generic architecture. This being said, it would be useful in the future to also study other networks like SpyNet [38] using our methodology.

References

[1] J. J. Gibson, *The perception of the visual world.* Oxford, England: Houghton Mifflin, 1950.

[2] J. Feng, "Computational neuroscience: a comprehensive approach." Chapman and Hall/CRC, 2003.

[3] A. Borst, J. Haag, and D. F. Reiff, "Fly Motion Vision," *Annual Review of Neuroscience*, vol. 33, no. 1, pp. 49–70, jun 2010.

[4] A. Borst and M. Helmstaedter, "Common circuit design in fly and mammalian motion vision," pp. 1067–1076, aug 2015.

[5] I. Kajo, A. S. Malik, and N. Kamel, "An evaluation of optical flow algorithms for crowd analytics in surveillance system," in *International Conference on Intelligent and Advanced Systems, ICIAS 2016*. Institute of Electrical and Electronics Engineers Inc., jan 2017.

[6] G. de Croon, H. W. Ho, C. D. Wagter, E. van Kampen, B. Remes, and Q. P. Chu, "Optic-Flow Based Slope Estimation for Autonomous Landing," *International Journal of Micro Air Vehicles*, vol. 5, no. 4, pp. 287–297, 2013.

[7] H. W. Ho, C. De Wagter, B. D. Remes, and G. C. De Croon, "Optical flow for self-supervised learning of obstacle appearance," in *IEEE International Conference on Intelligent Robots and Systems*, vol. 2015-Decem. Institute of Electrical and Electronics Engineers Inc., dec 2015, pp. 3098–3104.

[8] K. Chen and D. A. Lorenz, "Image sequence interpolation based on optical flow, segmentation, and optimal control," *IEEE Transactions on Image Processing*, vol. 21, no. 3, pp. 1020–1030, mar 2012.

[9] P. Anandan, "A computational framework and an algorithm for the measurement of visual motion," *International Journal of Computer Vision*, vol. 2, no. 3, pp. 283–310, 1989.

[10] A. Singh, *Optic Flow Computation: A Unified Perspective*. IEEE Computer Society Press Los Alamitos, 1991.

[11] D. J. Heeger, "Optical flow using spatiotemporal filters," *International Journal of Computer Vision*, vol. 1, no. 4, pp. 279–302, jan 1988.

[12] D. Fleet and A. Jepson, "Computation of normal velocity from local phase information," in *Proceedings CVPR'89: IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1989, pp. 379–386.

[13] B. K. Horn and B. G. Schunck, "Determining optical flow," *Artificial Intelligence*, vol. 17, no. 1-3, pp. 185–203, aug 1981.

[14] B. D. Lucas and T. Kanade, "An Iterative Image Registration Technique with an Application to Stereo Vision," 1981.

[15] D. Gabor, "Theory of communication," *Journal of the Institution of Electrical Engineers - Part I: General*, vol. 94, no. 73, pp. 58–58, 1945.

[16] S. Ullman, *The interpretation of visual motion*. MIT Press, 1979.

[17] O. P. Agrawal, "Formulation of Euler-Lagrange equations for fractional variational problems," *Journal of Mathematical Analysis and Applications*, vol. 272, no. 1, pp. 368–379, aug 2002.

[18] H. Zimmer, A. Bruhn, and J. Weickert, "Optic flow in harmony," *International Journal of Computer Vision*, vol. 93, no. 3, pp. 368–388, 2011.

[19] Y. Mileva, A. Bruhn, and J. Weickert, "Illumination-Robust Variational Optical Flow with Photometric Invariants," *Pattern Recognition*, pp. 152–162, 2007.

[20] A. Dosovitskiy, P. Fischery, E. Ilg, P. Hausser, C. Hazirbas, V. Golkov, P. V. D. Smagt, D. Cremers, and T. Brox, "FlowNet: Learning optical flow with convolutional networks," *Proceedings of the IEEE International Conference on Computer Vision*, vol. 2015 Inter, pp. 2758–2766, 2015.

[21] Z. Tu, W. Xie, D. Zhang, R. Poppe, R. C. Veltkamp, B. Li, and J. Yuan, "A survey of variational and CNN-based optical flow techniques," *Signal Processing: Image Communication*, vol. 72, pp. 9–24, mar 2019.

[22] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox, "FlowNet 2.0: Evolution of optical flow estimation with deep networks," *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, vol. 2017-Janua, pp. 1647–1655, 2017.

[23] D. Sun, X. Yang, M.-Y. Liu, and J. Kautz, "Models Matter, So Does Training: An Empirical Study of CNNs for Optical Flow Estimation," *arXiv preprint arXiv:1809.05571*, pp. 1–15, 2018.

[24] T.-W. Hui, X. Tang, and C. C. Loy, "A Lightweight Optical Flow CNN - Revisiting Data Fidelity and Regularization," *arXiv preprint arXiv:1903.07414*, pp. 1–13, 2019.

[25] J. P. Jones, A. Stepnoski, and L. A. Palmer, "The two-dimensional spectral structure of simple receptive fields in cat striate cortex," *Journal of Neurophysiology*, vol. 58, no. 6, pp. 1212–1232, 1987.

[26] D. G. Albrecht, R. L. De Valois, and L. G. Thorell, "Visual cortical neurons: Are bars or gratings the optimal stimuli?" *Science*, vol. 207, no. 4426, pp. 88–90, jan 1980.

[27] J. P. Jones and L. A. Palmer, "An evaluation of the two-dimensional Gabor filter model of simple receptive fields in cat striate cortex," *Journal of neurophysiology*, vol. 58, no. 6, pp. 1233–1258, 1987.

[28] G. C. DeAngelis, I. Ohzawa, and R. D. Freeman, "Receptive-field dynamics in the central visual pathways," pp. 451–458, 1995.

[29] J. H. Van Hateren and D. L. Ruderman, "Independent component analysis of natural image sequences yields spatio-temporal filters similar to simple cells in primary visual cortex," *Proceedings of the Royal Society B: Biological Sciences*, vol. 265, no. 1412, pp. 2315–2320, 1998.

[30] B. A. Olshausen, "Learning sparse, overcomplete representations of time-varying natural images," in *IEEE International Conference on Image Processing*, vol. 1, 2003, pp. 41–44.

[31] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert, "High Accuracy Optical Flow Estimation Based on a Theory for Warping," in *European conference on computer vision*. Springer, 2004, pp. 25–36.

[32] T. Brox and J. Malik, "Large displacement optical flow: Descriptor matching in variational motion estimation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 3, pp. 500–513, 2011.

[33] J. Revaud, P. Weinzaepfel, Z. Harchaoui, and C. Schmid, "EpicFlow: Edge-preserving interpolation of correspondences for optical flow," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 07-12-June, pp. 1164–1172, 2015.

[34] P. Weinzaepfel, J. Revaud, Z. Harchaoui, and C. Schmid, "DeepFlow: Large displacement optical flow with deep matching," *Proceedings of the IEEE International Conference on Computer Vision*, no. Section 2, pp. 1385–1392, 2013.

[35] S. Zweig and L. Wolf, "InterpoNet, a brain inspired neural network for optical flow dense interpolation," in *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, vol. 2017-Janua, nov 2017, pp. 6363–6372.

[36] X. Xiang, M. Zhai, R. Zhang, Y. Qiao, and A. El Saddik, "Deep Optical Flow Supervised Learning With Prior Assumptions," *IEEE Access*, vol. 6, pp. 43 222–43 232, aug 2018.

[37] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*, vol. 9351. Springer, 2015, pp. 234–241.

[38] A. Ranjan and M. J. Black, "Optical flow estimation using a spatial pyramid network," *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, vol. 2017-Janua, pp. 2720–2729, 2017.

[39] D. Teney and M. Hebert, "Learning to extract motion from videos in convolutional neural networks," in *Asian Conference on Computer Vision*. Springer, Cham, 2016, pp. 412–428.

[40] E. Ilg, C. Ozgun, S. Galesso, A. Klein, O. Makansi, F. Hutter, and T. Brox, "Uncertainty Estimates and Multi-Hypotheses Networks for Optical Flow," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 652–667.

[41] A. Ranjan, J. Janai, A. Geiger, and M. J. Black, "Attacking Optical Flow," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 2404–2413.

[42] J. T. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller, "Striving for Simplicity: The All Convolutional Net," *arXiv preprint arXiv:1412.6806*, 2014.

[43] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *European conference on computer vision*. Springer, Cham, 2014, pp. 818–833.

[44] D. Erhan, Y. Bengio, A. Courville, and P. Vincent, "Visualizing higher-layer features of a deep network," *Bernoulli*, no. 1341, pp. 1–13, 2009.

[45] C. Olah, A. Mordvintsev, and L. Schubert, "Feature Visualization," nov 2017.

[46] A. Mordvintsev, "Inceptionism: Going Deeper into Neural Networks," 2015. [Online]. Available: https://ai.googleblog.com/2015/06/inceptionism-going-deeper-into-neural.html

[47] A. Nguyen, A. Dosovitskiy, J. Yosinski, T. Brox, and J. Clune, "Synthesizing the preferred inputs for neurons in neural networks via deep generator networks," in *Advances in Neural Information Processing Systems*, 2016, pp. 3395–3403.

[48] D. Wei, B. Zhou, A. Torrabla, and W. Freeman, "Understanding Intra-Class Knowledge Inside CNN," *arXiv preprint arXiv:1507.02379*, 2015.

[49] S. Marcelja, "Mathematical description of the responses of simple cortical cells." *Journal of the Optical Society of America*, vol. 70, no. 11, pp. 1297–1300, nov 1980.

[50] J. G. Daugman, "Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters," *Journal of the Optical Society of America A*, vol. 2, no. 7, p. 1160, jul 1985.

[51] R. Bracewell, *The Fourier transform and its applications*, vol. 31999 ed. New York: McGraw-Hill, 1986.

[52] G. C. Deangelis, I. Ohzawa, and R. D. Freeman, "Spatiotemporal Organization of Simple-Cell Receptive Fields in the Cat's Striate Cortex. I. General Characteristics and Postnatal Development," *Journal of Neurophysiology*, vol. 69, no. 4, 1993.

[53] N. Petkov and E. Subramanian, "Motion detection, noise reduction, texture suppression, and contour enhancement by spatiotemporal Gabor filters with surround inhibition," *Biological Cybernetics*, vol. 97, no. 5-6, pp. 423–439, 2007.

[54] L. A. Palmer and T. L. David, "Receptive-field structure in cat striate cortex," *Journal of Neurophysiology*, vol. 46, no. 2, pp. 260–276, 1981.

[55] G. C. DeAngelis, I. Ohzawa, and R. D. Freeman, "Spatiotemporal organization of simple-cell receptive fields in the cat's striate cortex. II. Linearity of temporal and spatial summation," *Journal of Neurophysiology*, vol. 69, no. 4, pp. 1118–1135, 1993.

[56] J. P. Jones and L. A. Palmer, "The two-dimensional spatial structure of simple receptive fields in cat striate cortex," *Journal of Neurophysiology*, vol. 58, no. 6, pp. 1187–1211, 1987.

[57] J. M. Zanker, "Theta motion: a paradoxical stimulus to explore higher order motion extraction," *Vision Research*, vol. 33, no. 4, pp. 553–569, mar 1993.

[58] D. J. Fleet and K. Langley, "Computational analysis of non-Fourier motion," *Vision research*, vol. 34, no. 22, pp. 3057–3079, 1994.

[59] S. Beauchemin and J. Barron, "The frequency structure of one-dimensional occluding image signals," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 2, pp. 200–206, 2000.

[60] A. Araujo, W. Norris, and J. Sim, "Computing Receptive Fields of Convolutional Neural Networks," nov 2019. [Online]. Available: https://distill.pub/2019/computing-receptive-fields

[61] H. Komatsu, "The neural mechanisms of perceptual filling-in," *Nature Reviews Neuroscience*, vol. 7, no. 3, pp. 220–231, feb 2006.

[62] R. Von Der Heydt, H. S. Friedman, and H. Zhou, "Searching for the Neural Mechanism of Color Filling-In," *Filling-In: From Perceptual Completion to Cortical Reorganization*, pp. 106–127, 2003.

[63] J. Poort, F. Raudies, A. Wannig, V. A. Lamme, H. Neumann, and P. R. Roelfsema, "The role of attention in figure-ground segregation in areas V1 and V4 of the visual cortex," *Neuron*, vol. 75, no. 1, pp. 143–156, 2012.

[64] M. Liang and X. Hu, "Recurrent convolutional neural network for object recognition," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 07-12-June, 2015, pp. 3367–3375.

[65] G. Kanizsa, "Margini Quasi-Percettivi in Campi con Stimolazione Omogenea," *Rivista di Psicologia*, vol. 49, pp. 7–30, 1955.

[66] A. R. Conn, N. I. M. Gould, and P. L. Toint, *Trust region methods*. Siam, 2000, vol. 1.

[67] Y.-x. Yuan, "A Review of Trust Region Algorithms for Optimization," *Iciam*, vol. 99, no. 1, pp. 271–282, 2000.

[68] L. Lu, Y. Shin, Y. Su, and G. E. Karniadakis, "Dying ReLU and Initialization: Theory and Numerical Examples," *arXiv preprint arXiv:1903.06733*, 2019.

[69] S. Baker, D. Scharstein, J. P. Lewis, S. Roth, M. J. Black, and R. Szeliski, "A database and evaluation methodology for optical flow," *International Journal of Computer Vision*, vol. 92, no. 1, pp. 1–31, 2011.

[70] N. Mayer, ·. E. Ilg, P. Fischer, ·. C. Hazirbas, D. Cremers, ·. A. Dosovitskiy, ·. T. Brox, E. Ilg, C. Hazirbas, A. Dosovitskiy, and T. Brox, "What Makes Good Synthetic Training Data for Learning Disparity and Optical Flow Estimation?" *International Journal of Computer Vision*, vol. 126, pp. 942–960, 2018.

[71] R. L. De Valois, D. G. Albrecht, and L. G. Thorell, "Spatial frequency selectivity of cells in macaque visual cortex," *Vision Research*, vol. 22, no. 5, pp. 545–559, 1982.

[72] R. L. De Valois, E. William Yund, and N. Hepler, "The orientation and direction selectivity of cells in macaque visual cortex," *Vision Research*, vol. 22, no. 5, pp. 531–544, 1982.

[73] D. Regan and K. I. Beverley, "Looming detectors in the human visual pathway," *Vision Research*, vol. 18, no. 4, pp. 415–421, jan 1978.

[74] H. C. Longuet-Higgins and K. Prazdny, "The interpretation of a moving retinal image," *Proceedings of the Royal Society of London. Series B. Biological Sciences*, vol. 208, no. 1173, pp. 385–397, 1980.

[75] N. Mayer, E. Ilg, P. Hausser, P. Fischer, D. Cremers, A. Dosovitskiy, and T. Brox, "A Large Dataset to Train Convolutional Networks for Disparity, Optical Flow, and Scene Flow Estimation," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2016-Decem, 2016, pp. 4040–4048.

[76] D. J. Butler, J. Wulff, G. B. Stanley, and M. J. Black, "A naturalistic open source movie for optical flow evaluation," in *European conference on computer vision*. Springer, 2012, pp. 611–625.

APPENDIX A
MODEL DETAILS

Table I outlines the full details of our version of FlowNetS. The name of the 'conv', 'flow' and 'predict_flow' layer is abbreviated to `c`, `f` and `pf` respectively throughout the paper. The output of the 'predict_flow' layer is called 'flow'.

In Table II a performance comparison between our version of FlowNetS and the original version of *Dosovitskiy et al.* [20] can be found on the FlyingChairs [20] and MPI sintel [76] datasets.

APPENDIX B
GRID SEARCH PARAMETERS

In Table III the parameter ranges used for the translation gridsearch can be found. Furthermore, in Table IV the parameters used for the spectral Gabor response profile fitting can be found. These are the three ranges along which the output of the FlowNetS `c6` filters will be evaluated. The parameter ranges used for the dilation gridsearch can found in Table V. Note that due to rotational symmetry the initial orientation θ_0 only varies from 0 to 170 degrees. The parameters used for the rotation gridsearch can be found in Table VI. Also for this gridsearch the θ_0 is constrained from 0 to 170 degrees due to rotational symmetry. Note that the half spatial wavelength $\lambda/2$ can be transformed to spatial frequency F_0 using the relation $F_0 = \frac{1}{2\lambda}$.

APPENDIX C
FLOW FIELD COLOR CODING

In Figure 19 the flow field color coding from *Baker et al.* [69] is shown. Note that a pixel coordinate system is used which defines the positive y-axis downward.

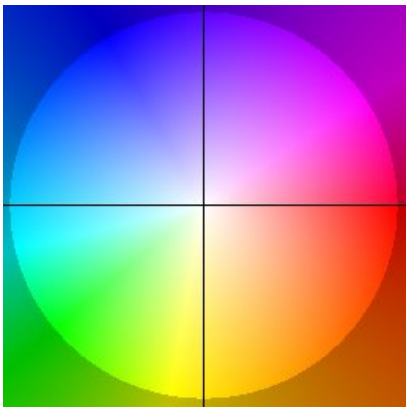


Figure 19: Flow field color coding similar to *Baker et al.* [69].

Name	Kernel	Stride	Padding	Ch I/O	In Res	Out Res	Input
conv1	7x7	2	3	6/64	512x384	256x192	Images
conv2	5x5	2	2	64/128	256x192	128x96	conv1
conv3_0	5x5	2	2	128/256	128x96	64x48	conv2
conv3_1	3x3	1	1	256/256	64x48	64x48	conv3_0
conv4_0	3x3	2	1	256/512	64x48	32x24	conv3_1
conv4_1	3x3	1	1	512/512	32x24	32x24	conv4_0
conv5_0	3x3	2	1	512/512	32x24	16x12	conv4_1
conv5_1	3x3	1	1	512/512	16x12	16x12	conv5_0
conv6_0	3x3	2	1	512/1024	16x12	8x6	conv5_1
conv6_1	3x3	1	1	1024/1024	8x6	8x6	conv6_0
predict_flow6	1x1	1	1	1024/2	8x6	8x6	conv6_1
upconv5	4x4	2	1	1024/512	8x6	16x12	conv6_1
predict_flow5	1x1	1	1	1026/2	16x12	16x12	upconv5+conv5_1 +flow6
upconv4	4x4	2	1	1026/256	16x12	32x24	upconv5+conv5_1 +flow6
predict_flow4	1x1	1	1	770/2	32x24	32x24	upconv4+conv4_1 +flow5
upconv3	4x4	2	1	770/128	32x24	64x48	upconv4+conv4_1 +flow5
predict_flow3	1x1	1	1	386/2	64x48	64x48	upconv3+conv3_1 +flow4
upconv2	4x4	2	1	386/64	64x48	128x96	upconv3+conv3_1 +flow4
predict_flow2	1x1	1	1	192/2	128x96	128x96	upconv2+conv2+flow3

Table I: Full details of our version of FlowNetS. Note that the expansive part of the network starts at 'flow6'.

Model name	Model details	FlyingChairs test [EPE]	MPI Sintel clean train [EPE]	MPI Sintel Final train [EPE]
FlowNetS [20]	original	2,71	4,50	5,45
FlowNetS-ours	ReLu activation function, `pf` layers with 1x1 kernels and no bias term, 300K training iterations, no data augmentation between frames	3,10	5,06	5,81

Table II: Performance comparison between the original version of FlowNetS and ours on the MPI-Sintel [76] and FlyingChairs [20] datasets.

Parameter	Unit	Range [start, stop, step size]
$\lambda_0/2$	pixels	[16, 800, 16]
θ_0	degrees	[0, 350, 10]
f_{t_0}	cycle per frame	[0.0, 0.5, 0.01]
φ_0	degrees	[-180, 170, 10]

Table III: Parameter ranges used for the translating plane wave gridsearch.

Parameter	Unit	Range [start, stop, number of points]
$\lambda_0/2$	cycle per pixel	[16, 800, 50]
θ_0	degrees	[0, 350, 36]
f_{t_0}	cycle per frame	[-0.5, 0.5, 50]

Table IV: Parameter ranges used for the Gabor spectral profile fitting process.

Parameter	Unit	Range [start, stop, step size]
$\lambda_0/2$	pixels	[50, 400, 10]
θ_0	degrees	[0, 170, 10]
s_f	-	[0.5, 2.0, 0.1]
φ_0	degrees	[-180, 170, 10]

Table V: Parameter ranges used for the dilating wave gridsearch.

Parameter	Unit	Range [start, stop, step size]
$\lambda_0/2$	pixels	[50, 400, 10]
θ_0	degrees	[0, 170, 10]
ω_0	cycle per frame	[-0.5, 0.5, 0.1]
φ_0	degrees	[-180, 170, 10]

Table VI: Parameter ranges used for the rotation gridsearch. The angular velocity ω is limited between -0.5 and 0.5 cycle per sample which corresponds to $-\frac{1}{2}\pi$ and $\frac{1}{2}\pi$ radians per frame respectively.

II

Literature Study

2

Time-varying Image Formation

The concept of optical flow was first defined by Gibson, 1950 as follows: consider a moving pattern of light which hits the retina when an observer moves relative to the environment. The distribution of these patterns is called *optical flow*. This information can be exploited by the observer for ego-motion and scene interpretation. However, the goal of optical flow is application dependent. Baker et al., 2011 state, based on the taxonomy of B. Horn, 1986, that the motion field is the 2D projection of the 3D motion surfaces in the world. The optical flow is the apparent motion of the brightness patterns in the image. These two are not always the same, and the goal is application dependent. Regarding applications where the motion is used to interpret or reconstruct the 3D world, the motion field is what is desired. In the scope of this thesis, the motion field is of interest. Henceforth, if a reference is made to optical flow, the motion field is what is referred to. Note that transparency is excluded in this analysis, which requires the estimation of multiple motions per pixel. This occurs when am object is translucent enough to emit light through its surface. In this chapter, a derivation of optical flow will be given. After this, photometric factors will be discussed. Lastly, capturing optical flow ground truth and error metrics will be discussed. An explanation about temporal sampling and motion blur can be found in Section 5.1 because it relates to the frequency domain.

2.1 Modeling optical flow

In this section a derivation of optical flow based on the pinhole camera model will be given. Note that here the main quantities of interest are the visual observables which can be derived using the assumption that the rotational rates of the observer are known. This section is limited to the derivation of optical flow for point correspondences. For a derivation of optical flow for planar patches, relating different 3D surface models, 3D motion models, camera models and 2D motion field models, often used in frame interpolation and structure from motion applications, the reader is referred to Konrad, 1999.

2.1.1 The pinhole camera model

Longuet-Higgins and Prazdny, 1980 were the first to formulate optical flow. For this derivation, the authors assumed a pinhole camera model. This means that the hole in which light enters the camera can be modeled as a point and the retina can be seen as a plane. Note that these assumptions are not valid for cameras with a wide-angle lens. Consider point A with coordinates (X, Y, Z) in the observer reference frame $OXYZ$ where O is the aperture of the camera as depicted in Figure 2.1. Note that the image plane (x, y) is one focal length f away from the aperture O. The observer moves through

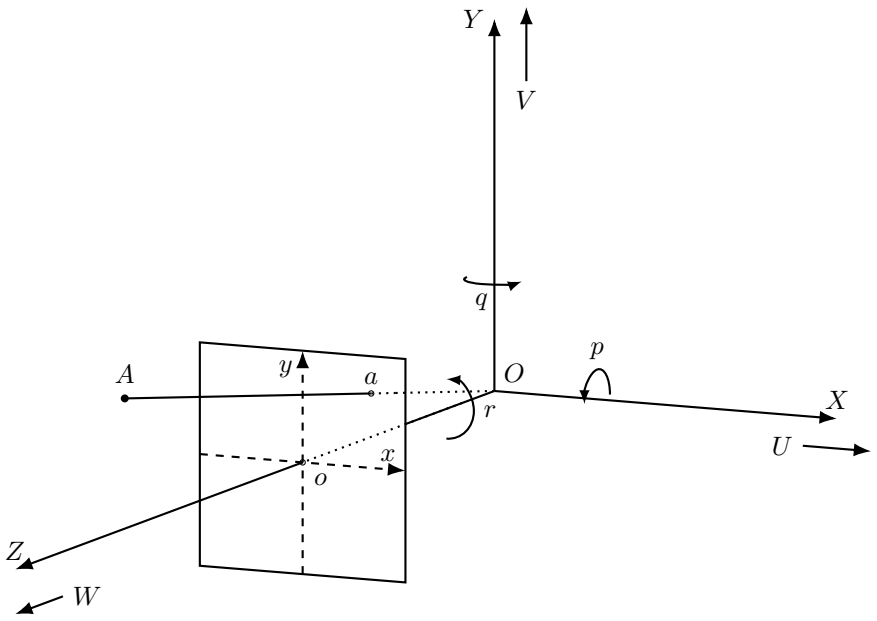


Figure 2.1: The pinhole camera with coordinate system OXYZ. Adapted from Longuet-Higgins and Prazdny, 1980.

the environment with velocities U, V, W and rotational velocities p, q, r. It can the be shown that the velocity component of world point A relative to the moving aperture O are given by:

$$\begin{aligned} \dot{X} &= -U - qZ + rY \\ \dot{Y} &= -V - rX + pZ \\ \dot{Z} &= -W - pY + qX \end{aligned} \tag{2.1}$$

Note the minus signs, as the velocities of A are opposite to the observer in O. The position of the projection of A on the image plane given by point a are related to each other by:

$$(x, y, f)^T = \frac{f}{Z}(X, Y, Z)^T \tag{2.2}$$

Then, for notational convenience $f = 1$ is assumed so that the location of points on the image plane can be written as vectors two dimensions. Then, using the quotient rule, optical flow components u and v can be computed by taking the time derivative of x and y:

$$\begin{aligned} u &= \dot{X}/Z - X\dot{Z}/Z^2 = (-U/Z - q + ry) - x(-W/Z - py + qx) \\ v &= \dot{Y}/Z - Y\dot{Z}/Z^2 = (-V/Z - rx + p) - y(-W/Z - py + qx) \end{aligned} \tag{2.3}$$

Which can also be rewritten in a form that separates the optical flow components in terms of a translation u^T, v^T and rotational u^R, v^R component of the motion of the observer:

$$u = u^T + u^R \qquad v = v^T + v^R \tag{2.4}$$

$$u^T = (-U + xW)/Z \qquad v^T = (-V + yW)/Z \tag{2.5}$$

$$u^R = -q + ry + pxy - qx^2 \qquad v^R = -rx + p + py^2 - qxy \tag{2.6}$$

2.1.2 Visual observables from optical flow

In the previous section, the relation between a point on the image plane and the motion of an observer in a static environment was derived. For this setting, the state of the observer is thus equal for all world points, while the depth Z is unknown and varies per point. Hence, multiple points on the image plane can be combined to estimate the unknown quantities. However, due to the high number of unknowns and high computational complexity often a set of simplifying assumptions is used. These result in sets of parameters related to the motion of the observer and are called visual observables.

Derotation

If the observer has access to its own rotational rates, using gyroscopes for example, the flow can be derotated. Vision-based applications in MAVs often use this assumption (de Croon et al., 2013). Longuet-Higgins and Prazdny, 1980 shows that intersection of the line of motion of the observer with the image plane can be written as:

$$x_0 = U/W \qquad y_0 = V/W \tag{2.7}$$

Assuming the rotation components u^R and v^R to be zero, equation 2.5 can be rewritten using equation 2.7 as follows:

$$u = (x - x_0)\, W/Z \qquad\qquad v = (y - y_0)\, W/Z \tag{2.8}$$

From which follows that:

$$u/v = (y - y_0) \,/\, (x - x_0) \tag{2.9}$$

Thus, it can be seen that at point (x_0, y_0) on the image plane the optical flow will be zero and not dependent on the depth Z of the world point. Points further away from this point of expansion will have an increasing magnitude and therefore the point is referred to as the Focus of Expansion (FoE) when the observer is moving toward the world point or Focus of Contraction (FoC) when it is moving away. Also, the relative depth Z/W of world points can be estimated when the location of the FoE or FoC on the image plane is known. In case the depth Z is known the Time-to-Contact (TTC) is given by $\tau = Z/W$. This gives an estimate how fast the observer approaches the FoE world point.

Planar flow

de Croon et al., 2013 introduces, apart from the static scene assumption, also the assumption of a planar scene in order to simplify the set of equations for optical flow can be simplified even further. Then, h is defined as the distance between the observer's camera pinhole O and the planar surface. Furthermore, the angles α and β are the slopes between the X and Y-axis of the observer. Lastly, the velocity components of the observer are scaled with the depth $u_0 = U/h, v_0 = V/h$ and $w_0 = W/h$, then the the expression for the planar flow field can be defined as:

$$\begin{aligned} u &= -u_0 + (\alpha u_0 + w_0)\, x + \beta u_0 y - \alpha w_0 x^2 - \beta w_0 xy \\ v &= -v_0 + \alpha v_0 x + (\beta v_0 + w_0)\, y - \beta w_0 y^2 - \alpha w_0 xy \end{aligned} \tag{2.10}$$

If the slopes α and β are of negligible magnitude, which is the case when the planar surface is perpendicular to the Z-axis, the equations further simplify to:

$$\begin{aligned} u &= -u_0 + w_0 x \\ v &= -v_0 + w_0 y \end{aligned} \tag{2.11}$$

From these equations it becomes apparent that the depth-scaled velocities of the observer are essential for optical flow estimation of planar flow. These normalized velocities are the primary cues, also known as visual observables, used for navigation. Using these velocities the previously introduced TTC can be estimated as $\tau = Z/W = 1/w_0$. Furthermore, using the mathematical definition of divergence:

$$\nabla \cdot (x, y) = \frac{\partial u}{\partial x}(x, y) + \frac{\partial v}{\partial y}(x, y) \tag{2.12}$$

Combined with equation 2.11, leads to the definition of optical flow divergence D. Which is defined as

$$D = 2w_0 = \frac{2}{\tau} \tag{2.13}$$

2.2 Photometric factors

In this section photometric factors are discussed which influence time-varying image intensity. Some optical flow estimation methods make use of pixel intensity representations which are invariant to different types of changes in lighting over time and are called photometric invariants. For further discussion refer to section 3.3.1.

Photometric factors include the different illumination sources, material reflectance properties and the illumination of a scene to account for highlights and shadows. All these factors are difficult to model accurately. Therefore, instead of providing a complex model for the general case, the dichromatic reflection model (Shafer, 1985) is used, which ignores all but the simplest forms of reflectance and illumination. The model states that color distributions will form a parallelogram in RGB space as can be seen in Figure 2.2. This model relates the observed RGB color $\mathbf{cx}) = (R(\mathbf{x}), G(\mathbf{x}), B(\mathbf{x}))^T$ at a certain location as the sum of an interface reflection component $\mathbf{c}_i(\mathbf{x})$ and a body reflection component $\mathbf{c}_b(\mathbf{x})$:

$$\mathbf{c}(\mathbf{x}) = \mathbf{c}_i(\mathbf{x}) + \mathbf{c}_b(\mathbf{x}) \tag{2.14}$$

Where the interface reflection component is caused by specular (direction-specific) components of reflection. Note that the specular component is dependent upon the surface normal and the viewing direction of the camera. For a perfectly diffuse (Lambertian) surface, however, the body component depends only on the angle between the surface normal and the sources of illumination and not on the viewing direction of the camera. This model does not take into account the characteristics of the camera. Using the assumption that the illumination is spectrally uniform, equation 2.14 can be written in terms of overall intensity of the light source e, geometrical reflection factor $m(\mathbf{x})$ and the reflectance color $\widehat{\boldsymbol{c}}$:

$$\mathbf{c}(\mathbf{x}) = e\left(m_i(\mathbf{x})\widehat{\mathbf{c}}_i(\mathbf{x}) + m_b(\mathbf{x})\widehat{\boldsymbol{c}}_b(\mathbf{x})\right) \tag{2.15}$$

Thus, it is assumed that the interface ($\widehat{\boldsymbol{c}}_i$) and body ($\widehat{\boldsymbol{c}}_b$) reflectance colors combine linearly and the geometrical reflection factors $m(\mathbf{x})$, which depend on material properties, serve as weights. Because of the assumption of spectrally uniform illumination, only achromatic colors ((white to gray to black)

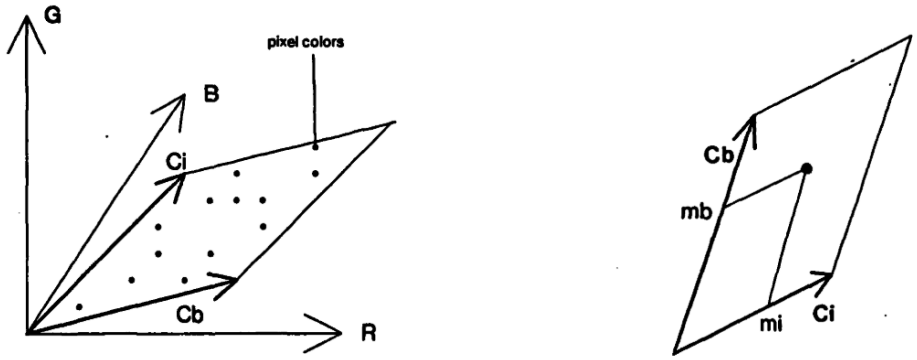


Figure 2.2: (left) Linear combinations of $\hat{\mathbf{c}}_i$ and $\hat{\mathbf{c}}_i$, denoted by Ci and Cb respectively, lie on a parallelogram. (right) The position within the parallelogram determines the magnitude of the geometrical reflection factors $m_i(\mathbf{x})$ and $m_b(\mathbf{x})$, denoted by mb and mi respectively. Both figures taken from Shafer, 1985.

can be modeled. Then all channels of $\widehat{\mathbf{c}}_i$ contribute proportionally; $\widehat{R}_i(\mathbf{x}) = \widehat{G}_i(\mathbf{x}) = \widehat{B}_i(\mathbf{x}) =: w_i(\mathbf{x})$. Lastly, if a neutral interface reflection (Lee, Breneman & Schulte, 1990) is assumed, the value of $w_i(\mathbf{x})$ becomes independent of the spatial position and equation 2.15 can be rewritten as:

$$\mathbf{c}(\mathbf{x}) = e\left(m_i(\mathbf{x})w_i\mathbf{1} + m_b(\mathbf{x})\widehat{\mathbf{c}}_b(\mathbf{x})\right) \tag{2.16}$$

Where $\mathbf{1} = (1,1,1)^T$. Equation 2.16 is the dichromatic model as presented in Shafer, 1985. Based on the dichromatic model three types of photometric invariants (independence of photometric variables) can be distinguished:

1. **Global multiplicative illumination changes:** Concerning expressions of $\mathbf{c}(\mathbf{x})$ independent of the light source intensity e.
2. **Shadow and Shading:** Concerning expressions of $\mathbf{c}(\mathbf{x})$ independent of light source intensity e and body reflection factor m_b, given $m_i = 0$ (which is the case for matte surfaces).
3. **Highlights and specular reflections:** Concerning expressions of $\mathbf{c}(\mathbf{x})$ independent of light source intensity e and body reflection factor m_b and also interface reflection factor m_i.

The main limitations of the dichromatic model are the fact that is does not account for ambient light and causes problems for uncalibrated images (van de Weijer & Beigpour, 2011).

2.3 Optical flow performance evaluation

For other computer vision areas such as stereo, segmentation and object recognition, ground truth data can be captured by using specialized sensors or manual labeling. The creation of ground-truth data for optical flow is difficult because there are no sensors available for optical flow and manual labeling is often difficult and very time-consuming. This section deals with the capturing of ground truth optical flow and the evaluation thereof.

2.3.1 Capturing optical flow ground truth

This section deals with the capturing of optical flow ground truth data for the specific purpose of testing optical flow estimation methods on real-world scenarios and/or providing a benchmark to compare the

performance of different optical flow estimation methods. Note that the motivation for the creation of these datasets is highlighting difficulties of optical flow, which include:

- The aperture problem and regions with no or weak texture. Amplifying the fact that the optical flow estimation problem is ill-posed. Further details will be discussed in Section 3.1.
- Nonrigid motion, motion discontinuities and occlusions.
- Large motion of small scale structure which is a well-know drawback of the coarse-to-fine scheme. Further discussed in Section 3.3.
- Illumination changes, highlights and specular reflections.
- Motion blur, defocus blur and atmospheric effects (such as fog).
- Camera noise

An overview of the datasets used for benchmarking and testing can be seen in 6.2. A subvision between the datasets is made based on the method used for obtaining the ground truth of the image sequences: 'Natural sequences' refer to sequences made with a real camera and 'synthetic sequences' are made digitally.

Natural sequences

The 'Middlebury' (Baker et al., 2011), 'KITTI 2012' (Geiger, Lenz & Urtasun, 2012), 'KITTI 2015' (Menze & Geiger, 2015) and 'HD1K' (Kondermann et al., 2016) contain ground truth data from natural scenes. Baker et al., 2011 use fluorescent paint (which is only visible to the ground truth camera), down-sampling of high-resolution images and sequential tracking of small motion to obtain a dense, sub-pixel accurate ground truth containing non-rigid motion. Geiger et al., 2012 use a car with two high-resolution cameras, a laser scanner, and a high-accuracy localization system. Their dataset is focused on the application of autonomous driving. While KITTI 2012 contains sequences from different geospatial locations, Kondermann et al., 2016 create an autonomous driving dataset which is recorded in only a single street using a similar setup to KITTI 2012. Their main motivation for creating this dataset is that it represents challenges specific to urban autonomous driving. The KITTI 2015 uses a data acquisition method similar to KITTI 2012, however they produce a scene flow dataset which means a 3D representation of the motion field is made. This 3D motion field can be projected onto the 2D plane to obtain ground truth which we refer to as optical flow.

Synthetic sequences

Both the Middlebury and 'MPI-Sintel' (Butler et al., 2012) datasets contain synthetic optical flow ground truth. The main advantage of generating synthetic datasets is that the ground truth accuracy is optimal. The drawback is that the rendered ground truth is only as close to real-world scenarios as the models that describe them. Baker et al., 2011 use the rendering program '3Delight' to generate ground truth without motion blur. Butler et al., 2012 modify the rendering program 'Blender' to obtain dense optical flow ground truth for the open-source action movie 'Sintel'. Their synthetic dataset contains long-sequences, large motions, motion blur, and atmospheric effects. The movie is rendered using different passes, at which different illumination models are active. The 'Albedo pass' is for flat unshaded surfaces, the 'Clean pass' includes shading and specular reflections and the 'Final pass' includes motion blur and atmospheric effects (such as fog).

Dataset	Published in	Synthetic/natural	#Frames for testing	Resolution
Middlebury	Baker et al., 2011	S/N	8	640 x 480
KITTI 2012	Geiger, Lenz and Urtasun, 2012	N	194	1,242 x 375
MPI-Sintel	Butler, Wulff, Stanley and Black, 2012	S	1,064	1,024 x 436
KITTI 2015	Menze and Geiger, 2015	N	200	1,242 x 375
HD1K	Kondermann et al., 2016	N	3,563	2,560 x 1080

Table 2.1: Overview of both synthetic and natural datasets with dense optical flow ground truth. Note that datasets with a private testset can be used a benchmark. The benchmark most often used is MPI-Sintel. Adapted from Mayer et al., 2018.

2.3.2 Flow error metrics

In this section, the flow error metrics commonly used in literature are presented. The Angular Error (AE) was first introduced by D. Fleet and Jepson, 1989 but gained wide adoption thanks to the use of the measure in the work of Barron, Fleet and Beachemin, 1994. The AE flow field vector estimate (u, v) and the ground-truth flow vector (u_{GT}, v_{GT}) is defined as the 3D angle in spatiotemporal space between $(u, v, 1.0)$ and $(u_{GT}, v_{GT}, 1.0)$. It is defined as:

$$\mathrm{AE} = \cos^{-1}\left(\frac{1.0 + u \times u_{\mathrm{GT}} + v \times v_{\mathrm{GT}}}{\sqrt{1.0 + u^2 + v^2}\sqrt{1.0 + u_{\mathrm{GT}}^2 + v_{\mathrm{GT}}^2}}\right) \tag{2.17}$$

The consequence of using this error metric is that errors in flow fields with large magnitude are penalized less than errors in flow fields with a small magnitude. Otte and Nagel, 1994 defined optical flow error in terms of magnitude of the distance vector between the ground truth and flow field vector estimate. This is also known as the Endpoint Error (EE) and is defined as:

$$\mathrm{EE} = \sqrt{(u - u_{GT})^2 + (v - v_{GT})^2} \tag{2.18}$$

3

Differential Methods

In this Chapter differential image intensity methods are discussed. Firstly, the optical flow constraint equation will be derived, followed by an explanation of the aperture problem. Next, the taxonomy of S. S. Beauchemin and Barron, 1995 is used for the distinction between local and global methods. In the section about global methods the taxonomy of Baker et al., 2011 is used, and a condensed overview of global energy methods is given. For an in-depth review of the entire field of differential-based optical flow estimation methods the reader is encouraged to study Barron et al., 1994 and Baker et al., 2011.

3.1 Optical flow constraint

Consider the image intensity function $I(\mathbf{x}, t)$ which provides the brightness of a pixel at image plane location $\mathbf{x} = (x, y)^T$ at time t and let the optical flow be denoted by $\mathbf{v} = (u, v)^T$. B. K. Horn and Schunck, 1981 were the first to introduce the brightness constancy assumption, assuming that under a short period of time the intensity of a pixel, when it flows from one image to another, does not change. This combines a number of assumptions; that the scene is Lambertian, the illumination in the scene is uniform and that there is no vignetting in the camera. The latter means there is no reduction in brightness towards the edges of the camera compared to the center of the image plane. Now consider the simple case of translation:

$$I(\mathbf{x}, t) = I(\mathbf{x} + \mathbf{v}, t + 1) \tag{3.1}$$

This equation can be linearized by applying a first-order Taylor expansion and leads to the *Optical Flow Constraint* equation:

$$\nabla I(\mathbf{x}, t) \cdot \mathbf{v} + I_t(\mathbf{x}, t) = 0 \tag{3.2}$$

Where $I_t(\mathbf{x}, t)$ denotes the temporal image intensity derivative and $\nabla I(\mathbf{x}, t) = (I_x(\mathbf{x}, t), I_y(\mathbf{x}, t))^T$. Note that Equation 3.2 has the two optical flow components $(u, v)^T$ as unkowns. S. S. Beauchemin and Barron, 1995 shows that equation 3.2 leads to the aperture problem (Ullman, 1979). Meaning, only image velocity in the direction of the local image gradient can be determined. Because of the two unknowns, only one component of velocity can be resolved simultaneously. This problem is illustrated in Figure 3.1 for a diagonally translating intensity pattern.

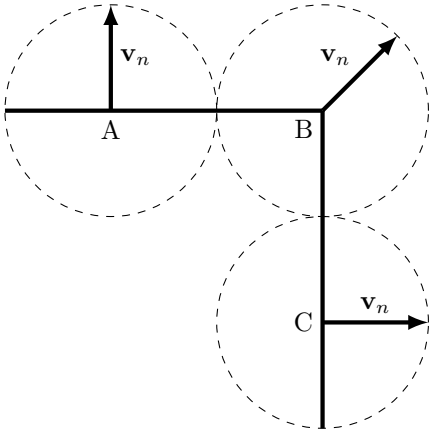


Figure 3.1: Illustration of the aperture problem for a diagonally translating intensity pattern. Note that through the apertures A and B only the velocity component normal to the intensity pattern can be estimated. Inside aperture B both velocity components can be resolved. Adapted from S. S. Beauchemin and Barron, 1995.

Note that the optical flow components $(u,\, v)^T$ can be rewritten in terms of the velocity component perpendicular to the contours of constant intensity, also known as component velocity $\mathbf{v}_n$. Note that $\mathbf{v}_n = s\mathbf{n}$, where s is the normal speed and $\mathbf{n}$ the normal direction. Equation 3.2 can then be rewritten as follows:

$$s(\mathbf{x}, t) = \frac{-I_t(\mathbf{x}, t)}{\|\nabla I(\mathbf{x}, t)\|}, \quad \mathbf{n}(\mathbf{x}, t) = \frac{\nabla I(\mathbf{x}, t)}{\|\nabla I(\mathbf{x}, t)\|} \tag{3.3}$$

In order to solve for Equation 3.2 more constraints are needed which will be discussed in the following sections.

3.2 Local methods

Lucas and Kanade, 1981 introduce additional constraints in the form of a local smoothness assumption. This means it is assumed the optical flow in a local region is constant. Consider a window function $W(\mathbf{x})$, which gives more weight to the centre. Then the optical flow v can be estimated around the small spatial neighborhood Ω using:

$$\sum_{\mathbf{x}\in\Omega} W^2(\mathbf{x}) \Big[\nabla I(\mathbf{x}, t) \cdot \mathbf{v} + I_t(\mathbf{x}, t)\Big]^2 \tag{3.4}$$

Which can be seen as a weighted minimization of the local least-squares solution of the optical flow constraint. Simoncelli, Adelson and Heeger, 1991 extend the approach to incorporate the eigenvalues of the least-squares matrix as a confidence measure. Based on the Harris corner detector (Harris & Stephens, 1988), which makes use of the same least-squares matrix (also know as the second moment matrix), the interpretation of the eigenvalues threshold for eigenvalues is as follows; when both eigenvalues are small, it is considered a flat region. When one or two eigenvalues are large there is an edge, and if both are large a corner is detected. Barron et al., 1994 find that using a threshold for both eigenvalues outperforms using a threshold for the sum of the eigenvalues.

Uras, Girosi, Verri and Torre, 1988 make use of the second-order intensity derivatives to further constrain Equation 3.2:

$$(\nabla\nabla I(\mathbf{x},t))\mathbf{v}^T = -\nabla I_t(\mathbf{x},t) \tag{3.5}$$

Although this equation provides enough constraints for the velocity to be resolved for a single image point, estimates from an 8x8 region are used as input, from which they select the eight estimates which best fit the constraint $\|(\nabla\mathbf{v})^T\nabla I(\mathbf{x},t)\| \ll \|\nabla I_t(\mathbf{x},t)\|$. The drawback of this method is the fact that the constancy assumption on the gradient (second-order derivatives) are invalid for rotation, dilation, and shear. While the brightness constancy assumptions allows for affine deformations. Nagel and Enkelmann, 1986 were the first to show that image points which have high Gaussian curvature, such as corners, can solve the aperture problem. Note that Gaussian curvature is expressed as $det(\nabla\nabla I(\mathbf{x},t))$. It is for this reason Barron et al., 1994 use a threshold on $det(\nabla\nabla I(\mathbf{x},t))$ for the method of Uras et al., 1988 to obtain reliable estimates.

3.3 Global methods

Most global intensity-based differential methods phrase the optical flow estimation problem in terms of a global energy function:

$$E_{\text{ Global }} = E_{\text{ Data }} + \lambda E_{\text{ Prior}} \tag{3.6}$$

Where E_{Data} is the data term and E_{Prior} the prior term and λ is a weight factor between both terms. The data term measures the consistency of the flow with the input images, high energy corresponds to more deviations from this consistency. The prior term favors certain types of flow fields over others. In the following sections, the data term, prior term, and optimization of Equation 3.6 will be discussed. Lastly, sparse-to-dense correspondence matching approaches will be discussed. These methods use sparse feature matches as an initialization for the global energy function.

3.3.1 Data term

As a starting point for the data term either the non-linearized brightness constancy assumption (Equation 3.1) or the optical flow constraint (Equation 3.2) is used. When the non-linearized brightness constancy assumption is used, it is usually converted to the optical flow constraint during optimization. Which will be further discussed in Section 3.3.3. Note that both of these equations provide an error measure per pixel. B. K. Horn and Schunck, 1981 used a quadratic error function (L2 norm) to aggregate the error. This leads to the following data term:

$$E_{\text{Data}} = \sum_{\mathbf{x}} \Big[\nabla I(\mathbf{x},t)\cdot\mathbf{v} + I_t(\mathbf{x},t)\Big]^2 \tag{3.7}$$

Brox et al., 2004 note that using a quadratic penalty function gives too much weight to outliers in the estimation. Therefore the L1 norm with a small positive constant ϵ is used which benefits the optimization because it ensures the resulting function is still convex. This results in the following data term:

$$E_{\text{Data}} = \sum_{\mathbf{x}} \sqrt{\|\nabla I(\mathbf{x},t)\cdot\mathbf{v} + I_t(\mathbf{x},t)\|_1^2 + \epsilon^2} \tag{3.8}$$

Where $\|\cdot\|_1$ denotes the L1 norm of the errors.

Concept		Photometric invariance	#Ch	AAE orig.	AEE mult.	AEE mult. + add.
Standard	RGB	None	3	2.65 °	43.44 °	43.44 °
Color space	HSL (Hue)	Shading, highlights and specularities	1	4.28 °	4.28 °	4.28 °
	spherical (ϕ, θ)	Shading	2	**2.07** °	**2.07** °	**3.37** °
Normalization	RGB (arithm. mean)	Shading	3	2.22 °	2.22 °	3.71 °
	RGB (geom. mean)	Shading	3	2.26 °	2.26 °	5.64 °
Log-derivatives	∇ln(RGB)	Shading	6	2.89 °	3.04 °	4.35 °
Brox et al. (2D)	RGB + ∇RGB	Highlights and specularities	9	2.64 °	3.89 °	3.92 °

Table 3.1: Performance of the constancy assumption for different concepts with intensity channels containing photometrically invariants under original, multiplicative and additive lighting in AAE on the street sequence (`http://of-eval.sourceforge.net`). Table is adapted from Mileva, Bruhn and Weickert, 2007.

Photometrically invariant features and color

The biggest drawback of the brightness constancy assumption is that it is not able to cope with illumination changes over time. Therefore, a gradient constancy is used by Brox et al., 2004, which makes the data term more robust to additive illumination changes. When taking the gradient of the brightness constancy assumption:

$$\nabla I(\mathbf{x}, t) = \nabla I(\mathbf{x} + \mathbf{v}, t + 1) \tag{3.9}$$

The gradient constancy assumption assumes the flow to be locally translational. Meaning that the gradient constancy assumption can be violated by local scale changes, while the brightness constancy assumption still holds. Note that when equation 3.9 is linearized using a first-order Taylor expansion the result will be the same as equation 3.5.

Instead of using the gradient constancy assumption, Mileva, Bruhn and Weickert, 2007 consider different concepts with photometrically invariant intensity channels. The performance of the different concepts on a synthetic sequence is measured for both multiplicative and additive lighting as can be seen in Table 3.1. From this, it can be seen that the brightness constancy assumption on RGB intensity channels fails when multiplicative lighting is added to the test sequence. In the HSV color space, the Hue channel is photometrically invariant to both shadow and shading as well as highlights and specularities. Therefore, the HSV color space is able to achieve the same level of performance for both multiplicative and additive lighting (section 2.2). However, because the transform of RGB to HSV color space involves the ratio of color channel differences, it also discards information. The channels ϕ and θ of the spherical color space (r, ϕ, θ) are invariant to shadow and shading and provide the best results. Also, note the robustness of the method of Brox et al., 2004 to additive illumination changes.

Zimmer et al., 2011 propose the use of the HSV color space for the data term. By allowing a separate robustification of each channel, the most suitable channel for each spatial location can be chosen. As can be seen in Figure 3.2, different channels in the HSV color space[1] posses various degrees of photometric invariance. In Figure 3.3 an example of the different weights given to different HSV channels for different spatial locations are given. Here it can be seen, different channels receive larger or smaller weights in shadow regions. Especially, note the black regions which correspond to no weight in the value channel (bottom right) which is sensitive to shading in Figure 3.3.

3.3.2 Prior term

Because the data term has more unknowns than constraints, the problem is ill-posed. Therefore, more constraints are needed. Most prior terms favor smoothly-varying flow fields. The simplest prior is by

[1]The hue channel is expressed as an angle in between 0° and 360° and in order to make the hue channel differentiable the cosine and sine of the angle are used as input.

Figure 3.2: HSV decomposition of the *Rubberwhale* image from the Middlebury optical flow dataset (Baker et al., 2011). (top left) RGB image with a zoom in on the shadow of the wheel. *Top right:* Hue channel with maximum values in the saturation and value channels. (bottom from left to right) The saturation and value channel respectively. Note that in the hue and saturation channel the shadow is not visible. Taken from Zimmer, Bruhn and Weickert, 2011.

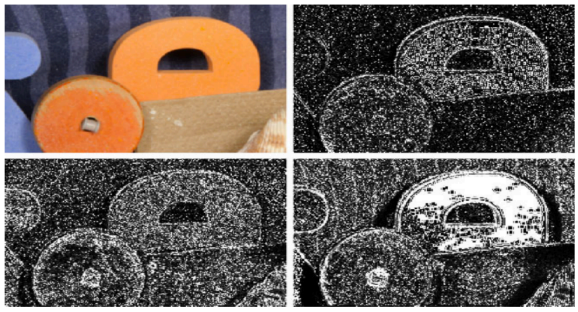


Figure 3.3: The weights visualized for different channels in the HSV color space. (top left) Zoom in of the *Rubberwhale* image of the Middlebury optical flow dataset(Baker et al., 2011). (top right) Weights applied to the hue channel. Note that a larger weight corresponds to brighter pixels. (bottom from left to right) Weights for the saturation and value channel respectively. Note that the value channel is not invariant to shading and therefore receives almost no weight in the shaded area. Taken from Zimmer, Bruhn and Weickert, 2011.

using an L2 norm for the optical flow gradients, first used by B. K. Horn and Schunck, 1981. This can be written as:

$$E_{\text{Prior}} = \sum_{\mathbf{x}} \left[\|\nabla u\|_2^2 + \|\nabla v\|_2^2 \right] \tag{3.10}$$

Where $\|\cdot\|_2$ denotes the L2 norm, and $\nabla = (\partial_{\mathbf{x}})$ is the spatial flow gradient. The combination of the L2 norm for the data (Equation 3.7) and prior (Equation 3.10) leads to the energy formulation of B. K. Horn and Schunck, 1981. Note that if more than two frames are used, a spatiotemporal gradient, defined as $\nabla = (\partial_{\mathbf{x}}, \partial_t)$, can also be incorporated. Brox et al., 2004 use an L1 norm with a small positive constant ϵ similar to Equation 3.8 where the absolute values of the gradients are first added, after which the penalty function is used. A spatial weighting function as a function of the gradient $w(\nabla I)$ to reduce the influence of the prior term near edges can also be used:

$$E_{\text{Prior}} = \sum_{\mathbf{x}} w(\nabla I) \left[\|\nabla u\|_2^2 + \|\nabla v\|_2^2 \right] \tag{3.11}$$

The use of such a function is based on the assumption edges have a high gradient, and that motion boundaries often coincide with edges. This term can also be used in combination with a segmentation algorithm to vary the weight between different segments (Seitz & Baker, 2009). Note that in equation 3.11 the weighting function treats all directions equally. Nagel and Enkelmann, 1986 use an anisotropic weighting function which penalizes the direction along the image gradient more than the direction perpendicular to it.

3.3.3 Optimization

One approach used in the minimization of the global energy function is called gradient descent. Let $\mathbf{f}$ denote the vector which results from the concatenation of all optical flow components for every pixel. Baker and Matthews, 2004 use the simplest form of gradient descent called steepest descent, which makes steps in the direction of the gradient $-\frac{\partial E_{\text{Global}}}{\partial \mathbf{f}}$. Also, more advanced approaches have been used which update the step size for every iteration based on the second derivatives of the global energy

function with respect to **f** (M. J. Black & Anandan, 1996). For a non-linear global energy function, there is no guarantee it will converge to the global minimum, however.

The approaches of Brox et al., 2004; B. K. Horn and Schunck, 1981; Zimmer et al., 2011 allow the formulation of the global energy function to be written as:

$$E_{\text{Global}} = \int E\left(u(\mathbf{x}), v(\mathbf{x}), \mathbf{x}, u_{\mathbf{x}}, v_{\mathbf{x}}\right) \mathrm{d}\mathbf{x} \tag{3.12}$$

Where $(u_{\mathbf{x}}, v_{\mathbf{x}})$ denote the spatial derivative of the optical flow components and the optical flow components are treated as unknown 2D functions rather than unknown values. Then the Euler-Lagrange equations (Agrawal, 2002) can be used to find the minima of the differentiable global energy Equation 3.12. Note that Euler-Lagrange equations belongs to the mathematical analysis field called 'calculus of variations'. This field studies the use of small variations in functions to find local minima. For this reason optical flow methods that use this optimization scheme are called *variational* methods. Note that for the method of B. K. Horn and Schunck, 1981 the Euler-Lagrange equations are linear and the linear system of equations can be solved using standard methods like Gauss-Seidel or Successive Over-Relaxation. Brox et al., 2004 numerically approximate their non-linear model to derive a linear set of equations which can be solved using the same methods. For non-linear Euler-Lagrange equations, it can be solved using an iterative method comparable to gradient descent.

Many approaches use a coarse-to-fine strategy to deal with large motions and significantly reduce computation time (M. J. Black & Anandan, 1996; Bruhn, Weickert & Schnörr, 2005) in the form of an image pyramid. From bottom to top, the image is down-sampled, resulting in reduced resolution. The optical flow is then first estimated for the top part of the pyramid, corresponding to the coarsest resolution. Then, the flow is up-sampled and used as an initialization for the next level. Because the top-level requires fewer parameters to be estimated and is used as initialization for the next level, the amount of computation time is significantly reduced. The main limitation of the coarse-to-fine approach is that it tends to produce erroneous flow on small scale fast-moving objects, which will be discussed in the following Section.

3.3.4 Matching-based extensions

As mentioned, the drawback of the coarse-to-fine warping scheme is that as soon as the motion of small-scale structures is larger than its scale, the motion estimation is often incorrect. Human motion estimation often suffers from this problem as small limbs can move very fast, as can be seen in Figure 3.4. Brox and Malik, 2011 propose to solve this problem by adding a descriptor matching term to their global energy function(Brox et al., 2004). This matching term makes use of Histogram of Oriented Gradients (HOG) descriptors (Dalal, Histograms & Triggs, 2005) to produce sparse feature correspondences. The drawback of using HOG descriptors is that they implicitly assume rigid motions.

Weinzaepfel, Revaud, Harchaoui and Schmid, 2013 use convolutions from the target image with patches from the reference image to produce dense matches as can be seen in Figure 3.5. The response maps of these convolutions are then 'aggregated' to obtain response maps equivalent to convolutions of the target image with larger patches of the reference image at different scales. For details about the 'aggregation' process the reader is encouraged to read Weinzaepfel et al., 2013. Figure 3.6 shows that the larger patches of the reference image provide more distinct activations in the response maps. The matching architecture works in a bottom-up fashion, the convolutions of the target image with the smallest patches of the reference image are considered first. As the algorithm moves on to coarser response maps, the matching problem gets easier, and larger patch matches receive a larger weight. They show that their approach called 'DeepFlow' is able to cope with non-rigid deformations, such as scale changes and rotations.

Revaud, Weinzaepfel, Harchaoui and Schmid, 2015 tackle the problem of large displacements with

Figure 3.4: Illustration of large displacement of small structures. (left) The two input images overlayed. (middle) Flow field produced by Brox, Bruhn, Papenberg and Weickert, 2004. The flow field color coding can be found in Appendix B with the exception that black instead of white corresponds to zero flow. (right) Flow field produced by Brox and Malik, 2011. Note the improved optical flow estimation for the hands and balls. Taken from Brox and Malik, 2011.

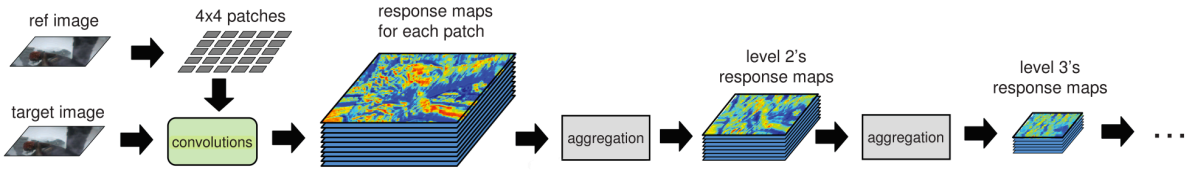


Figure 3.5: The architecture of DeepFlow. The target image is convolved with 4x4 patches of the reference image. The response maps are then aggregated to obtain the response maps of convolutions with the reference image at different scales. Adapted from Weinzaepfel, Revaud, Harchaoui and Schmid, 2013.

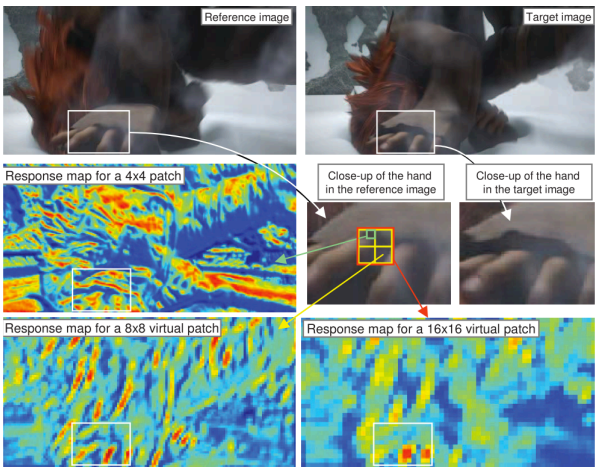


Figure 3.6: Visualization of the response map at different scales used in DeepFlow. This illustrates that the larger scale patches elicit more distinct responses and therefore receive a larger weight. Taken from Weinzaepfel, Revaud, Harchaoui and Schmid, 2013.

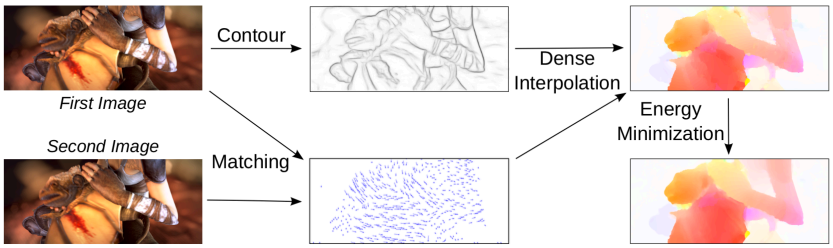


Figure 3.7: Explanation of 'EpicFlow.' The matches generated using the matching part of the *DeepFlow* model are interpolated using the image contours obtained by an edge detector. The contours and matches are used as an input into the OFH model. Taken from Revaud, Weinzaepfel, Harchaoui and Schmid, 2015.

significant occlusions and build upon the assumption that motion discontinuities often coincide with contours. Therefore, they interpolate matches obtained by the matching part of the DeepFlow architecture, using the contours of the reference image as depicted in Figure 3.7 as an additional input.

4

Correlation-based Methods

This Chapter discusses correlation-based methods. S. S. Beauchemin and Barron, 1995 conclude that spatiotemporal sampling rates for the computation of spatiotemporal derivatives are often underestimated in their importance, and too often the assumption of aliasing free imaging (see Chapter 5) is made. At their time of writing the authors conclude that conventional cameras often produce imagery which contains severe aliasing. Increasing the spatiotemporal sampling or prefiltering the images often helps. However, when the number of image frames is small, or the image motion is large, accurate and reliable spatiotemporal derivatives are not always obtainable. In such a case, correlation-based matching approaches are a natural choice. Note that correlation-based parametric *apparent* motion estimation of image regions is often used in video-compression algorithms such as MPEG (Konrad, 1999).

Correlation-based methods try to maximize the similarity between two different intensity regions between different frames. Finding the best match is then defined as finding the shift[1] $\mathbf{d} = (d_x, d_y)$ which maximizes a similarity score or minimizes a distance measure, such as the Sum-of-Squared Differences (SSD):

$$SSD(\mathbf{x}, \mathbf{d}) = \sum_{j=-n}^{n} \sum_{i=-n}^{n} W(i,j)(I(\mathbf{x} + (i,j), t) - I(\mathbf{x} + (i,j), t+1))^2 \tag{4.1}$$

Where $W(i,j)$ denotes a 2D window function over search space Ω. Note that equation 4.1 can be seen as a window-weighted average of a first-order approximation to the temporal derivative $I_t(\mathbf{x}, t)$ (Barron et al., 1994).

Anandan, 1989 implements a Laplacian pyramid and a coarse-to-fine matching strategy using the SSD as a distance measure. Using the coarse-to-fine strategy, which first estimates small motions and later small motions, makes this method more computationally tractable. Also, the Laplacian pyramid helps to enhance image structure such as edges which are useful for matching. The method of Singh, 1991 is very similar to the one of Anandan, 1989 because it also minimizes the SSD distance measure. It consists of a two-stage computation. In the first stage, the SSDs of three time-adjacent images are computed. The motivation for using two SSD surfaces is that it avoids getting trapped in local minima due to noise or periodic texture. Both methods suffer severely from temporal aliasing (see Chapter 5) as becomes apparent on the *Sinusoid1* synthetic sequence (Barron et al., 1994). On top of this, the performance on sub-pixel motions for correlation-based matching techniques is also poor.

Another drawback of correlation-based methods is that the search space Ω scales with $\mathcal{O}(n^2)$ as n increases. Camus, 1997 describes a way to make the computational complexity of the search space for

[1] which is an approximation to velocity

correlation-based methods scale with $\mathcal{O}(n)$ by using a spatiotemporal search range. He achieves this by formulating a spatiotemporal search space definition which grows linear in time.

5

Frequency-based Methods

In this Chapter, frequency-based methods will be discussed. Firstly, the representation of spatiotemporal image structure in the frequency domain is explained. Next, the trade-off between frequency resolution and localization in the space-time will be discussed. Finally, amplitude-based (Heeger, 1987) and phase-based (D. Fleet & Jepson, 1989; Gautama & Van Hulle, 2002) methods for optical flow estimation will be discussed.

5.1 Image velocity in the frequency domain

Consider a moving sine wave. The spatial frequency f_x of this sine wave is expressed in the number of cycles per pixel. The latter of which is pixels in our case. The temporal frequency (f_t) is expressed in the number of cycles per frame. Using this relation velocity (which is expressed in pixels per frame) is found to be:

$$v = f_t / f_x \tag{5.1}$$

Now consider a signal composed of two sinuses y_1 and y_2 with each a different spatial frequency f_x. The signal moves with a velocity v, which means that each sine given their spatial frequency has a temporal frequency of $f_t = v f_x$. This means that y_2, which has a spatial frequency component twice as high as wave y_1, also has a temporal frequency which is twice as high as y_2. This corresponds mathematically to $f_{t_1} = f_{x_1} v = 2 f_{t_2} = 2 f_{x_2} v$ and this is illustrated in Figure 5.1. Now consider a wave moving at velocity v that has many spatiotemporal frequencies. Note that all these frequencies will lie on a line passing through the origin as illustrated in Figure 5.2.

Equation 5.1 can be extended to 2D velocity where the spatial and temporal frequencies can be related to each other using the following equation:

$$f_t = u f_x + v f_y \tag{5.2}$$

Where $\mathbf{v} = (u, v)$ is the 2D velocity of a translating pattern. Now consider a texture translating with a constant velocity in the 2D domain. It is expected that all frequency components of this translating texture will lie on a plane in the 3D spatiotemporal frequency space analogously to the line in the 2D

case. Note that the slope of the plane corresponds to the magnitude of the 2D velocity vector $\mathbf{v}$ and the direction of $\mathbf{v}$ corresponds to the orientation of the plane around the f_t axis.

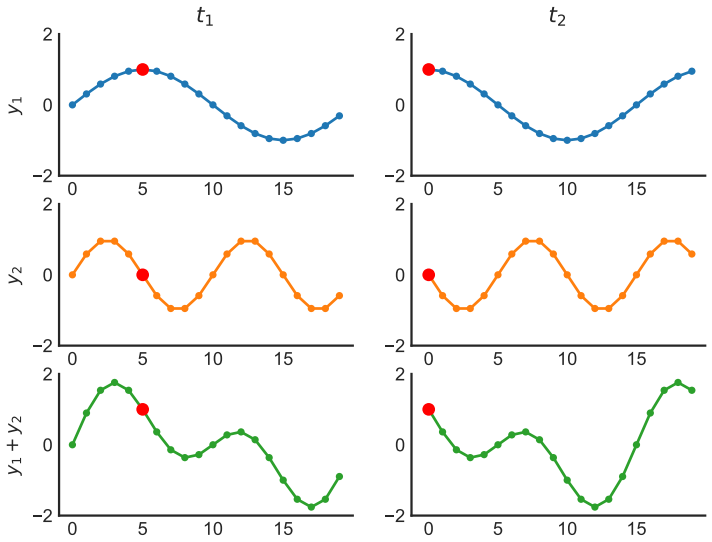


Figure 5.1: Line y_1 and y_2 denote a sine-wave with $f_{x_1} = \frac{1}{20}$ and $f_{x_2} = \frac{1}{10}$ cycles per pixel respectively. Note that the velocity is $v = 5$ pixels per frame. This corresponds to a temporal frequency of $f_{t_1} = \frac{5}{20}$ and $f_{t_2} = \frac{5}{10}$ cycles per frame. This means y_1 is displaced a quarter of its wavelength and y_2 half its wavelength while they are both moving at the same velocity.

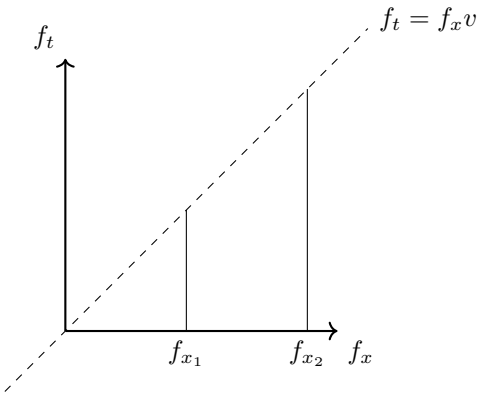


Figure 5.2: All spatiotemporal frequency components for a wave moving at a constant velocity v will lie on a line passing through the origin with slope $\arctan \frac{f_t}{f_x}$.

Temporal sampling and motion blur

Consider a signal $x(t)$ which is band-limited. Meaning, it has power up to and including a certain $f_{t_{\max}}$. Then, according to Shannon's sampling theorem (Shannon, 1948), it should be sampled with a sampling frequency larger than $2f_{t_{\max}}$. Therefore, the requirement for frequency of the sampling function f_s in order to avoid aliasing, also known as the Nyquist frequency, corresponds to $f_s/2 > f_{t_{\max}}$.

If a signal is not sufficiently band-limited before it is sampled, aliasing can occur. Figure 5.3 illustrates the 'window of visibility' (D. Fleet & Jepson, 1989) of a spatiotemporal signal in the frequency domain and the location of the aliased power. Note that the temporal sampling will remove high spatial

frequencies moving at fast speeds. Motion blur in the spatial and frequency domain is illustrated in Figure 5.4.

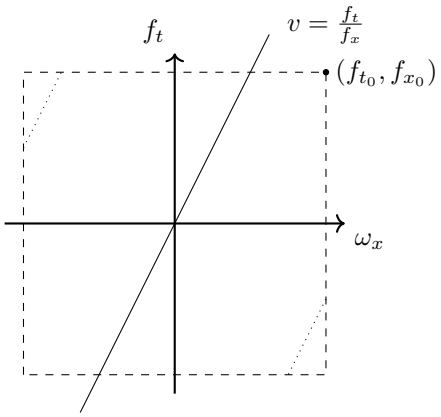


Figure 5.3: After sampling a spatiotemporal signal translating with velocity v the highest spatial and temporal frequencies are denoted by (f_{t_0}, f_{x_0}). The dotted lines near the corners of the spatiotemporal window refers to frequency components with a significant magnitude which occurs due to aliasing when the signal is not properly band-limited prior to sampling. Adapted from D. Fleet and Jepson, 1989.

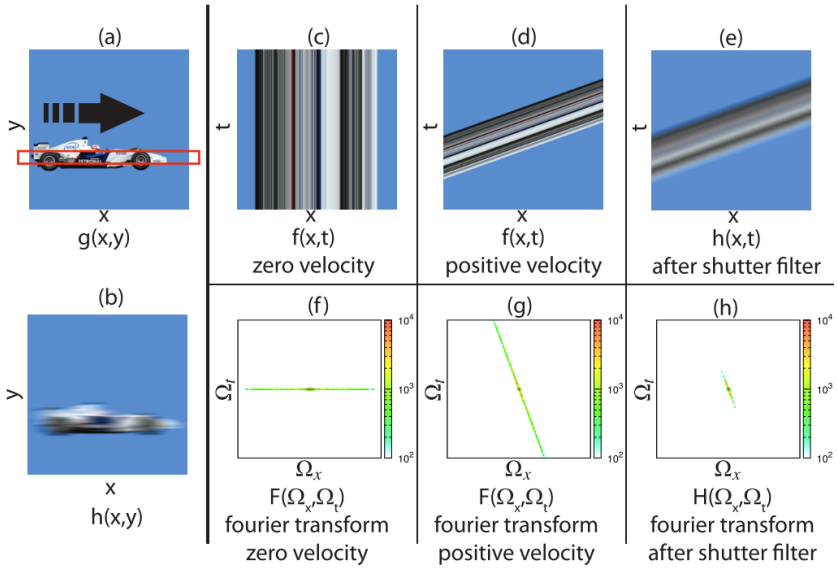


Figure 5.4: (a) Original signal $g(x, y)$. (c) A spatiotemporal representation of the original signal moving with zero velocity along the red scanline. (d)The spatiotemporal representation of the original signal moving with a positive velocity. (e) The same representation after applying a vertical blur to the time axis which corresponds to a shutter filter. (b) Motion blurred version of $g(x, y)$ after applying the vertical motion blur. (f,g,h) Fourier transform of (c), (d) and (e) respectively. Note that (h) has frequencies limited to $\Omega_t \in [-\Omega_t^{\max}, \Omega_t^{\max}]$ corresponding to shutter filter. Taken from Egan, Tseng, Holzschuch, Durand and Ramamoorthi, 2009.

The aperture problem in the frequency domain

Note that the aperture problem in the frequency domain corresponds to the fact that there are two degrees of freedom in equation 5.2 and only one of two velocity components can be extracted at a time. This means that a single line like the one illustrated in Figure 5.2 corresponds to the many planes possible that contain the line of the single component velocity estimate.

5.2 The uncertainty relation

Consider a smooth Gaussian window function $g(t)$ and its Fourier transform depicted in Figure 5.5. Note that the narrow Gaussian window in the time domain becomes 'broad' in the frequency domain (Mulder, van der Vaart, van Staveren, Chu & Mulder, 2016) and vice versa. Note that as the width of the Gaussian window in time decreases the Gaussian window in frequency space will increase and thus the ability to resolve different frequencies will be reduced.

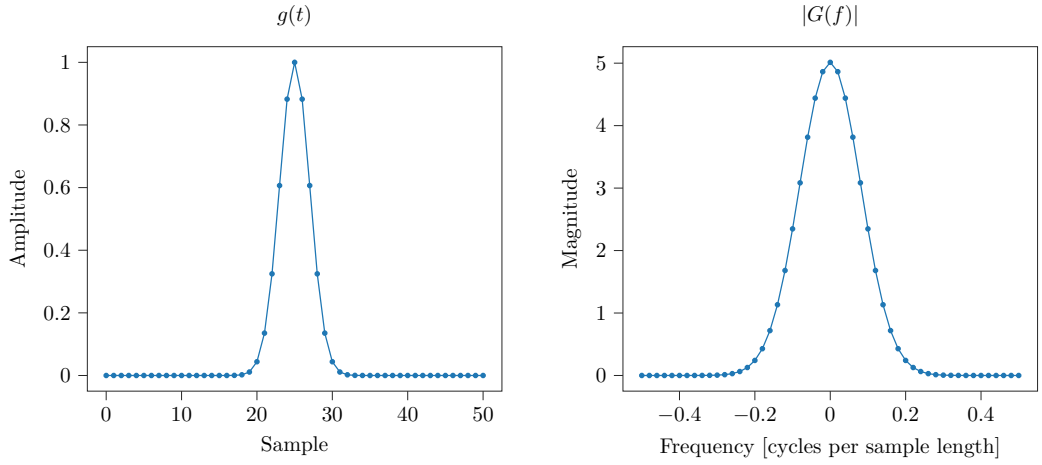


Figure 5.5: A Gaussian window function $g(t)$ and the magnitude of it's Fourier transform $|G(f)|$ for $\sigma = 2$.

This inability to achieve both resolution in time and the frequency domain is referred to as the uncertainty relation (Bracewell, 1986; Gabor, 1945). Gabor formulated a theoretical minimum on the product of the temporal width of a signal and the width of the signal's power spectrum. Different definitions of width (or 'extent') in the context of this formulation exist. The most common definition is given in terms of the variance of the signal in both the temporal and frequency domain. Note that for a 1D time signal $f(t)$ the uncertainty principle is as follows:

$$\sigma_t \sigma_\omega \geq \frac{1}{2}, \quad \sigma_t \sigma_f \geq \frac{1}{4\pi} \approx 0.08 \text{ cycles} \tag{5.3}$$

This means that there is a lower bound on what can be achieved in terms of resolution in time and the frequency domain. It can be shown that a Gaussian window actually attains this lower bound. Consider a Gaussian window function $g(t)$ and it's Fourier transform $G(f)$:

$$g(t) = e^{-\pi\sigma^2 t^2}, \quad G(f) = \frac{1}{\sigma} e^{\frac{\pi f^2}{\sigma^2}} \tag{5.4}$$

Then the variances and lower bound of the uncertainty relation are given by:

$$\sigma_t^2 = \frac{\int t^2 |f(t)|^2 dt}{\int |f(t)|^2 dt} = \frac{1}{4\pi\sigma^2}, \quad \sigma_f^2 = \frac{\int f^2 |G(f)|^2 df}{\int |G(f)|^2 df} = \frac{\sigma^2}{4\pi} \tag{5.5}$$

$$\sigma_t \sigma_f = \frac{1}{\sigma\sqrt{4\pi}} \frac{\sigma}{\sqrt{4\pi}} = \frac{1}{4\pi} \tag{5.6}$$

An extension of the proof above for multi-dimensional signals is given in D. Fleet and Jepson, 1989 Appendix B.

5.3 Gabor filters

In order to extract motion from images it desirable to be able to extract frequency ranges and to localize this response in the space-time domain. These naturally leads to a Gabor filter (Gabor, 1945) because these filters operate on the fundamental limit of the uncertainty relation and are thus optimally localized in both the time and frequency domain. An odd-phase Gabor filter is defined as a Gaussian window multiplied with a sine:

$$g(t)h(t) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left\{\frac{-t^2}{2\sigma^2}\right\} \sin(2\pi f_{t_0} t) \tag{5.7}$$

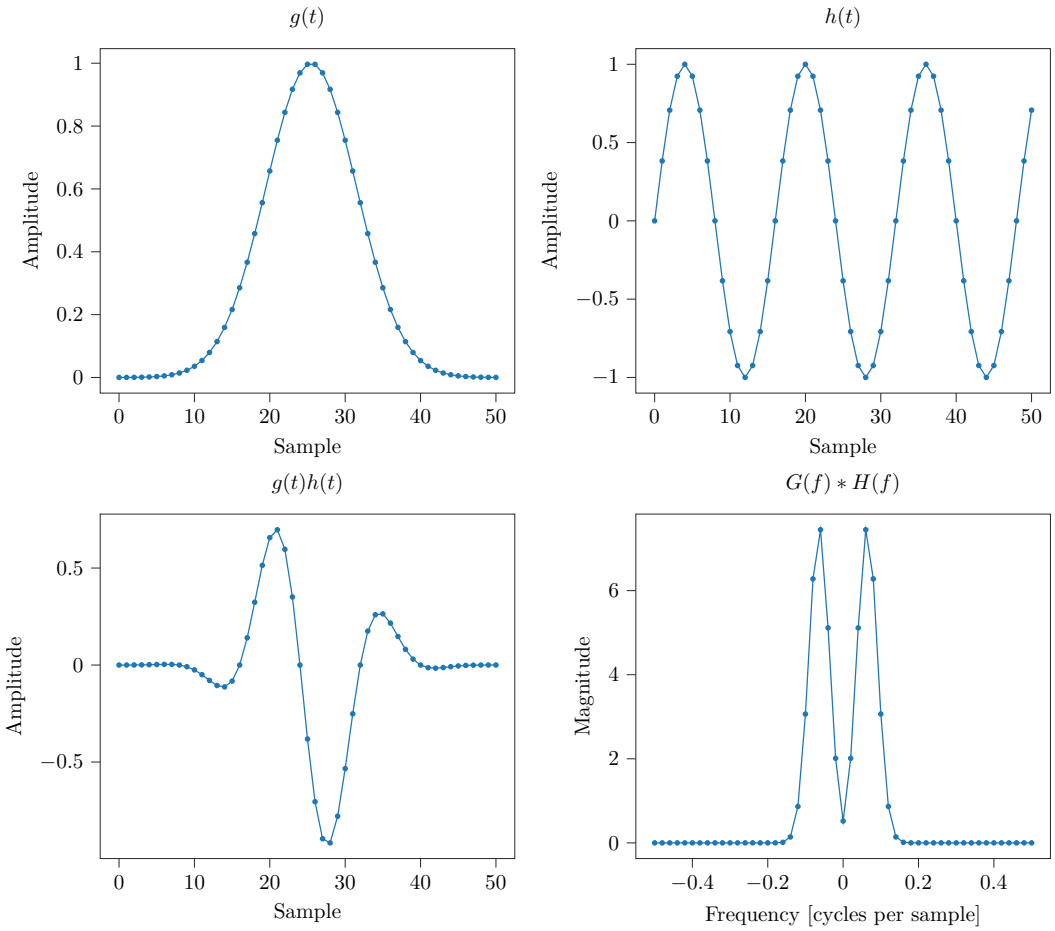


Figure 5.6: (top left) Gaussian window $g(t)$ with $\sigma = 6$. (top right) Sine with frequency$f_t = \frac{1}{16}$ [cycles/sample]. (bottom right) Multiplication of a Gaussian window $g(t)$ and sine wave $h(t)$ leads to the Gabor filter $g(t)h(t)$. The Fourier transform of the Gabor filter corresponds to a convolution of the two Fourier transformed signals in the frequency domain $G(f) * H(f)$. The resulting signal in the frequency domain can is characterized by the 2 Gaussians centered around $+f_t$ and $-f_t$.

The Gabor kernel can also be generalized to 3D space-time signals Heeger, 1988 and is given by the following equation:

$$
\begin{aligned}
g(x,y,t) = & \frac{1}{\sqrt{2}\pi^{3/2}\sigma_x\sigma_y\sigma_t} \\
& \times \exp\left\{-\left(\frac{x^2}{2\sigma_x^2}+\frac{y^2}{2\sigma_y^2}+\frac{t^2}{2\sigma_t^2}\right)\right\} \\
& \times \sin\left(2\pi f_{x_0}x + 2\pi f_{y_0}y + 2\pi f_{t_0}t\right)
\end{aligned} \tag{5.8}
$$

Where $(f_{x_0}, f_{y_0}, f_{t_0})$ are the spatiotemporal center frequencies to which this filter is tuned to and $(\sigma_x, \sigma_y, \sigma_t)$ are the standard deviations which control the spread of the 3D Gaussian window. Then to change the tuning of the filter, the spatiotemporal center frequencies can be adjusted separately. Also, decreasing the width of the Gaussian window in the space-time domain broadens the Gaussian window in the frequency domain, thereby trading frequency resolution for localization in the space-time domain.

Gabor filters are often used in quadrature[1]. Thus, the response of the linear convolution between a Gabor filter pair and an image sequence is complex-valued:

$$
S(\mathbf{x},t) = \rho(\mathbf{x},t)e^{i\phi(\mathbf{x},t)} = I(\mathbf{x},t) * \text{Gabor}\,(\mathbf{x},t) \tag{5.9}
$$

By taking the $\rho(\mathbf{x},t)$ of the quadrature filter pair, a phase invariant response is achieved. Meaning the response of the magnitude of the Gabor kernel is invariant to local phase. By taking the argument of the real and the complex part of the Gabor kernel, the local phase $\phi(\mathbf{x},t)$ can be obtained. The former is the signal on which the Energy-based method of Heeger, 1988 is based. The later is the signal which D. Fleet and Jepson, 1989 exploit for the measurement of image velocity.

5.4 Energy-based methods

Heeger, 1988 uses the magnitude of Gabor kernels to extract motion from image sequences. They use a family of Gabor filters which are all tuned to different spatial orientations but to the same spatial frequency band (meaning $F_0 = \sqrt{f_{x_0}^2 + fy_0{}^2}$). They use a total of twelve filters pairs, where eight are tuned to patterns moving in a certain direction, and the remaining four are tuned to stationary patterns at different orientations. Note that the Gabor filters pairs used are *not* velocity selective but tuned to a temporal frequency f_t. Instead, in order to extract motion, Heeger derives an expression for the expected response of a Gabor filter pair for translating white noise. Remember that the power spectrum of a moving texture is contained in a tilted plane with its center in the origin in the spatiotemporal frequency domain. A different tilt of the plane corresponds to a different velocity magnitude, and a different velocity direction corresponds to a different orientation of the plane about the origin as illustrated in Figure 5.7. The expected energy response of a Gabor filter, tuned to certain center frequencies, is a function of the flow vector $(R\,(u,v))$. Then let the measured motion energy by the Gabor filters be denoted by $m_i (i = 1-12)$ and the corresponding expected motion energy by $R_i\,(u,v)$. Furthermore, let the sum of the measured and predicted motion energy for filters with the same *spatial* orientation be denoted by the following:

$$
\overline{m}_i = \sum_{j \in M_i} m_j \tag{5.10}
$$

[1]Sine and cosine are in quadrature, meaning they are 90 degrees out of phase with each other.

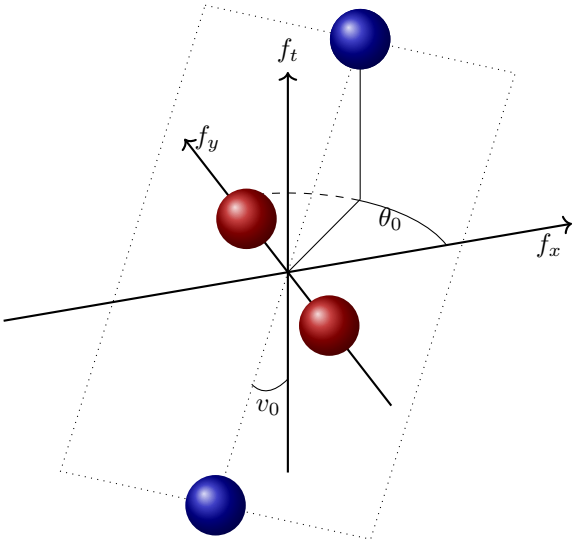


Figure 5.7: The half-magnitude power profile of two Gabor filter pairs in the spatiotemporal frequency domain. The blue filter pair is tuned to translating patterns while the red filter pair reacts to stationary patterns over time. The plane contains all the power related to translation with velocity v_0. The tilt of the plane represents the motion magnitude and the orientation θ_0 represents the orientation of the blue Gabor filter pair. The method of Heeger, 1988 uses 4 filter pairs tuned to stationary patterns (red) and 8 filter pairs tuned to varying orientations and motions (blue). Adapted from Heeger, 1988.

$$\overline{R}_i = \sum_{j \in M_i} R_j(u, v) \tag{5.11}$$

Then a cost function $\mathcal{L}$ can be defined which minimizes the difference between the predicted and measured motion energy:

$$\mathcal{L}(u, v) = \sum_{i=1}^{12} \left[m_i - \overline{m}_i \frac{R_i(u, v)}{\overline{R}_i(u, v)} \right]^2 \tag{5.12}$$

Minimizing this cost function is analogous to the 2D problem of determining the slope of the line passing through the origin while only being able to 'view' it with a certain amount of circular windows. Then the optimization problem corresponds to determining the slope of the line which minimizes the least-squares distance between the line and the center of the circular windows. Where the circular windows correspond to a 2D side-view of the half-magnitude profile of Gabor filters.

Barron et al., 1994 report that the frequency-based methods only produce satisfactory results if the input images contain translation with spatial and temporal frequencies close to the center frequencies of the Gabor filters. The primary source of error for realistic image sequences is the fact that the model only assumes image translation as motion and is unable to deal with deviations from this assumption such as rotation, dilation and occlusion.

5.5 Phase-based methods

In this Section the phase-based methods of D. Fleet and Jepson, 1989 and Gautama and Van Hulle, 2002 will be discussed. The former is a method based on spatiotemporal convolutions, while the latter uses spatial convolutions.

5.5.1 Spatiotemporal filter-based

D. Fleet and Jepson, 1989 define velocity in terms of the phase behavior of velocity-tuned Gabor filters (equation 5.9). They show phase is more stable to noise as well as small deviations from image translations which are typical in 3D scenes. As an example, they show that the constant phase contours of a Gabor kernel provide a better approximation to the motion field of dilating sinusoid as depicted in Figure 5.8.

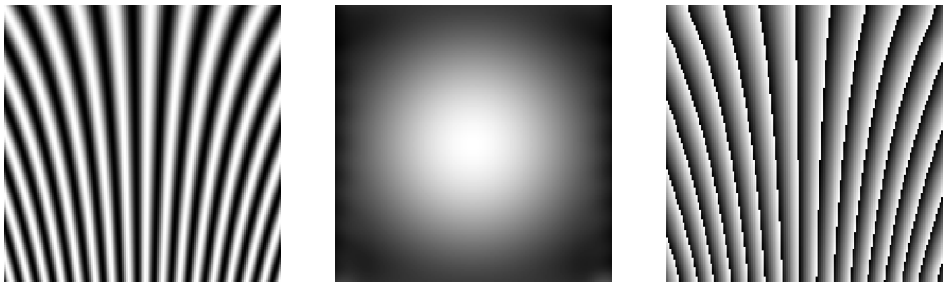


Figure 5.8: (left) A dilating sinusoid given by $I(x,t) = \sin(2\pi x f_{t_0}(1-\alpha t))$. Where $f_{t_0} = \frac{1}{12.5}$ cycles per pixel, $alpha = 0.005$ and the image width and height is 150 pixels. Time is on the vertical axis and the spatial variable x on the horizontal axis. (middle and right) The magnitude and phase output of the image convolved with a Gabor kernel. Note that the Gabor kernel is tuned to a velocity of 0 and the same frequency f_{t_0} as the dilating sinusoid. From these images it can be seen that when there is a small deviation from translation the magnitude of the response quickly vanishes while the constant phase contours still provide a reasonable approximation to the motion field. As the constant phase contours coincide with the lines from the dilating sinusoid along the time axis. Adapted from D. Fleet and Jepson, 1989.

Therefore, solutions to constant response (equation5.9) phase ($\phi(\mathbf{x},t) = c$) are considered. The component of velocity perpendicular to the level phase contours is denoted by $\mathbf{v}_n = s\mathbf{n}$, where the speed and direction are given by:

$$s = \frac{-\phi_t(\mathbf{x},t)}{\|\nabla\phi(\mathbf{x},t)\|}, \quad \mathbf{n} = \frac{\nabla\phi(\mathbf{x},t)}{\|\nabla\phi(\mathbf{x},t)\|} \tag{5.13}$$

From this definition, it can be seen that this method is, in fact, a differential technique applied to phase rather than the pixel intensities (see equation 3.3). They also show that phase information can be unstable, and they impose two constraints on the response of the Gabor kernels. A frequency constraint and an amplitude constraint. The former is a constraint which constrains the accepted frequency to 25% of the nominal tuning range of the filter ($\|(\tilde{f}_{x_0}, \tilde{f}_{y_0}, \tilde{f}_{t_0}) - (f_{x_0}, f_{y_0}, f_{t_0},)\| \leq \sigma_f 1.25$). The latter is a constraint that makes sure the local signal amplitude is as large as the average local amplitude and at a minimum 5% of the largest response across all filters. The component velocity estimates of the different filters are combined into a single 2D velocity estimate using a least-squares technique. Estimates are combined from 5x5 patches, and to this, the best linear 2D velocity model is fitted which minimizes the least-squares error.

Barron et al., 1994 report that the method performs among the best ones they tested. However, due to the high amount of convolutions necessary to obtain the phase information, the computational load is high. They also conclude that the method is sensitive to temporal aliasing, which they claim is due to the frequency tuning of the filters.

5.5.2 Spatial filter-based

Gautama and Van Hulle, 2002 take a spatial filter-based approach as opposed to a spatiotemporal filter-based approach by D. Fleet and Jepson, 1989. They filter an image sequence using a bank of 2D spatial Gabor filters. From this filter output, the temporal phase gradient is computed. If the temporal phase gradient is not sufficiently linear over a specific period, the estimate is deemed unreliable and rejected. The different component velocity estimates are combined to form 2D velocity estimates using a Recurrent Neural Network (RNN). The component velocity estimates of the different filters each impose a constraint on the final state of the flow vector $\mathbf{v} = (u, v)$. The RNN minimizes the distance between the constraints imposed by the component velocities and the flow vector state. They illustrate how this approach can tackle the aperture problem by considering a circle translating with velocity $\mathbf{v} = (1.5, 0.5)$ pixels per frame. The component velocities obtained from the different filters can be seen in part A of Figure 5.9 denoted by the oriented lines. The thick black line in part B of Figure 5.9 denotes the orthogonal distance between the constraints and the flow vector state $\mathbf{v}$. u and v are denoted by the thin solid line and thin dashed line respectively.

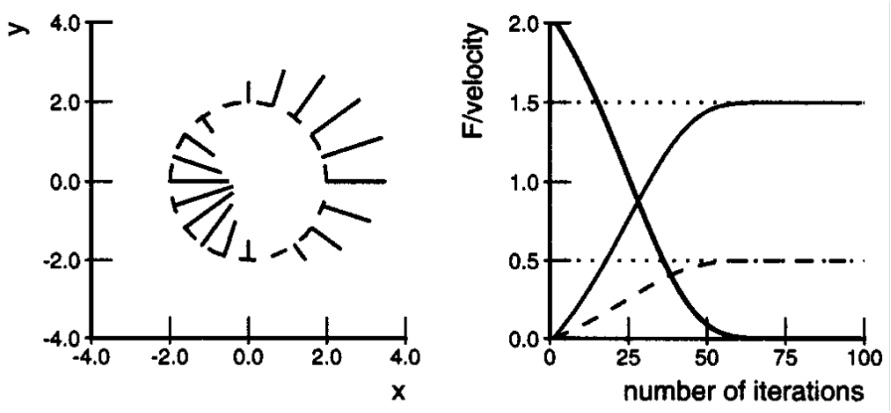


Figure 5.9: (left) Illustration of the aperture problem handling by the architecture of Gautama and Van Hulle, 2002. A circle translating with a velocity of $\mathbf{v} = (1.5, 0.5)$ pixels/frame and the component velocity estimates. (right) Convergence of the RNN to the correct flow vector state. Taken from Gautama and Van Hulle, 2002.

Gautama and Van Hulle, 2002 report that their method is outperformed by far by the method of D. Fleet and Jepson, 1989. They attribute this to the fact that they limit their approach to estimates at a single location, whereas the method of D. Fleet and Jepson, 1989 pools component velocity estimates from a small spatial neighborhood. They mention the possibility of incorporating spatial pooling into their architecture as well. However, by forfeiting the purely local aspect of computations, it would be harder to implement parallel computations and make an efficient implementation. An advantage of their technique is that it allows computation of flow vectors over arbitrary time spans. They conclude that increasing the number of frames for optical flow estimation from two to five frames improves the results initially, but after five frames, the performance remains constant. If a translating object moves beyond a Gabor filter's spatial extent, the phase estimation becomes non-linear and thus rejected. The longer the time span, the higher the chances of this happening. The authors state that the lower flow densities of the methods of D. Fleet and Jepson, 1989 and Lucas and Kanade, 1981 are due to their long time spans.

6

Learning-based Approaches

In this Chapter leaning-based optical flow estimation approaches are discussed. Firstly, the machine-learning based approaches are explained in Section 6.1. Secondly, the basic theory behind CNNs is explained. Thirdly, the CNN-based architectures for optical flow estimation are presented. Lastly, the different aspects of training CNNs for optical flow estimation are introduced.

6.1 Machine-learning-based approaches

M. Black, Yacoob, Jepson and Fleet, 1997; D. J. Fleet, Black, Yacoob and Jepson, 2000 are the first to propose a complex image motion representation as a linear sum of learned orthogonal basis flows as can be seen in Figure 6.1. Because the linear sum of orthogonal basis flows can approximate a large varierty of motions fields. M. Black et al., 1997 extract the learned basis flows from a small synthetic training set using Principal Component Analysis (PCA). Wulff and Black, 2015 take a similar approach. However, in order to compute the orthogonal basis flows, they use the optical flow algorithm 'GPUFlow' (Werlberger et al., 2009) to compute the optical flow for four Hollywood movies. Then, they use robust PCA to extract the orthogonal basis flow fields. Also, they use sparse feature matching as initialization to cope with long-range correspondences and regress these matches to a dense flow field using the orthogonal basis flow fields.

Roth, Black, Roth and Black, 2009 propose a Field-of-Experts model which can be used to learn image priors that reflect the spatial statistics of natural scenes. The Field-of-Experts model can be seen as a shallow CNN. Sun et al., 2008 use this FoE and use a global energy function in which the prior term is replaced by the Field-of-Experts model and the data term consists of a small set of convolutional filters. Because, at the time, there was not sufficient training data and only a small number of filters was used, it did not display the potential of learning-based approaches.

Van Hateren and Ruderman, 1998 has shown that Independent Component Analysis (ICA) of natural image sequences lead to Gabor filters (which are also found in simple cells in the primary visual cortex in the human brain). Thus when applying these filters it is possible to extract motion from image sequences. ICA is different from PCA in that it imposes an higher order independence. PCA only allows for second-order independence.

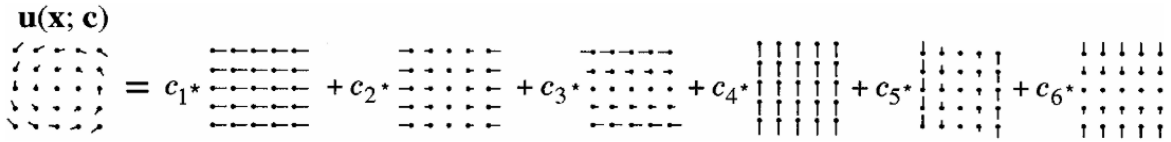


Figure 6.1: A motion field can be represented as a linear sum of orthogonal basis flows. Taken from D. J. Fleet, Black, Yacoob and Jepson, 2000.

6.2 Convolutional Neural Networks

First used by LeCun, 1989, CNNs are designed to process data that come in the form of arrays (LeCun, Bengio & Hinton, 2015). Commonly used input signals are 1D arrays for signals and sequences (such as language), 2D arrays for images, and 3D arrays for video. In order to categorize a neural network as a CNN it needs to contain at least one convolutional layer. Consider Figure 6.2, the solid red squares in the output feature maps correspond to a convolution operation which was performed by the multiplication of a weight matrix called convolutional kernel (LeCun, 1989) with a local path in the input feature map. Note that this multiplication happens channel-wise, and often a bias term is added.

It follows that every single value in a channel in the output feature maps corresponds to a different spatial location in the input feature maps multiplied with the same convolutional kernel. Therefore, the weights are shared across different regions of the image. This makes sense for images; if a pattern is to be extracted from an image, this pattern can be located in any part of the image. Also, local groups of pixel intensities are often highly correlated and form distinctive features (LeCun et al., 2015).

The convolutional operation is often followed by a pooling operation; semantically similar features are spatially merged into one. A max-pooling unit computes the maximum of a local patch in a feature map and often a stride[1] of more than one is used to reduce the dimension of the output feature maps.The pooling operation makes the input invariant to small shifts. CNNs exploit the fact that many signals are composed of several lower-level ones. For images, local patches form edges, edges form patterns, patterns form into parts and parts form into objects. CNNs often have many layers. In the first convolutional layers lower-level features such as lines are extracted and in the deeper layers more complex patterns can be distinguished. After the pooling layer usually an activation function is used. The most popular activation function is called Rectified Linear Unit (ReLU), given by the function $f(z) = max(z, 0)$. The ReLU activation function enables much faster in neural networks with a lot of layers (LeCun et al., 2015).

Ever since the object recognition competition called ImageNet in 2012, the research interest in CNNs has improved dramatically. A CNN designed by Krizhevsky, Sutskever and Hinton, 2012 called 'AlexNet' achieved spectacular results by almost halving the error rates of the best competitors. The main advances of this approach were an efficient GPU implementation which improved training times, and a new regularization technique called dropout (Srivastava, Hinton, Krizhevsky & Salakhutdinov, 2014). AlexNet caused a revolution in computer vision and CNNs have since been the state-of-the-art approach on almost all recognition and detection tasks (Razavian, Azizpour, Sullivan & Carlsson, 2014).

6.3 CNN architectures for optical flow estimation

In this Section different CNNs architectures of optical flow estimation are discussed. Firstly, the encoder-decoder type networks are presented. Which have been the dominant approach in CNN-based methods for optical flow estimation. Note that also the many additions to the original encoder-decoder structure are included. After which an architecture based on signal processing principles is discussed.

[1]The number of spatial shifts in the input feature map between convolutions in the input feature maps

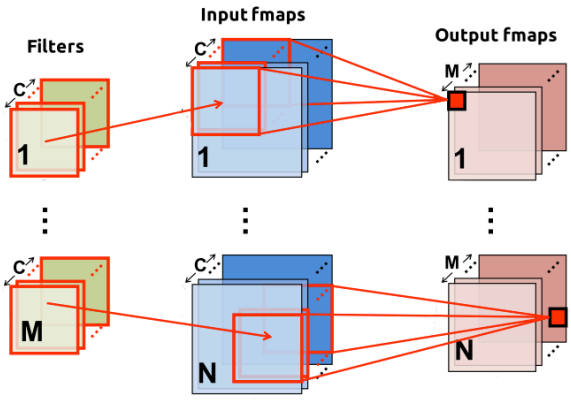


Figure 6.2: The high dimensional convolution operation in CNNs. The input feature maps are N-dimensional corresponding to bath-size N and have C channels. Consider the 2D convolution of a single RGB (N=1, C=3) image with M filters. Note that the channel-wise convolutions of the filters with the input feature maps are summed and often a bias term is added. The amount of channels in the output feature maps is therefore equal to the amount of filters. Adapted from Sze, Chen, Yang and Emer, 2017 .

6.3.1 Encoder-decoder

In Dosovitskiy et al., 2015 the first end-to-end trained CNNs for optical flow estimation were introduced. Based on the U-Net architecture (Ronneberger et al., 2015), two architectures called 'FlowNetSimple' and 'FlowNetCorr' (further referred to as FlowNetS and FlowNetC) are designed which take an image pair as input and produce a corresponding optical flow field as output. Both of these architectures are built on the underlying idea that in the contractive part of the network the spatial image information is compressed and the amount of distinct features increases, while in the expanding part up-convolutions(Eigen, Puhrsch & Fergus, 2014) are used to increase the resolution of the optical flow field. FlowNetS takes two stacked RGB images in a six-channel input. In the first three layers of FlowNetC two separate image processing streams are used which contain the same weights (also known as a Siamese network). After these three layers, a *correlation-layer* is used which computes the similarity score between the two input streams. The output of the correlation-layer is called a *cost-volume*. In order to make the method computationally tractable, the maximum displacement for patch comparison is limited. This means the maximum optical flow which FlowNetC can register is also limited. Skip connections are used in both FlowNetS and FlowNetC to transfer information from a resolution level in the contracting part to its corresponding resolution in the expansive part. The contracting part of both FlowNetS and FlowNetC can be seen in Figure 6.3 and the expansive part in Figure 6.4.

Each convolutional block consists of a bank of convolutional filters and a ReLU activation function. In the contractive part of the network the filter size decreases, while the number of feature maps increases. For the reduction of spatial resolution, each convolution in the contracting part uses a stride of 2[2]. In order to increase the resolution in the expansive part, a stride of 2 is used for the up-convolutions as well. The authors generate a synthetic dataset in order to train their CNN (which will be further discussed in subsection 6.4.1). In Ilg et al., 2017 it was found that FlowNetC consistently outperforms FlowNetS[3]. In Table 6.1 a detailed breakdown of the performance of FlowNetS, FlowNetC and SpyNet (a pyramid spatiotemporal filter-based CNN discussed next section) can be seen. From this table it can be seen that the estimates produced by FlowNetS are more blurry and the error for large velocities is significantly higher than FlowNetC.

[2]According to Springenberg, Dosovitskiy, Brox and Riedmiller, 2014, the max-pooling operation in an encoder-decoder network can be replaced by an increased stride without loss in accuracy on several image recognition benchmarks.
[3]Originally, Dosovitskiy et al., 2015 report similar performance of FlowNetS and FlowNetC. However, Ilg et al., 2017 later conclude a mistake in training FlowNetC is made thus show that FlowNetC significantly outperforms FlowNetS.

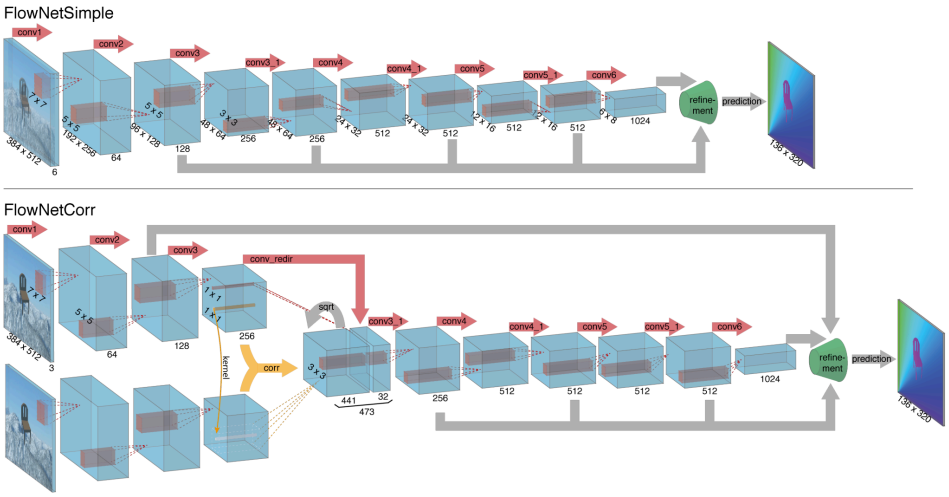


Figure 6.3: (top) The contractive part of the architecture of FlowNetS. Two RGB pictures are convolved with several layers of convolutional filters followed by a stride of 2. (bottom) The contractive part of the FlowNetC architecture with three convolutional layers which share identical weights. In the correlation layer, patchwise multiplicative similarity scores are computed. Taken from Dosovitskiy et al., 2015.

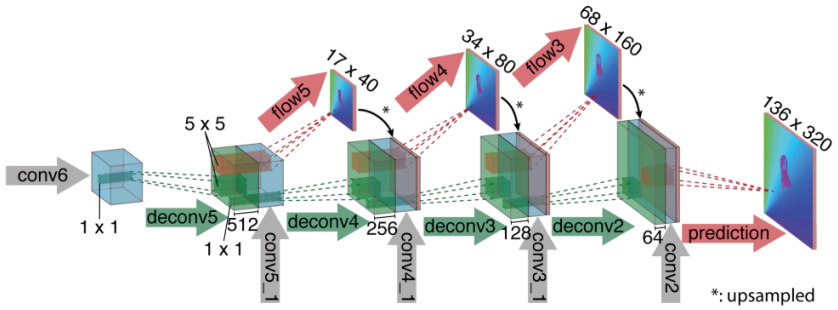


Figure 6.4: Expansive part of the architecture used in both FlowNetS and FlowNetC. Up-convolution is used to obtain a high resolution pixel-wise prediction. Taken from Dosovitskiy et al., 2015.

Model	all	d_{0-10}	d_{10-60}	d_{60-140}	s_{0-10}	s_{10-40}	s_{40+}
FlowNetS+ft	6.96	5.99	3.56	2.19	1.42	3.81	40.10
FlowNetC+ft	6.85	5.57	3.18	1.99	1.62	3.97	33.37
SpyNet+ft	6.64	5.50	3.12	1.71	0.83	3.34	43.44

Table 6.1: Detailed breakdown of the performance of SpyNet, FlowNetS and FlowNetC on the MPI-sintel clean pass for different velocities (s) and distances (d) from motion boundaries. '+ft' corresponds to trained on the FlyingChairs dataset and finetuned on the MPI-Sintel clean pass (see Section 6.4) . Values correspond to AEE per breakdown element. Note the decreased performance at high velocities for the spatiotemporal filter-based CNNs (SpyNet and FlowNetS) and near motion boundaries. The relative error near motion boundaries as fraction of all AEE is also higher for FlowNetS and SpyNet. Taken from Ranjan and Black, 2017.

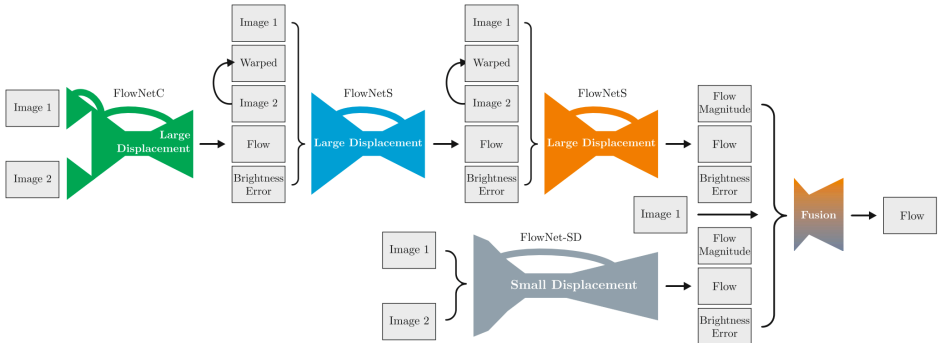


Figure 6.5: Architecture of FlowNet2. One FlowNetC and two FlowNetS architectures are stacked in series and combined in parallel with a single FlowNetSD architecture. Their output is fed to the Fusion architecture to produce a final flow estimate. In the FlowNet2-CSS architecture the two input images along with the warped image, initial flow estimate and brightness error[4] are concatenated and used as input for the intermediate FlowNetS architectures. The braces indicate concatenation of different elements. Taken from Ilg et al., 2017.

Domain knowledge: warping, pyramid structure, and flow refinement

In a follow-up, Ilg et al., 2017 use a stacked architecture, warp the target image towards the reference image using intermediate optical flow and generate more synthetic training data with a higher degree of complexity (see section 6.4.1). Stacking a network with one FlowNetC and two subsequent FlowNetS models (further referred to as 'FlowNet2-CSS') performs best. However, FlowNet2-CSS still performs poorly at estimating small displacements, and therefore a new architecture called 'FlowNetSD' is designed. The FlowNet2-CSS and FlowNet2-SD output are fed into a 'Fusion' network to obtain the final flow estimate. The complete architecture is called 'FlowNet2' as can be seen in Figure 6.5.

Also, the network width of FlowNetS versus performance is investigated, and a wider network (which corresponds to an increased amount of convolutional filters) does not significantly improve performance. A network width factor of $\frac{3}{8}$ produces fast execution times while still producing reasonable results. FlowNet2 achieves near state-of-the-art performance at a runtime of two orders of magnitude lower. It should be noted that the runtime is compared to other methods such as EpicFlow (Revaud et al., 2015) and Deepflow (Weinzaepfel et al., 2013) which are executed on a CPU while FlowNet2 runs on a GPU.

'SpyNet' (Ranjan & Black, 2017) addresses the model size issue of FlowNet by using a spatial pyramid network and warping of the target image towards the reference image in between different pyramid levels. SpyNet uses a spatiotemporal filter-based approach similar to FlowNetS. The architecture of SpyNet can be seen in Figure 6.6. They find that unlike the filters in FlowNetC the filters found in SPyNet resemble Gabor filters, as can be seen in Figure 6.7, and most are equally sensitive to all color channels and thus appear grayscale. Because SpyNet uses a pyramid structure, it runs into the well-

[4]The difference between the reference and the target image warped with the previously estimated flow.

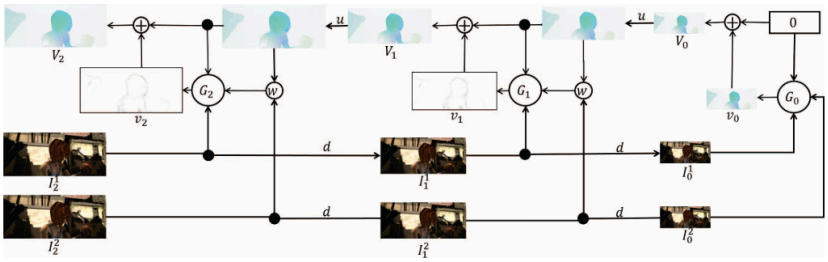


Figure 6.6: Architecture of SpyNet for a 3-level pyramid network. The G_0 network produces an initial flow estimation v_0 using the images I_0^1 and I_0^2 as input which correspond to a downsampled version of the original input images I_2^2 and I_1^2. The initial flow estimate v_0 is upsampled and used to warp I_1^2. Then, the output of G_1, v_1, is added to the upsampled flow V_0 which leads to V_1. This process repeats in every layer in the pyramid. Taken from Ranjan and Black, 2017.

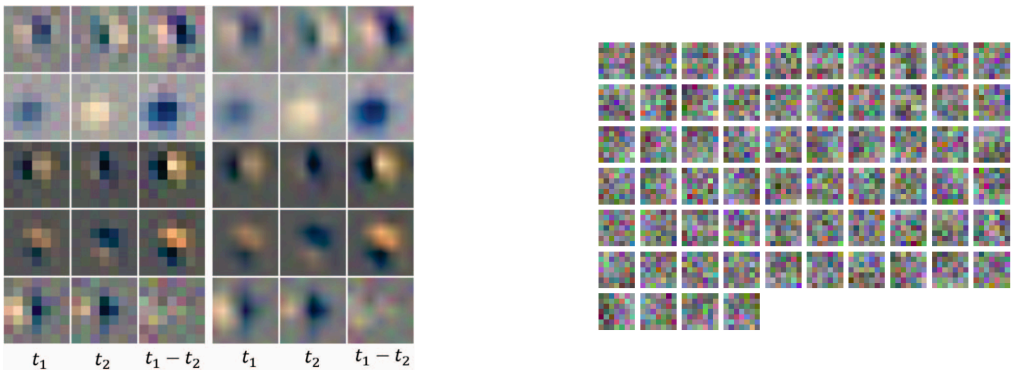


Figure 6.7: (left) Visualization of the filters of the first convolutional layer of the third level of the pyramid of SpyNet. The left and right filters are upsampled using nearest-neighbor and bilinear interpolation respectively. Note that $t_1 - t_2$ represents the temporal difference between the spatial filters. The filters resemble second derivative Gaussian or Gabor filters. Taken from Ranjan and Black, 2017. (right) Filters taken from the first layer of FlowNetC. The filters show a high frequency structure unlike the classic spatiotemporal filters. Taken from Dosovitskiy et al., 2015.

known limitation for dealing with large motions of small scale structures. Small scale objects that move with a high velocity thus often result in erroneous flow estimations.

Sun, Yang, Liu and Kautz, 2018 use a correlation-based architecture, similar to the one in FlowNetC, feature warping, pyramid structure, and a context network for flow refinement. However, in their architecture, called 'PWC-Net' they compute the cost-volume after six convolutional layers instead of three like FlowNetC. Thus the search range is smaller while the receptive field is larger. After obtaining an initial flow estimate, the authors warp the feature maps of the lower level convolutional layers. It should be noted that PWC-net also fails to detect large motion of small scale objects, which is a consequence of the pyramid structure. In a similar manner to Güney and Geiger, 2017, the initial flow estimate is used as an input to a context network which makes use of dilated convolutions which increase the receptive field in order to integrate context information into the final flow estimate (Yu & Koltun, 2015). Note that PWC-Net is often used as a starting point for subsequent researchers.

In both T. W. Hui, Tang and Loy, 2018 and Zweig and Wolf, 2017, the authors show that a fully convolutional network (without stride and max-pooling) is able to perform flow refinement efficiently. T. W. Hui et al., 2018 use flow refinement after their correlation-based architecture performs an initial flow estimation. Using the warped feature maps and initial flow estimate, the flow is further refined to sub-pixel level. Zweig and Wolf, 2017 show that a data-driven approach for sparse to dense interpolation using a (sparse) optical flow, edge map as an extra input and multi-layer loss outperforms EpicFlow. The architecture of Zweig and Wolf, 2017 can be seen in Figure 6.8. Note that this architecture is not a fully end-to-end trained CNN for optical flow estimation. Instead, it interpolates (sparse) flow maps to

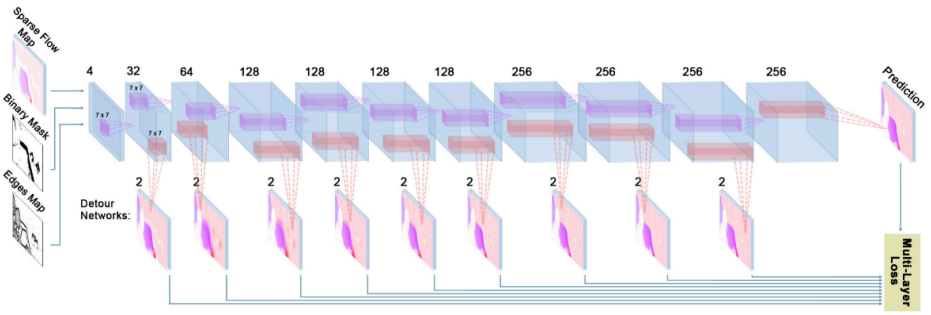


Figure 6.8: Fully convolutional network for flow field refinement. An initial (sparse) flow field is used as input along with an edge map and binary mask containing all the missing pixels and a multi-layer loss is used for training. Taken from Zweig and Wolf, 2017.

a dense flow output. This network serves as an explanation of the decoder part used in both FlowNetS and FlowNetC.

Occlusion estimation

Ilg, Saikia, Keuper and Brox, 2018 extend FlowNet2 to perform joint estimation of occlusions, motion boundaries, and optical flow. The authors include residual connections in between the different models in the stacked architecture and omit the brightness error as inputs. They conclude that joint estimation of optical flow and occlusions does not improve nor degrade the results. Based on this finding, they conclude that FlowNet already implicitly performs all necessary occlusion reasoning and by making the occlusion output explicit, the occlusion output is obtained as an additional output whilst performance stays the same.

Neoral, Šochman and Matas, 2019 extend the architecture of PWC-Net to include information from multiple frames and improve occlusion performance. They perform an explicit occlusion estimation based on the output of the cost volume and the optical flow estimation from the previous frame pair. They perform occlusion estimation based on the cost volume because intuitively when the cost for all nearby displacements is high for a pixel, it is likely occluded in the next frame. Because they use the cost-volume instead of optical flow for their occlusion prediction, they avoid the use of flow already corrupted by occlusion. The authors set a new state-of-the-art on occluded regions on the MPI-sintel Final pass.

Liu, Lyu, King and Xu, 2019 also extend the architecture of PWC-Net to include multiple frames for optical flow estimation and improved occlusion estimation. However, they use a self-supervised approach using two separate networks called the 'non-occluded-model' (NOC-model) and the 'occluded-model' (OCC-model). Where the former is focused on accurate flow estimation for non-occluded pixels. This output is used as a guide for the learning of optical flow of occluded pixels (OCC-model) where the ground truth is optical flow with self-induced occlusions. This method is currently overall state of the art for all both supervised and unsupervised methods on the MPI-Sintel Final pass even though the method of Neoral et al., 2019 performs significantly better on occluded pixels.

Uncertainty estimation

Motivated by the claim that uncertainty information is vital makings decision based on supervised learning estimates, 'FlowNetH' is introduced by Ilg, Ozgun et al., 2018 which produces real-time uncertainty estimates for optical flow estimation. FlowNetH is based on the architecture of FlowNetC. The authors recognize the drawbacks of black-box learning-based approaches and state there is no guarantee that a CNN for optical flow estimation such as FlowNet2 will work under challenging scenarios.

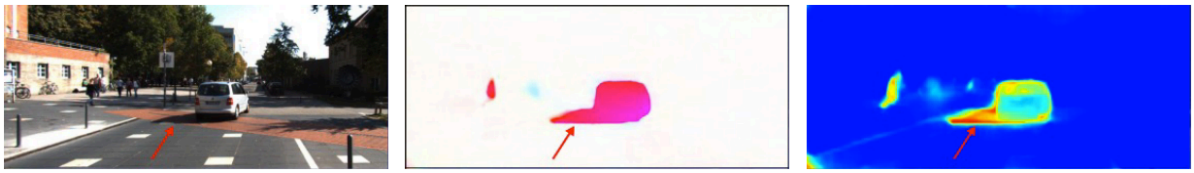


Figure 6.9: Uncertainty and optical flow estimation by FlowNetH. (Left) reference image of image pair taken from KITTI 2015. (Middle) The estimated optical flow. (Right) Uncertainty estimation, higher values correspond to red. Note that the shadow has a high uncertainty value unlike the car. Taken from Ilg, Ozgun et al., 2018.

Inspired by the approach of Rupprecht et al., 2017, FlowNetH is adapted to make multiple hypotheses and Winner-Takes-All loss (Guzmán-rivera, Batra & Kohli, 2012) is used to only penalize the best prediction. FlowNetH produces state-of-the art uncertainty estimations and the authors demonstrate that the uncertainty estimation is high for various difficult cases[5], one of them can be seen Figure 6.9.

6.3.2 Signal processing principles

In Teney and Hebert, 2016 the authors design a shallow CNN based on signal processing principles and revisit the approach of Heeger, 1988. Instead of producing an output optical flow map that produces pixel-wise estimations, their approach outputs a distributed representation of orientations and speeds per pixel. The network consists of two convolutional layers, 1 pooling layer and two pixelwise weights (1x1 convolutions). This allows them to estimate transparent and overlapping motions which most traditional optical flow methods are not capable of. They design their CNN in such a way that it is invariant to additive brightness changes, in-plane rotations, local image phase, and local image structure. Local image structure invariance is desired to account for intensity differences of patterns at different orientations moving with a different velocity magnitude and direction (e.g., a grid pattern of horizontal lines crossing fainter vertical ones). The result is a relatively shallow CNN with four convolutional layers. The consequence of their design is that the estimation of large motions is limited due to the small receptive field size. Therefore, a coarse-to-fine warping strategy is incorporated into the architecture, which allows the network to estimate the flow iteratively. The approach fails near occlusions boundaries, and the authors conclude that proper performance near occlusion boundaries requires reasoning over a broad temporal and/or spatial extent which their shallow CNN is not able to do. The authors conclude that the different design choices in FlowNet and their CNN seem complimentary and that it would be interesting to investigate a combination of the two.

6.4 Training CNNs for optical flow estimation

In this Section the different aspects which come into play when training CNNs for optical flow estimation will be discussed. Firstly, the hugely contributing factor of synthetic training datasets is presented. Secondly, the data augmentation and learning rates are discussed.

6.4.1 Synthetic training datasets

Even though MPI-Sintel has over a 1,000 image pairs available for training, Dosovitskiy et al., 2015 found this amount not sufficient for training a CNN and rendered their own synthetic training dataset called 'FlyingsChairs'. Chairs were chosen because they come in many different shapes (topologically diverse) and textures and the fact that they are not semantically similar to real-world scenes. Meaning, that the trained networks are able to generalize point correspondence estimation from the synthetic data

[5] https://www.youtube.com/watch?v=HvyovWSo8uE

Figure 6.10: An example of the three different lighting models used for generating a synthetic dataset used to train FlowNetC and test on MPI-Sintel. (left to right) The dynamic, static and shadeless lighting model respectively. Taken from Mayer et al., 2018.

to real-world scenes. Note that FlyingChairs only contains 2D affine transformations. Inspired by the success of FlyingChairs a new dataset called 'FlyingsThings3D' (Mayer et al., 2016) was made containing 3D motion, 3D objects models, camera motion, and realistic lighting. In Ilg et al., 2017 FlowNetS was trained on both these datasets, and it was found that training on FlyingChairs outperformed training on the more realistic and diverse FlyingsThings3D dataset.

This was the motivation for an extensive ablation study by Mayer et al., 2018 to determine what factors make a synthetic dataset for optical flow estimation successful. Regarding the superior performance of FlyingChairs, the authors reason that introducing a too sophisticated dataset too early might confuse the network because it has not yet developed an understanding of the concept of finding point correspondences. Therefore, curriculum learning[6] is used in Ilg, Ozgun et al., 2018 to first train the network on FlyingChairs and then on FlyingThings3D. They also conclude that diversity for a synthetic dataset is important, and having knowledge about the camera helps. When camera distortions, such as Bayer-artifacts and lens-distortion, were introduced in the synthetic dataset, the performance improved on real data. Furthermore, three different lighting models in the synthetic dataset were used to test the effect on performance. One model without lighting or shading, a static lighting model where a fixed 'shadow texture' is used for each object and a dynamic model with a single source shining onto the scene from a random direction which also includes specular highlights. The network trained on the static lighting model dataset performs best even though the testset (MPI-Sintel) contains both specular highlights and Lambertian surfaces. While the network can effectively exploit the latter, for the former, it needs to distinguish between different surface materials and this confuses the network. Based on their ablation study, Mayer et al., 2018 conclude that synthetic training data can be improved if it is possible to reason about the target domain or testset. They did find that when they created such synthetic training datasets with extra effects they noticed a drop in performance on the original datset. This implies that there is no single best general-purpose synthetic training dataset. A conclusion which they find 'disappointing'.

Recently, even more synthetic datasets have become available for optical flow concerning a specific application. Like Virtual KITTI (Gaidon, Wang, Cabon & Vig, 2016) for autonomous driving and SceneNet for indoor scenes (McCormac, Handa, Leutenegger & Davison, 2017). An overview of optical flow datasets, both synthetic and natural can be found in Table 6.2.

6.4.2 Data augmentation and learning rates

Data augmentation is deemed a crucial step in the training of CNNs for optical flow estimation(Dosovitskiy et al., 2015; Ilg et al., 2017; Mayer et al., 2018; Sun et al., 2018). Mayer et al., 2018 perform an ablation study for augmentations on color and geometry (cropping, rotating, etc.) on one frame or both frames and conclude that all the augmentations used to train FlowNetS, FlowNetC and FlowNet2 work complimentary. Comparing the training of FlowNetC with and without augmentation, it is found that a 100-fold reduction in the training data with augmentation still provides better results than training without data augmentation. However, the best results were achieved when using both augmentation

[6]Training a network on a gradually increasing complex task

Dataset	Published in	Synthetic/natural	Private testset	#Frames for training	Resolution
UCL	Mac Aodha, Brostow and Pollefeys, 2010	S		4	640 x 480
Middlebury	Baker et al., 2011	N	✓	8	640 x 480
KITTI 2012	Geiger, Lenz and Urtasun, 2012	N	✓	194	1,242 x 375
MPI-Sintel	Butler, Wulff, Stanley and Black, 2012	S	✓	1,064	1,024 x 436
KITTI 2015	Menze and Geiger, 2015	N	✓	200	1,242 x 375
FlyingChairs	Dosovitskiy et al., 2015	S		21,818	512 x 384
FlyingThings3D	Mayer et al., 2016	S		22,872	960 x 540
Monkaa	Mayer et al., 2016	S		8,591	960 x 540
Driving	Mayer et al., 2016	S		4,392	960 x 540
Virtual KITTI	Gaidon, Wang, Cabon and Vig, 2016	S		21,260	1,242 x 375
HD1K	Kondermann et al., 2016	N	✓	3,563	2,560 x 1080
SceneNet RGB-D	McCormac, Handa, Leutenegger and Davison, 2017	S		~5,000,000	320 x 240

Table 6.2: Overview of both synthetic and natural datasets with dense optical flow ground truth. Note that datasets with a private testset can be used a benchmark. The benchmark most often used is MPI-Sintel. Adapted from Mayer et al., 2018.

Figure 6.11: Flow predictions for different image pairs from MPI-Sintel. FlownetC is retrained using training schedule with a disruptive learning rate.The retrained CNN is called FlowNetC+. Taken from Sun, Yang, Liu and Kautz, 2018.

and as much training data as possible.

Sun et al., 2018 modify the augmentation schedule of FlowNet and report improved performance; horizontal flips are added, and they no longer add additive Gaussian noise to training data. Also, they use a disruptive learning rate schedule for their PWC-Net architecture. A disruptive learning rate schedule is a schedule which also increases the learning rate over time at certain intervals. When they applied this disruptive schedule to FlowNetS and FlowNetC, they were able to reduce the AEE on MPI-Sintel by about $\approx 20\%$ and $\approx 50\%$ respectively, which results in FlowNetC even outperforming FlowNet2. Qualitative results can be seen in Figure 6.11.

7

Synthesis of literature

In this Chapter a synthesis of the conducted literature study is performed. All the relevant literature for the understanding of spatiotemporal filter-based CNNs has been collected and analyzed, and the knowledge gap can be identified.

7.1 Conventional optical flow estimation methods

Local differential methods have proven to be fast, reliable, and computationally tractable for computing sparse optical flow fields. For this reason, it is often used in robotic navigation applications for vehicles such as MAVs. The drawback of second-order differential methods is that they are not able to deal with deviations from translation such as affine motion. Global methods produce dense flow fields using additional constraints. However, the very restrictive isotropic global smoothness assumption of B. K. Horn and Schunck, 1981 does not hold for realistic scenes. Barron et al., 1994 also points out that the local smoothness assumption of Lucas and Kanade, 1981 is more stable to noise than the global smoothness assumption. Ever since the work of Brox et al., 2004, global differential methods have dominated the optical flow estimation benchmarks for over a decade. Various improvements have been made to deal with long-range motions and occlusions. However, a significant drawback is the long computation-time of these methods. Differential methods are based on assumptions, such as the brightness constancy assumption (equation 3.1), which are coarse approximations to reality and these assumptions limit the performance. Research has focused on improving these assumptions and making the methods more robust to deviations from these assumptions. Thus, leading to slow but steady progress.

Heeger, 1988 derives an expected response of Gabor filters based on translating white noise. They sacrifice the main advantage of frequency-based methods; the ability to resolve velocity components of intensity patterns at different orientations (e.g. a grid pattern of vertical lines crossing fainter horizontal ones). Furthermore, the amplitude signal is not stable to deviations from 2D motion and illumination changes (D. Fleet & Jepson, 1989). Phase is more robust to deviations to global scene illumination changes than amplitude and differential methods. However, both methods produce blurry flow maps due to the uncertainty relation and this has historically played in favor of the global differential methods(Teney & Hebert, 2016). Note that various time-varying image intensity phenomena are more easily described in the frequency domain, including motion blur, aliasing, and occlusion (S. Beauchemin & Barron, 2000). The method of D. Fleet and Jepson, 1989 can be seen as a local differential method applied to the phase signal of Gabor kernels. Barron et al., 1994 also conclude that phase-based methods are more sensitive to aliasing than local differential-based methods.

Spatiotemporal sampling rates are of great importance. If sampling rates are not high enough aliasing can occur in both the spatial and temporal domain (S. S. Beauchemin & Barron, 1995). Aliasing-free imagery is important for differential-based methods in order to compute accurate derivatives for correlation-based methods to reduce the search area because of the increased temporal resolution and for frequency-based methods to limit the number of frequency components in the Fourier domain due to aliasing. When it is not possible to increase the temporal sampling rate and obtain accurate derivatives, it is natural to use coarse-to-fine correlation-based matching approaches. It should be noted that these approaches do suffer from the inherent limitation of coarse-to-fine schemes; getting stuck in local optima.

7.2 Learning-based methods

Machine-learning-based methods
Sun et al., 2008 was among the first to propose an end-to-end trained learning-based optical flow estimation method. Although the first results seemed promising. It did not fully show the potential of learning-based approaches at the time. Wulff and Black, 2015 generated more training data using a computationally expensive optical flow estimation method to generate ground-truth data from four Hollywood movies. Interestingly, their method was able to outperform the method which was used to obtain the ground-truth flow. The performance of their method was still-below the state-of-the-art, however.

CNN-based methods
Ever since the work of Dosovitskiy et al., 2015 the research interest in CNN-based optical flow estimation methods has surged. Due to the generation of synthetic datasets, the use of an encoder-decoder CNN architecture and more computing power, CNN-based optical flow estimation methods have become the state-of-the-art on competitive benchmarks such as MPI-Sintel(Butler et al., 2012). Dosovitskiy et al., 2015 propose a correlation-based architecture which performs pixel-wise similarity matching (FlowNetC) and a spatiotemporal filter-based architecture which takes two stacked images as input (FlowNetS). Ilg et al., 2017 conclude that FlowNetC outperforms FlowNetS[1]. Subsequent researchers have often focused on improving the performance of the correlation-based architecture (T.-W. Hui, Tang & Loy, 2019; Sun et al., 2018), with the noteable exception of Ranjan and Black, 2017 who introduced a spatiotemporal filter-based architecture with a pyramid structure.

Real-world application considerations
Ilg et al., 2017 report that one of the limitations for real-world applications is the fact that both FlowNet architectures cannot detect small (sub-pixel) motions. For traditional methods, small motions are easier. The MPI-Sintel and FlyingChairs dataset both contain relatively large motions and therefore a new dataset containing small motions is constructed called ChairsSDHom. FlowNetC performs pixel-wise similarity matches and is therefore not able to extract sub-pixel motion. Spatiotemporal based convolutional filters are able to extract subpixel motion(T.-W. Hui et al., 2019). However, there is no quantitative evaluation on the sub-pixel performance of FlowNetS and FlowNetC. In order to estimate sub-pixel motions, Ilg et al., 2017 modify the original architecture of FlowNetS by using smaller convolutional kernels in the first few layers of new network, and extra convolutional layers are added in the expansive part of the network to deal with noise.

Also, Mayer et al., 2018 found that when the camera defects which occur in real imaging (lens distortion, blur and Bayer-artifacts) are synthesized into the synthetic training dataset the performance improves. This is an advantage of learning-based methods over conventional optical flow estimation methods. Because conventional methods require imagery free of camera distortions for the computation of accurate derivatives.

[1] Originally, Dosovitskiy et al., 2015 do not provide the complete breakdown of the performance on the MPI-Sintel dataset of their FlowNet architectures without variational refinement. They state that the performance of FlowNetC on larger motions is worse than that of FlowNetS. However, in Ilg et al., 2017 it is stated that in their previous work they made a mistake and conclude that Dosovitskiy et al., 2015 did not train FlowNetS under the same conditions as FlowNetC. This claim is supported by the work of Ranjan and Black, 2017 who reports the detailed performance breakdown of SpyNet, FlowNetC and FlowNetS on the MPI-Sintel clean pass.

Occlusion reasoning
The shallow end-to-end trained CNN of Teney and Hebert, 2016 is able to estimate optical flow but it fails near occlusion boundaries. Therefore, the authors conclude that occlusion estimation requires reasoning over a larger temporal or spatial span. Ilg, Saikia et al., 2018 train FlowNetS to perform occlusion estimation from just two input images without giving optical flow as input and conclude this is feasible. When adding the flow as input, the estimates clearly improve. Furthermore, they train variants of FlowNetC to perform optical flow estimation, occlusion estimation, and a version that estimates them jointly. They report no noticeable drop in performance when jointly estimating optical flow and occlusions and conclude, based on this finding, that both FlowNet architectures already perform occlusion reasoning.

Knowledge gap
While the workings of the correlation-based architecture are known due to the explicit patch-wise similarity computation, there has been no research on the workings of the spatiotemporal filter-based architecture. Only a visualization of the filters of the first layer of SpyNet has been done by Ranjan and Black, 2017 where they claim that the filters of the first layer resemble Gabor filters. It remains unclear if FlowNetS uses Gabor filters to estimate motion and if FlowNetS suffers from blurry flow maps due to the uncertainty relation. In the detailed performance breakdown in Table 6.1 it can be seen that the flow maps produced by FlowNetS are more blurry than FlowNetC. However, Zweig and Wolf, 2017 have shown that a fully convolutional network can be used to interpolate motion cues inside motion boundaries. Such a mechanism is thus able to overcome the main drawback of filter-based motion estimation. In order to gain a better insight into the workings of FlowNetS synthetic input is generated to examine the behavior of FlowNetS in Part III.

III

Preliminary Evaluation of Spatiotemporal filter-based CNNs

8

Methodology

The goal of these preliminary evaluations is to gain insight into how spatiotemporal filter-based CNNs perform optical flow estimation. These experiments serve as study of the behavior of these networks and what their limitations are. The error characteristics of these networks can also be compared to conventional methods to determine their similarities and differences. The latter does not provide a definitive answer as to how these CNNs but it can provide an indication.

This Chapter firstly present an outline of the analysis in section 8.1. Secondly, the specification and implementation of the models used in this preliminary evaluation is given in Section 8.2. Lastly, the creation of synthetic optical flow ground truth will be discussed in Section 8.3.

8.1 Outline of the experiments

In this Section an outline of the experiments will be given. These experiments serve to answer (parts of) the research question given in Chapter 1. A total of five experiments will be performed which, are described below:

- **Experiment 1:** Visualizing the first layer of the filter in the spatiotemporal filter-based CNNs.
- **The goal of experiment 1:** Gain insight into what kind of signals these networks exploit. In the past, it has been shown that often spatiotemporal filters emerge form learning-based approaches.
- **Experiment 2:** A breakdown of the AEE versus the motion magnitude for simple translational motion.
- **The goal of experiment 2:** To compare the optical flow error characteristics to conventional methods and to examine the performance in for different velocities.
- **Experiment 3:** A breakdown of the AEE versus the angle with a unit circle of constant magnitude.
- **The goal of experiment 3:** To determine if the filters in the network are orientation sensitive and if they are stable for different orientations of constant magnitude.
- **Experiment 4:** A breakdown of the AEE versus the scale of a diagonally translating object.
- **The goal of experiment 4:** To determine how the networks cope with the aperture problem and to estimate their performance on large motion of small-scale structures.

- **Experiment 5:** A breakdown of the AEE versus the amount of occlusion for a horizontally translating object.
- **The goal of experiment 5:** To estimate the occlusion 'reasoning' abilities of the networks.

The results of these experiments will be given in Chapter 9.

8.2 Model specification

In this Section the models which are going to be evaluated are discussed. The models used for evaluation are spatiotemporal filter-based CNNs (FlowNet2S and SpyNet), a correlation-based CNN (FlowNet2C) and a matching-based global energy method (LDOF). Note that FlowNet2S differs from FlowNetS in that it is first trained on FlyingChairs and then on FlyingsThings3D. FlyingChairs contains only planar motion while FlyingsThings3D contains true 3D motion. Ilg et al., 2017 conclude that FlowNetS and FlowNetC were not trained under the same conditions, and therefore a comparison between these models is not deemed feasible. Note that SpyNet was only trained on FlyingChairs. Because the experiments only include planar motion and the error characteristics are of primary interest this is not deemed an issue. For FlowNet2S and FlowNet2C the PyTorch[1] implementation of NVIDIA[2] was used. The weights of the original model (Ilg et al., 2017), which was implemented in Caffe [3], were converted to PyTorch. The implementation was verified by checking the output on the training set of the MPI-Sintel clean pass with the performance specified by the authors. Regarding SpyNet (the original model is written in Torch[4]), a PyTorch reimplementation[5] is used with the original weights provided by the author. Note that Torch and PyTorch use Lua and Python as their interface language, respectively. However, they both make use of THNN[6], a part of Torch which was written in C. Lastly, for the LDOF method, an executable from the author's website was used[7].

8.3 Creating synthetic optical flow ground truth

In this Section the methodology for the creation of optical flow ground truth will be given. Note that for the creation of optical flow ground truth the Pillow[8] python module was used. This module provides methods for image manipulation and the creation of geometry. The coordinate system used by this Pillow can be seen in Figure 8.1. Pillow does not support interpolation of a pixel when a geometry does not span a complete pixel. Therefore it is not possible to create sub-pixel accurate ground-truth with this module. It is also due to this inability to deal with sub-pixel displacement that solely rectangular objects were used in the experiments.

Most of the models are trained on the FlyingChairs synthetic dataset. This dataset contains background images taken from Flickr in the categories 'city', 'landscape' and 'mountain'. It was found that the CNNs trained on this dataset performed poorly if a background image was not present. Therefore, an image from the 'mountain' category was retrieved from Flickr and can be seen in Figure 8.2. The image size used in all experiments is 512x384. This is the same size as the images in the MPI-Sintel dataset, a benchmark used by all CNN-based methods.

[1] https://github.com/pytorch/pytorch
[2] https://github.com/NVIDIA/flownet2-pytorch
[3] https://github.com/BVLC/caffe
[4] https://github.com/anuragranj/spynet
[5] https://github.com/sniklaus/pytorch-spynet
[6] https://github.com/torch/nn/tree/master/lib/THNN
[7] https://lmb.informatik.uni-freiburg.de/resources/software.php
[8] https://github.com/python-pillow/Pillow

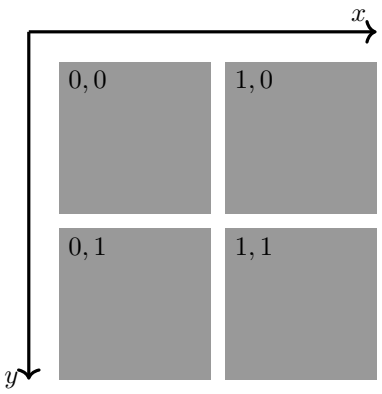


Figure 8.1: The pixel coordinate system used by the Pillow python module. The origin corresponds to the top left corner of the image.

Figure 8.2: Background used for the creation of synthetic optical flow ground truth.

9

Preliminary Results

In this Chapter the results of the preliminary experiments will be discussed. Firstly, the filters are visualized in order to determine if they show any resemblance to spatiotemporal filters. Secondly, the error versus motion magnitude is presented. After which, the orientation sensitivity will be analyzed. Furthermore, the scale up and until the models can resolve the aperture problem will be evaluated and the behavior of network near occluded image structure is examined.

9.1 Filter visualizations

In this Section the filters of FlowNet2S and SpyNet will be visualized. Note that Ranjan and Black, 2017 have already shown in their work that the filters in the first convolutional layer of SpyNet resemble Gabor filters. Therefore, the filter visualization, which can be seen in Figure 9.1, mainly serves as a comparison between FlowNet2S and SpyNet. From the FlowNet2S model, ten filters are randomly selected from the first convolutional layers, which has a total of 64 filters. From the SpyNet model ten filters are also randomly selected but from the first pyramid level. SpyNet is trained sequentially, meaning they train one pyramid level first and then add more pyramid levels at higher resolution with the previous pyramid level as initialization. Therefore, filters at different levels of the pyramid the filters become more defined (Ranjan & Black, 2017). Both the filter the filters of size 7x7 and are bi-linearly upsampled to size 14x14. In Figure 9.1 the weights of both models are visualized. Note that the third and sixth column correspond to the temporal difference. For SpyNet some of the filters clearly resemble Gabor filters (e.g., row 4, 7 and 10). For FlowNet2S the filters do not exhibit such clear resemblance to Gabor filters.

9.2 Motion magnitude

A synthetic test sequence is generated to investigate the characteristics of the error patterns of the different models. In Figure 9.2 the input sequence used for this experiment is visualized. Multiple image pairs are generated for a translating black square with an increasingly higher horizontal velocity component. The AEE for all pixels, inside and outside the square versus the magnitude of the horizontal velocity component can be seen in Figure 9.3. Multiple observations can be made from these three graphs. Firstly, for the LDOF model the AEE is increasing linearly from 2 till about 80 after which it fails. This can be seen even more clearly from the AEE inside the square which spikes around this magnitude. Both FlowNet2S and SpyNet exhibit similar error patterns. The main difference is that SpyNet shows better overall performance for lower velocities and fails more rapidly and at a lower

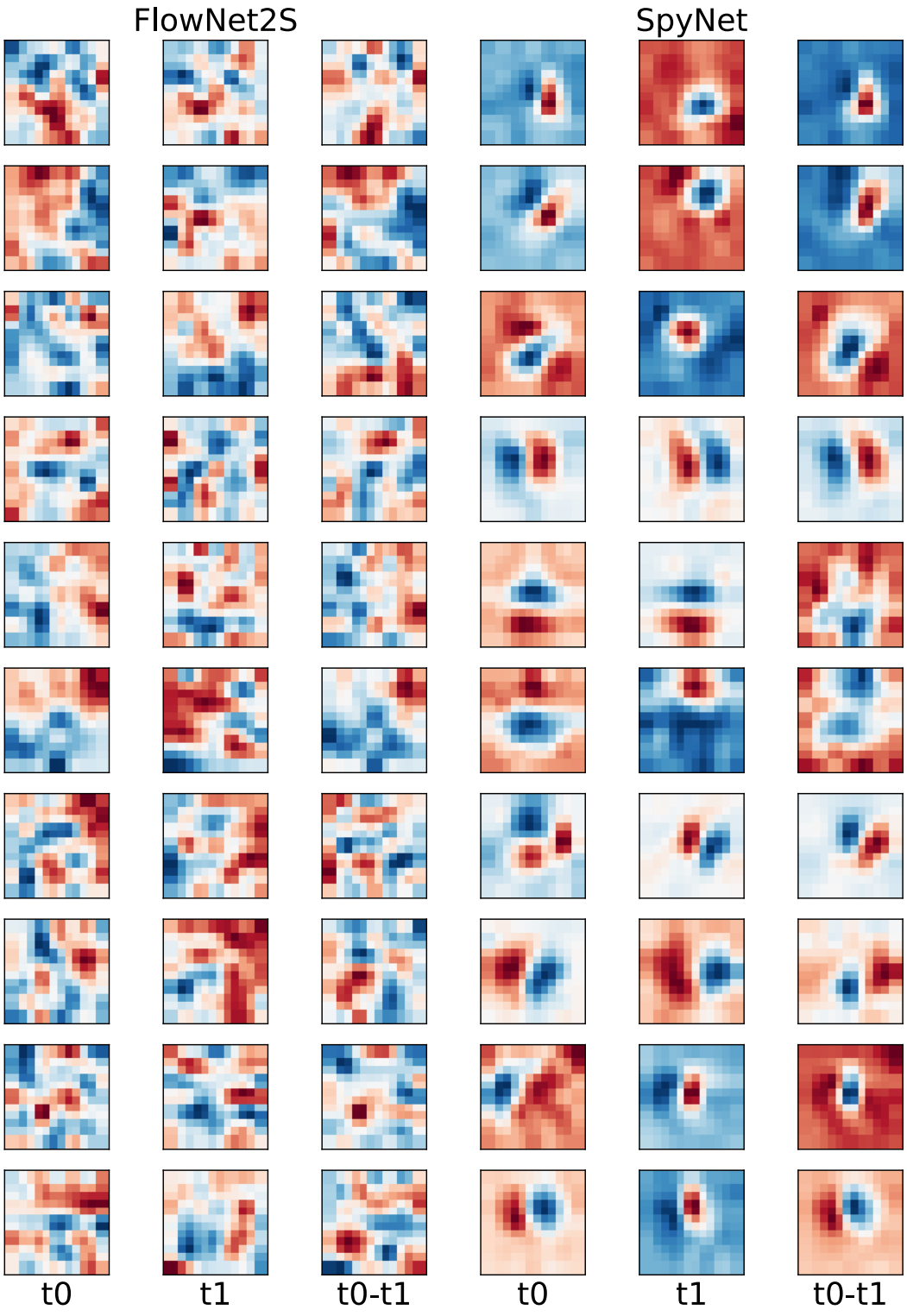


Figure 9.1: (Column 1 to 3) The weights of the first convolutional layer of FlowNet2S for t_0,t_1 and $t_0 - t_1$ respectively. (Column 4 to 6) The weights of the first convolutional layer of SpyNet for the first pyramid level for t_0,t_1 and $t_0 - t_1$ respectively. The rows correspond to different filters. Red details a high value and blue a low value. A relative depth map per filter is used, meaning every entry has its colors scaled to their own filters. These filters are visualized without the bias term added to them. This is because with a relative depth map the addition of a bias term does not influence the visual appearance of the filter.

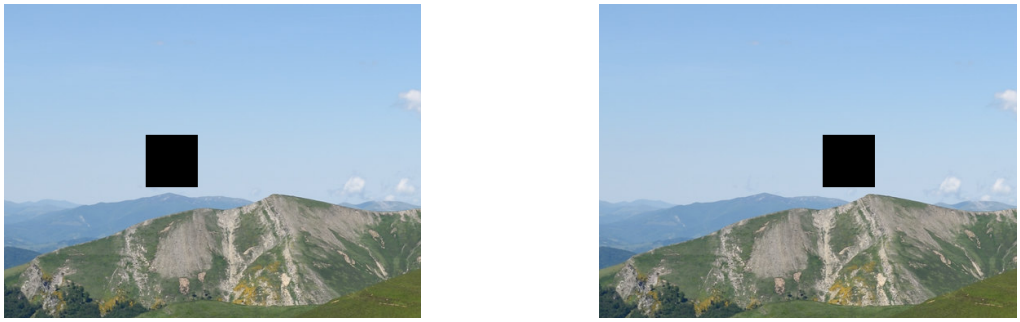


Figure 9.2: Synthetic test sequence used for the motion magnitude experiment. A background with a black square of size 64x64 is translated horizontally, symmetrically about the vertical axis, with an increasingly larger horizontal velocity magnitude. This image pair corresponds to a velocity magnitude of $\mathbf{v} = (100, 0)$.

velocity magnitude. This becomes apparent in the 'AEE inside the square' plot. Furthermore, it can be seen that the AEE for both FlowNet2S and SpyNet increase in a linear fashion until they fail. The error pattern for FlowNet2C outside the circle is quite erratic. It should also be noted that the four different models fail in different ways, and this can be seen in Figure 9.4.

9.3 Orientation sensitivity

In order to test the sensitivity and stability of optical flow estimation with respect to different orientations of constant image translation magnitude another synthetic test sequence is generated. For this sequence, a black square is placed in the center of the image and moved with a constant total velocity magnitude of $\|\mathbf{v}\| = 100$ at different angles with respect to the horizon. Due to the fact that the Pillow library does not support sub-pixel rendering for the angles of $45° + k90°$ the velocity components are rounded to the nearest integer. This results in a total magnitude of $\sqrt{2 * (71)^2} = 100.409$. This difference in magnitude is deemed negligible. The resulting error patterns can be seen in Figure 9.5. Note that the top-left radar plot corresponds to the AEE. The other three radar plots correspond to the deviation in percentage of the AEE to the mean of the AEE for all rotations. It is interesting to see that the models all have the highest deviation from their mean AEE at different orientations even though they were trained on mostly the same data. It should be noted that SpyNet seams particularly unstable at an orientation of 315°. Upon inspection, it was found that for this orientation SpyNet does estimate the correct angle but overestimates the magnitude of the velocity by $\approx 12\%$.

9.4 Aperture and scale problem

To test how the models cope with the aperture problem and with the velocity of small scale structures a square is diagonally translating with constant velocity $\mathbf{v} = (50, 50)$ for increasing square sizes. The results can be seen in Figure 9.6. In the top plot, it can be seen that the behavior is fairly consistent except for the largest square, which moves from edge to edge, which produces poor estimates for FlowNet2S and FlowNet2C. There is no obvious reason as to why they fail this way. The total maximum span of the movement is $334 + 50 = 384$ pixels. Using the model details from FlowNet2S as specified in Table A.1 and a readily available script[1] which allows for the calculation of the receptive field size. The receptive field size is defined as the region in the input space that a particular output feature is affected by. For the 'flow6' layer, the receptive field size was calculated to be 479 pixels. Using Table A.2 the receptive field size was calculated for SpyNet. This resulted in a value of 31 pixels. The lowest level of the pyramid has an input size of 32x24. Thus, when accounting for the number of times the flow estimate is upsampled. The receptive field size is equivalent to $31 * 2^4 = 496$. From the middle

[1] https://gist.github.com/Nikasa1889/781a8eb20c5b32f8e378353cde4daa51

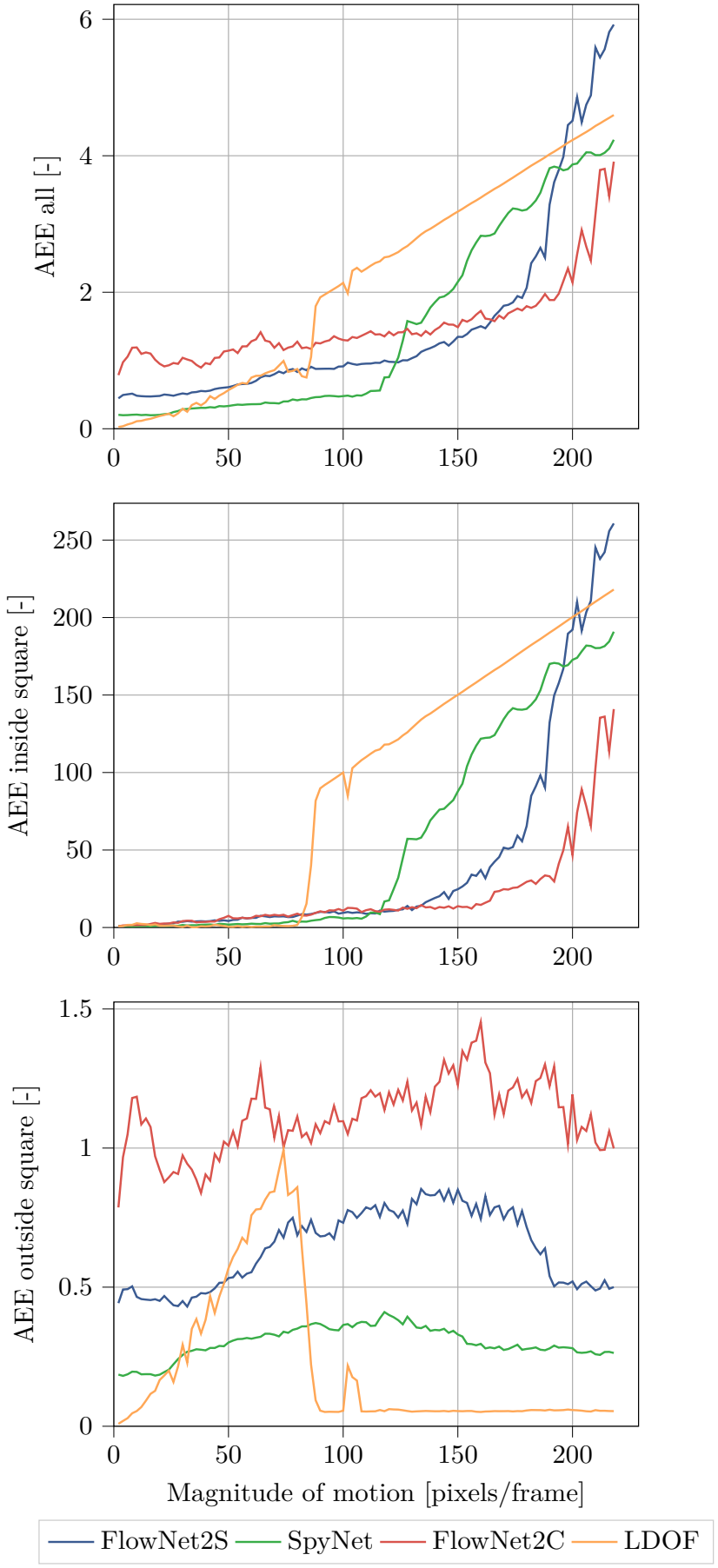


Figure 9.3: (top to bottom) Motion magnitude versus the AEE for all pixels, pixels inside the square and pixels outside the square respectively for the FlowNet2S,SpyNet, FlowNet2C and LDOF models. The magnitude of the square used as input is 64x64 pixels.

Figure 9.4: (all) Flow estimation produced by the four different models for $\mathbf{v} = (218, 0)$ corresponding to a bright red ground truth color. Note that the color coding is similar to Baker et al., 2011 is used an can be found in Appenfix B. (top left) Flow map corresponding to the FlowNet2S model. The model predicts that the square is moving outward (to the left) of the frame. (top right) Flow map corresponding to the SpyNet model. At larger velocities SpyNet has difficulty matching the squares at different timesteps and the estimate contains multiple colors corresponding to different velocity angles. (Bottom right) Flow map corresponding to the FlowNet2C model. Also FlowNet2C has trouble matching the complete patch at high velocities. At high velocities FlowNet2C does have the best performance. (Bottom right) Flow map corresponding to the LDOF model. This model produces a flow estimate corresponding to disappearing edges and appearing texture.

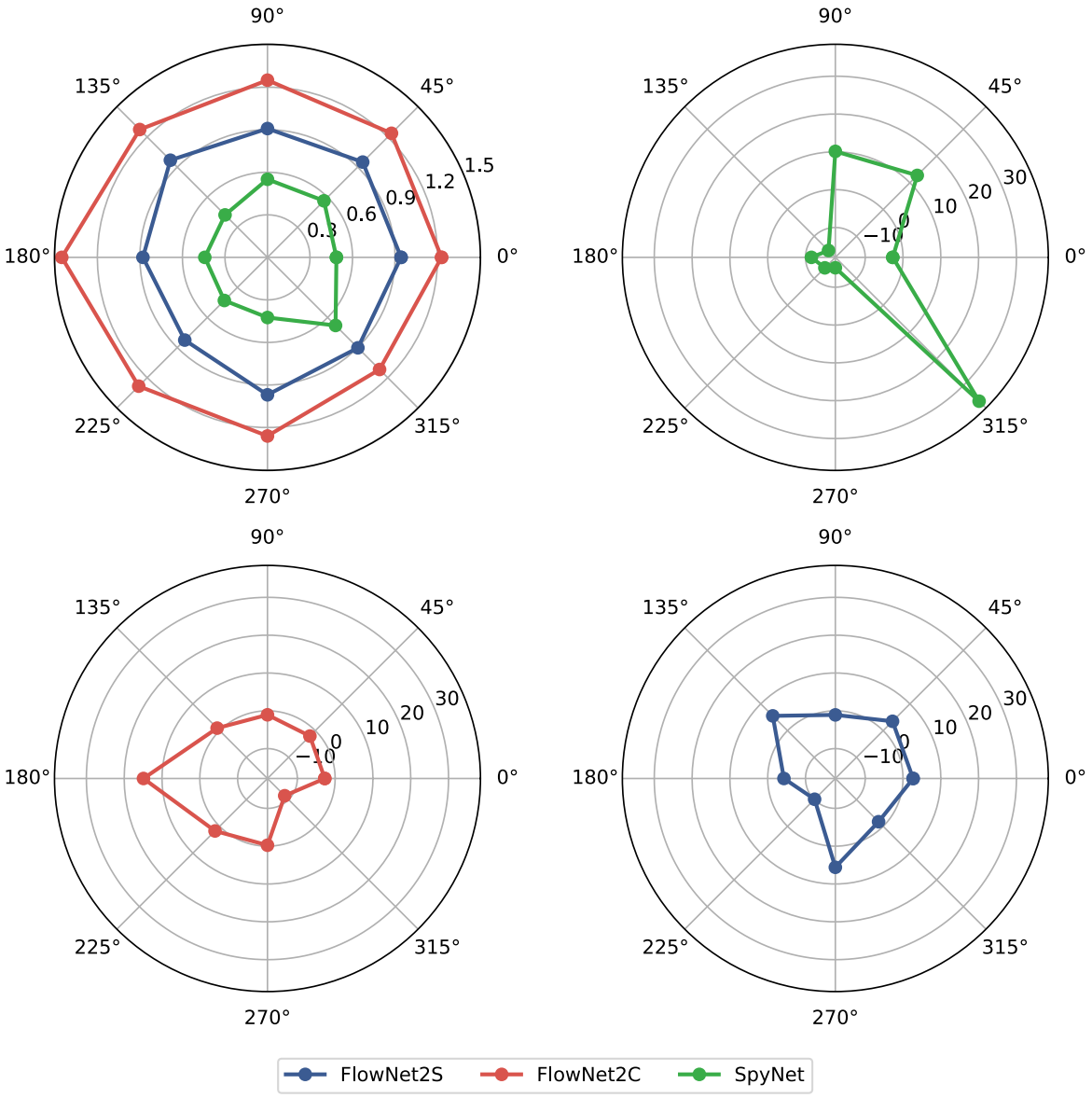


Figure 9.5: (top left) AEE for FlowNet2S, SpyNet and FlowNet2C for a square translating with $\|\mathbf{v}\| \approx 100$ at different orientations with respect to the horizon. (top right clockwise to bottom left) The deviation from the mean AEE in percentage at different orientations of the model for SpyNet, FlowNet2S and FlowNet2C respectively.

plot, it can be seen that all models still have issues with motion larger than the scale of the input. This issue seems to affect SpyNet the most. This is most likely due to their pyramidal structure, which is known to be prone to this issue. One of the main motivations for the U-net architecture of FlowNet2S and FlowNet2C is the fact that it can combine features from multiple scales. However, both models also seem to suffer from the issue of large motion of small scale structures.

9.5 Occlusion

In order to test the occlusion reasoning capabilities, a translating square with velocity $\mathbf{v} = (64, 0)$ is occluded by a white rectangle in the second frame, as can be seen in Figure 9.8, with increasing width. The results can be seen in Figure 9.7. In the top plot, it can be seen that the AEE for all pixels of FlowNet2S is fairly constant. Even if the object is completely occluded in the second frame, it is able to produce an estimate with consistent accuracy. SpyNet is able to cope with the occlusion for lower occlusion rectangle width. The performance starts to deteriorate rapidly after a rectangle width of 40 pixels. The performance of FlowNet2C is the worst for medium occlusion. The reason for this can be seen in Figure 9.9. It is only able to correlate the edges of the square between the two input frames.

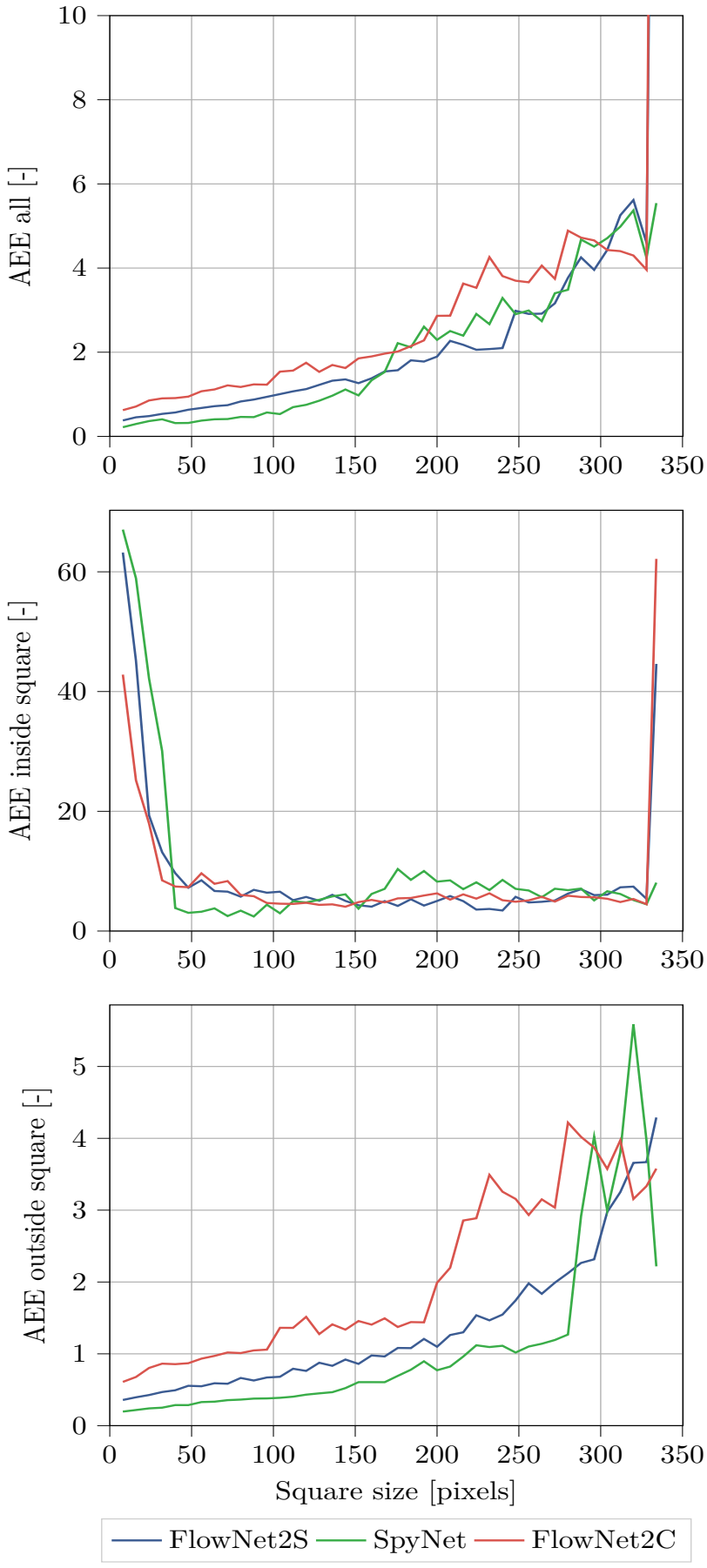


Figure 9.6: (top to bottom) AEE for FlowNet2S, SpyNet and FlowNet2C for a diagonally translating square with velocity $\mathbf{v} = (50, 50)$ symmetrically around the origin versus the size of the square for all pixels, pixels inside the square and pixels outside the square.

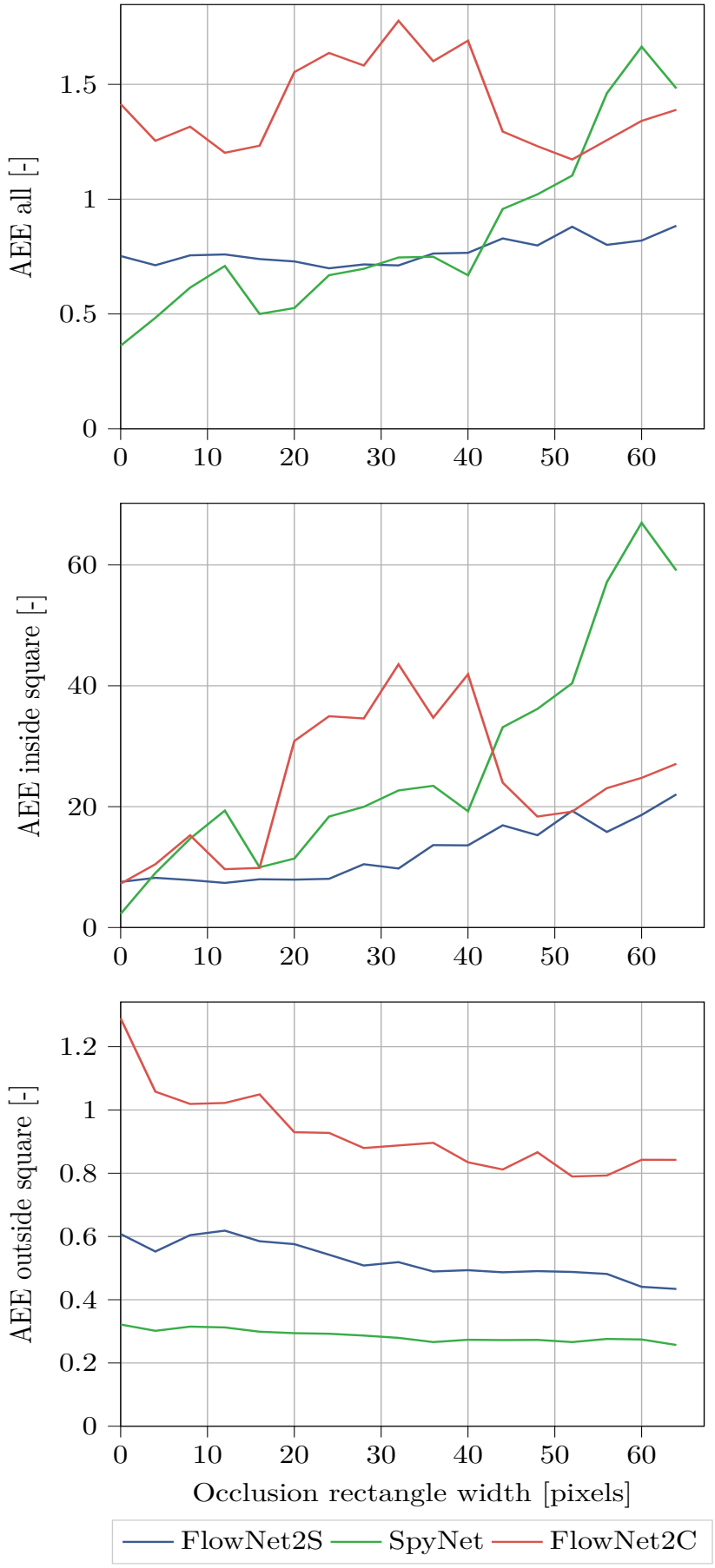


Figure 9.7: (top to bottom) AEE for FlowNet2S, SpyNet and FlowNet2C for a translating square with velocity $\mathbf{v} = (50, 50)$ which is occluded in the second frame by a rectangle of increasing width for all pixels, pixels inside the square and pixels outside the square.

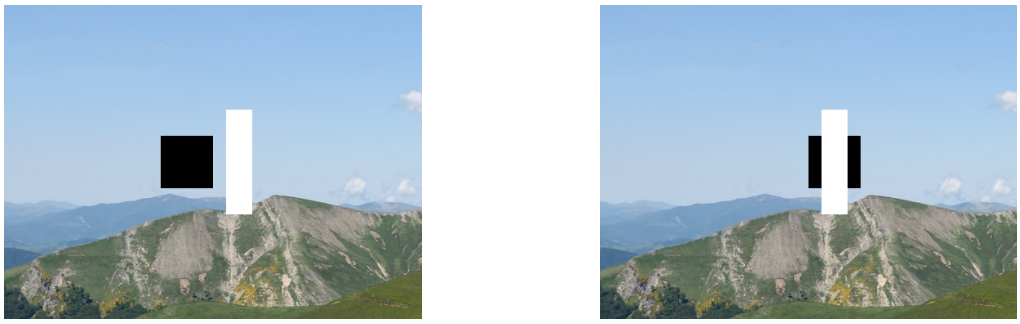


Figure 9.8: Synthetic test sequence used for the occlusion experiment. A background with a black square of size 64x64 is translated horizontally, symmetrically about the vertical axis. It is occluded in the second frame by a rectangle of increasing width. This image pair corresponds to a velocity magnitude of $\mathbf{v} = (64, 0)$ and an occlusion rectangle width of 32 pixels.

Figure 9.9: (left) The flow map of FlowNet2C for an occlusion rectangle width of 32 pixels, meaning half the square is occluded in the second frame. The model is only able to match the edges of the square in the two frames. (right) The flow map of SpyNet for an occlusion rectangle width of 64 pixels, meaning the square is completely occluded in the second frame. Here the model estimates the square disappears in the second frame.

10

Discussion of Preliminary Results

The preliminary evaluations have provided insight into the filters present (in the first layer) and have given an indication of the error characteristics of different CNN-based optical flow estimation methods. In this Chapter the results from these preliminary evaluations will be discussed.

In Section 9.1 the filters of FlowNet2S and SpyNet were visualized. It could be seen that some filters of SpyNet resemble Gabor filters. In Section 6.1 it was established that Gabor filters often emerge from learning-based approaches. This is due to the fact that they are optimally localized in space-time (Section 5.2). The filters of FlowNet2S on the other do not show such a clear temporal structure. Sun et al., 2018 showed that using a disruptive learning rate, the performance of FlowNetS can be vastly improved[1]. This is an indication that the performance of the FlowNetS architecture can still be vastly improved. The filters on all three networks appeared orientation sensitive. Sometimes deviating as much as 30% from the mean AEE of all orientations as can be seen in Figure 9.5. Even though the AEE for SpyNet is relatively small it is undesirable to have large deviations from the mean for different orientations.

Furthermore, it could be seen that the error outside of the square increased for the spatiotemporal filter-based models. This could be a consequence of the possible presence of Gabor filters which suffer from the uncertainty relation. Even though FlowNet2S and FlowNet2C have been trained on more data and have more parameters, for the simple case of image translation, SpyNet outperforms both these models up until velocities around 120 pixels per frame. This clearly shows the benefits of the pyramidal structure of SpyNet, which is different from the encoder-decoder architecture of FlowNet2S and FlowNet2C. The downside is that it does suffer most from the problem of large motion from small scale structures. Even though one of the main motivations for the U-net architecture in both FlowNet is to avoid this problem, both models still suffer from it. All models do have a receptive field size in their highest layer sufficiently high enough to deal with the aperture problem. This is not surprising, given the fact that in the training data, often the background or camera moves, and thus large regions are moving with the same velocity.

For more difficult optical flow problems, such as occlusion, something interesting happens. For the cases with highly occluded objects in the second frame both FlowNet2S and FlowNet2C 'guess' or 'reason' that the object must be behind the other object. SpyNet estimates that the square 'disappears' and shows an inverse color wheel. For the motion magnitude experiment, similar failure case could be seen. After a magnitude of 150 pixels, FlowNet2S estimates that the square moves outside of the frame to the left. The LDOF method, on the other hand, estimates a disappearing square for large motion magnitudes. Qualitatively, a case can be made for either failure mode. Quantitatively, the disappearing failure mode is better.

[1]The AEE on MPI-Sintel improved from 3.79 to 2.80 for FlowNetS.

IV

Appendices

A

Model Details

This appendix contains details about the FlowNetS and SpyNet models which are used in the prelimary evaluations. The model details of FlowNetS can be seen in Table A.1 and the details of SpyNet are given in Table A.2.

Name	Kernel	Stride	Padding	Ch I/O	In Res	Out Res	Input
conv1	7x7	2	3	6/64	512x384	256x192	Images
conv2	5x5	2	2	64/128	256x192	128x96	conv1
conv3	5x5	2	2	128/256	128x96	64x48	conv2
conv3_1	3x3	1	1	256/256	64x48	64x48	conv3
conv4	3x3	2	1	256/512	64x48	32x24	conv3_1
conv4_1	3x3	1	1	512/512	32x24	32x24	conv4
conv5	3x3	2	1	512/512	32x24	16x12	conv4_1
conv5_1	3x3	1	1	512/512	16x12	16x12	conv5
conv6	3x3	2	1	512/1024	16x12	8x6	conv5_1
conv6_1	3x3	1	1	1024/1024	8x6	8x6	conv6
flow6	3x3	1	1	1024/2	8x6	8x6	conv6_1
upconv5	4x4	2	1	1024/512	8x6	16x12	conv6_1
flow5	3x3	1	1	1026/2	16x12	16x12	upconv5+conv5_1+flow6
upconv4	4x4	2	1	1026/256	16x12	32x24	upconv5+conv5_1+flow6
flow4	3x3	1	1	770/2	32x24	32x24	upconv4+conv4_1+flow5
upconv3	4x4	2	1	770/128	32x24	64x48	upconv4+conv4_1+flow5
flow3	3x3	1	1	386/2	64x48	64x48	upconv3+conv3_1+flow4
upconv2	4x4	2	1	386/64	64x48	128x96	upconv3+conv3_1+flow4
flow2	3x3	1	1	192/2	128x96	128x96	upconv2+conv2+flow3

Table A.1: Model details of FlowNetS. The expansive part of the network starts at 'flow6'. Note the difference in the expansive part of the network is different than the dimensions provided in Dosovitskiy et al., 2015. Also note that even in Mayer et al., 2018 the dimensions are not correctly specified for the sizes of the upconvolutional kernels of FlowNet2C.

Name	Kernel	Stride	Padding	Ch I/O	In Res	Out Res	Input
convpyramid0	7x7	1	3	(8,32,64,32,16)/2	32x24	32x24	Images+flow0
convpyramid1	7x7	1	3	(8,32,64,32,16)/2	64x48	64x48	Images+convpyramid0
convpyramid2	7x7	1	3	(8,32,64,32,16)/2	128x96	128x96	Images+convpyramid1
convpyramid3	7x7	1	3	(8,32,64,32,16)/2	256x192	256x192	Images+convpyramid2
convpyramid4	7x7	1	3	(8,32,64,32,16)/2	512x384	512x384	Images+convpyramid3

Table A.2: Model details of SpyNet. 'flow0' refers to zero-valued initial flow map estimate. Between pyramid levels the flow estimate is bilinearly upsampled.

B

Flow Field Map

The flowfield color coding used throughout this thesis can be found in Figure B.1.

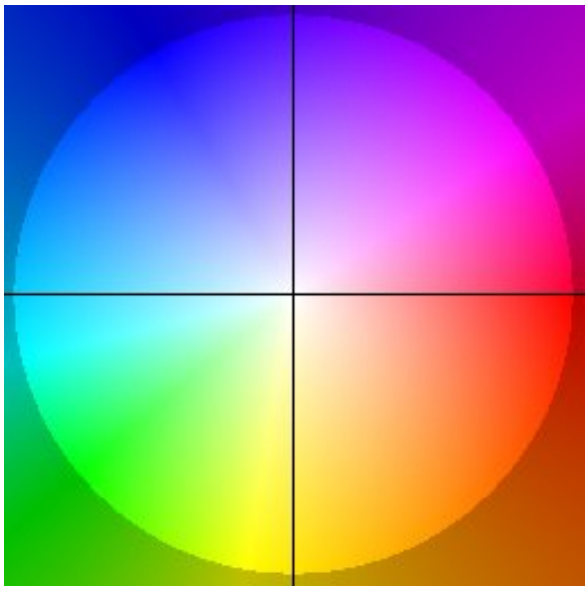


Figure B.1: Flow field color coding taken from Baker et al., 2011. Following the color coding rightward motion corresponds to a red color. Note that every flow field map in this thesis is normalized using their largest and lowest motion magnitudes.

Bibliography

Agrawal, O. P. (2002). Formulation of Euler-Lagrange equations for fractional variational problems. *Journal of Mathematical Analysis and Applications*, *272*(1), 368–379.

Anandan, P. (1989). A computational framework and an algorithm for the measurement of visual motion. *International Journal of Computer Vision*, *2*(3), 283–310.

Baker, S. & Matthews, I. (2004). Lucas-Kanade 20 Years On: A Unifying Framework. *International Journal of Computer Vision*, *56*(3), 221–255.

Baker, S., Scharstein, D., Lewis, J. P., Roth, S., Black, M. J. & Szeliski, R. (2011). A database and evaluation methodology for optical flow. *International Journal of Computer Vision*, *92*(1), 1–31.

Barron, J., Fleet, D. J. & Beachemin, S. (1994). Performance of Optical Flow Techniques. *International Journal of Computer Vision*.

Beauchemin, S. S. & Barron, J. L. (1995). The Computation of Optical Flow. *ACM Comput. Surv.* *27*(3), 433–466.

Beauchemin, S. & Barron, J. (2000). The frequency structure of one-dimensional occluding image signals. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *22*(2), 200–206.

Black, M., Yacoob, Y., Jepson, A. & Fleet, D. (1997). Learning parameterized models of image motion. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (pp. 561–567).

Black, M. J. & Anandan, P. [P.]. (1996). The robust estimation of multiple motions: Parametric and piecewise-smooth flow fields. *Computer Vision and Image Understanding*, *63*(1), 75–104.

Bracewell, R. (1986). *The Fourier transform and its applications* (Vol. 31999). New York: McGraw-Hill.

Brox, T., Bruhn, A., Papenberg, N. & Weickert, J. (2004). High Accuracy Optical Flow Estimation Based on a Theory for Warping. In *European conference on computer vision* (pp. 25–36).

Brox, T. & Malik, J. (2011). Large displacement optical flow: Descriptor matching in variational motion estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *33*(3), 500–513.

Bruhn, A., Weickert, J. & Schnörr, C. (2005). Lucas/Kanade Meets Horn/Schunck: Combining Local and Global Optic Flow Methods. *International Journal of Computer Vision*, *61*(3), 1–21.

Butler, D. J., Wulff, J., Stanley, G. B. & Black, M. J. (2012). A naturalistic open source movie for optical flow evaluation. In *European conference on computer vision* (pp. 611–625).

Camus, T. (1997). Real-time quantized optical flow. *Real-Time Imaging*, *3*(2), 71–86.

Dalal, N., Histograms, B. T. & Triggs, B. (2005). Histograms of Oriented Gradients for Human Detection. In *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)* (pp. 886–893).

de Croon, G., Ho, H. W., Wagter, C. D., van Kampen, E., Remes, B. & Chu, Q. P. (2013). Optic-Flow Based Slope Estimation for Autonomous Landing. *International Journal of Micro Air Vehicles*, *5*(4), 287–297.

Dosovitskiy, A., Fischery, P., Ilg, E., Hausser, P., Hazirbas, C., Golkov, V., … Brox, T. (2015). FlowNet: Learning optical flow with convolutional networks. *Proceedings of the IEEE International Conference on Computer Vision*, *2015 Inter*, 2758–2766.

Egan, K., Tseng, Y., Holzschuch, N., Durand, F. & Ramamoorthi, R. (2009). Frequency Analysis and Sheared Reconstruction for Rendering Motion Blur. *ACM Trans. Graph*, *28*(93).

Eigen, D., Puhrsch, C. & Fergus, R. (2014). Depth Map Prediction from a Single Image using a Multi-Scale Deep Network. In *Advances in neural information processing systems* (pp. 2366–2374).

Fleet, D. & Jepson, A. (1989). Computation of normal velocity from local phase information. In *Proceedings CVPR'89: IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (pp. 379–386).

Fleet, D. J., Black, M. J., Yacoob, Y. & Jepson, A. D. (2000). Design and use of linear models for image motion analysis. *International Journal of Computer Vision*, *36*(3), 171–193.

Gabor, D. (1945). Theory of communication. *Journal of the Institution of Electrical Engineers - Part I: General*, *94*(73), 58–58.

Gaidon, A., Wang, Q., Cabon, Y. & Vig, E. (2016). VirtualWorlds as Proxy for Multi-object Tracking Analysis. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 4340–4349).

Gautama, T. & Van Hulle, M. M. (2002). A phase-based approach to the estimation of the optical flow field using spatial filtering. *IEEE Transactions on Neural Networks*, *13*(5), 1127–1136.

Geiger, A. [A.], Lenz, P. & Urtasun, R. (2012). Are we ready for autonomous driving? The KITTI vision benchmark suite. In *2012 IEEE Conference on Computer Vision and Pattern Recognition* (pp. 3354–3361).

Gibson, J. J. (1950). *The perception of the visual world.* Oxford, England: Houghton Mifflin.

Güney, F. & Geiger, A. [Andreas]. (2017). Deep Discrete Flow. In *Computer Vision – ACCV 2016* (pp. 207–224).

Guzmán-rivera, A., Batra, D. & Kohli, P. (2012). Multiple Choice Learning: Learning to Produce Multiple Structured Outputs. In *Advances in neural information processing systems 25* (pp. 1799–1807). Curran Associates, Inc.

Harris, C. & Stephens, M. (1988). A combined corner and edge detector. Citeseer.

Heeger, D. J. (1987). Model for the extraction of image flow. *Journal of the Optical Society of America A*, *4*(8), 1455.

Heeger, D. J. (1988). Optical flow using spatiotemporal filters. *International Journal of Computer Vision*, *1*(4), 279–302.

Horn, B. K. & Schunck, B. G. (1981). Determining optical flow. *Artificial Intelligence*, *17*(1-3), 185–203.

Horn, B. (1986). *Robot vision*. MIT Press.

Hui, T. W., Tang, X. & Loy, C. C. (2018). LiteFlowNet: A Lightweight Convolutional Neural Network for Optical Flow Estimation. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 8981–8989.

Hui, T.-W., Tang, X. & Loy, C. C. (2019). A Lightweight Optical Flow CNN - Revisiting Data Fidelity and Regularization. *arXiv preprint arXiv:1903.07414*, 1–13.

Ilg, E., Mayer, N., Saikia, T., Keuper, M., Dosovitskiy, A. & Brox, T. (2017). FlowNet 2.0: Evolution of optical flow estimation with deep networks. *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, *2017-Janua*, 1647–1655.

Ilg, E., Ozgun, C., Galesso, S., Klein, A., Makansi, O., Hutter, F. & Brox, T. (2018). Uncertainty Estimates and Multi-Hypotheses Networks for Optical Flow. In *Proceedings of the European Conference on Computer Vision (ECCV)* (pp. 652–667).

Ilg, E., Saikia, T., Keuper, M. & Brox, T. (2018). Occlusions, Motion and Depth Boundaries with a Generic Network for Disparity, Optical Flow or Scene Flow Estimation. In *Proceedings of the European Conference on Computer Vision (ECCV)* (pp. 614–630).

Kondermann, D., Nair, R., Honauer, K., Krispin, K., Andrulis, J., Brock, A., … Jahne, B. (2016). The HCI Benchmark Suite: Stereo and Flow Ground Truth with Uncertainties for Urban Autonomous Driving. In *2016 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)* (pp. 19–28).

Konrad, J. (1999). Christoph Stiller Estimating Motion in Image Sequences. *IEEE Signal Processing Magazine*, 1–36.

Krizhevsky, A., Sutskever, I. & Hinton, G. (2012). Imagenet classification with deep convolutional neural networks. *papers.nips.cc*.

LeCun, Y. (1989). Generalization and network design strategies. In *Connectionism in perspective* (pp. 143–155).

LeCun, Y., Bengio, Y. & Hinton, G. (2015). Deep learning. *Nature*, *521*(7553), 436–444.

Lee, H.-C., Breneman, E. & Schulte, C. (1990). Modeling light reflection for computer color vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *12*(4), 402–409.

Liu, P., Lyu, M., King, I. & Xu, J. (2019). SelFlow: Self-Supervised Learning of Optical Flow. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 4571–4580).

Longuet-Higgins, H. C. & Prazdny, K. (1980). The interpretation of a moving retinal image. *Proceedings of the Royal Society of London. Series B. Biological Sciences*, *208*(1173), 385–397.

Lucas, B. D. & Kanade, T. (1981). An Iterative Image Registration Technique with an Application to Stereo Vision.

Mac Aodha, O., Brostow, G. J. & Pollefeys, M. (2010). Segmenting video into classes of algorithm-suitability. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (pp. 1054–1061).

Mayer, N., Ilg, E., Fischer, P., Hazirbas, C., Cremers, D., Dosovitskiy, A., … Brox, T. (2018). What Makes Good Synthetic Training Data for Learning Disparity and Optical Flow Estimation? *International Journal of Computer Vision*, *126*, 942–960.

Mayer, N., Ilg, E., Hausser, P., Fischer, P., Cremers, D., Dosovitskiy, A. & Brox, T. (2016). A Large Dataset to Train Convolutional Networks for Disparity, Optical Flow, and Scene Flow Estimation. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (Vol. 2016-Decem, pp. 4040–4048).

McCormac, J., Handa, A., Leutenegger, S. & Davison, A. J. (2017). SceneNet RGB-D: Can 5M Synthetic Images Beat Generic ImageNet Pre-training on Indoor Segmentation? In *2017 IEEE International Conference on Computer Vision (ICCV)* (pp. 2697–2706).

Menze, M. & Geiger, A. [Andreas]. (2015). Object scene flow for autonomous vehicles. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (Vol. 07-12-June, pp. 3061–3070).

Mileva, Y., Bruhn, A. & Weickert, J. (2007). Illumination-Robust Variational Optical Flow with Photometric Invariants. *Pattern Recognition*, 152–162.

Mulder, J. A., van der Vaart, J. C., van Staveren, W. H. J. J., Chu, Q. P. & Mulder, M. (2016). *Aircraft Responses to Atmospheric Turbulence*.

Nagel, H.-H. & Enkelmann, W. (1986). An Investigation of Smoothness Constraints for the Estimation of Displacement Vector Fields from Image Sequences. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *PAMI-8*(5), 565–593.

Neoral, M., Šochman, J. & Matas, J. (2019). Continual Occlusion and Optical Flow Estimation. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* (Vol. 11364 LNCS, pp. 159–174).

Otte, M. & Nagel, H. .-.-H. (1994). Optical flow estimation: Advances and comparisons. In *Computer Vision — ECCV '94* (pp. 49–60).

Ranjan, A. & Black, M. J. (2017). Optical flow estimation using a spatial pyramid network. *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, *2017-Janua*, 2720–2729.

Razavian, A. S., Azizpour, H., Sullivan, J. & Carlsson, S. (2014). CNN features off-the-shelf: An astounding baseline for recognition. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, 512–519.

Revaud, J., Weinzaepfel, P., Harchaoui, Z. & Schmid, C. (2015). EpicFlow: Edge-preserving interpolation of correspondences for optical flow. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, *07-12-June*, 1164–1172.

Ronneberger, O., Fischer, P. & Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention* (Vol. 9351, pp. 234–241).

Roth, S., Black, M. J., Roth, S. & Black, M. J. (2009). Fields of Experts. *Int J Comput Vis*, *82*, 205–229.

Rupprecht, C., Laina, I., Dipietro, R., Baust, M., Tombari, F., Navab, N. & Hager, G. D. (2017). Learning in an Uncertain World: Representing Ambiguity Through Multiple Hypotheses. In *Proceedings of the IEEE International Conference on Computer Vision* (Vol. 2017-Octob, pp. 3611–3620).

Seitz, S. M. & Baker, S. (2009). Filter flow. In *2009 IEEE 12th International Conference on Computer Vision* (pp. 143–150).

Shafer, S. A. (1985). Using color to separate reflection components. *Color Research & Application*, *10*(4), 210–218.

Shannon, C. E. (1948). A Mathematical Theory of Communication. *Bell System Technical Journal*, *27*(3), 379–423.

Simoncelli, E., Adelson, E. & Heeger, D. (1991). Probability distributions of optical flow. In *Proceedings. 1991 IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (pp. 310–315).

Singh, A. (1991). *Optic Flow Computation: A Unified Perspective*. IEEE Computer Society Press Los Alamitos.

Springenberg, J. T., Dosovitskiy, A., Brox, T. & Riedmiller, M. (2014). Striving for Simplicity: The All Convolutional Net. *arXiv preprint arXiv:1412.6806*.

Srivastava, N., Hinton, G., Krizhevsky, A. & Salakhutdinov, R. (2014). Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research*, *15*(1), 1929–1958.

Sun, D., Roth, S., Lewis, J. P. & Black, M. J. (2008). Learning optical flow. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* (Vol. 5304 LNCS, *PART 3*, pp. 83–97).

Sun, D., Yang, X., Liu, M.-Y. & Kautz, J. (2018). Models Matter, So Does Training: An Empirical Study of CNNs for Optical Flow Estimation. *arXiv preprint arXiv:1809.05571*, 1–15.

Sze, V., Chen, Y. H., Yang, T. J. & Emer, J. S. (2017). Efficient Processing of Deep Neural Networks: A Tutorial and Survey. *Proceedings of the IEEE*, *105*(12), 2295–2329.

Teney, D. & Hebert, M. (2016). Learning to extract motion from videos in convolutional neural networks. In *Asian Conference on Computer Vision* (pp. 412–428). Springer, Cham.

Ullman, S. (1979). *The interpretation of visual motion*. MIT Press.

Uras, S., Girosi, F., Verri, A. & Torre, V. (1988). A computational approach to motion perception. *Biological Cybernetics*, *60*(2), 79–87.

van de Weijer, J. & Beigpour, S. (2011). The Dichromatic Reflection Model-Future Research Directions and Applications. *Int. Joint Conf. on Computer Vision, Imaging and Computer Graphics Theory and Applications*, (January), 11.

Van Hateren, J. H. & Ruderman, D. L. (1998). Independent component analysis of natural image sequences yields spatio-temporal filters similar to simple cells in primary visual cortex. *Proceedings of the Royal Society B: Biological Sciences*, *265*(1412), 2315–2320.

Weinzaepfel, P., Revaud, J., Harchaoui, Z. & Schmid, C. (2013). DeepFlow: Large displacement optical flow with deep matching. *Proceedings of the IEEE International Conference on Computer Vision*, (Section 2), 1385–1392.

Werlberger, M., Trobin, W., Pock, T., Wedel, A., Cremers, D. & Bischof, H. (2009). Anisotropic Huber-L 1 Optical Flow. *BMVC*, *1*, 3.

Wulff, J. & Black, M. J. (2015). Efficient sparse-to-dense optical flow estimation using a learned basis and layers. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 120–130).

Yu, F. & Koltun, V. (2015). Multi-Scale Context Aggregation by Dilated Convolutions. *arXiv preprint arXiv:1511.07122*.

Zimmer, H., Bruhn, A. & Weickert, J. (2011). Optic flow in harmony. *International Journal of Computer Vision*, *93*(3), 368–388.

Zweig, S. & Wolf, L. (2017). InterpoNet, a brain inspired neural network for optical flow dense interpolation. In *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017* (Vol. 2017-Janua, pp. 6363–6372).