Institute for Visualization and Interactive Systems

University of Stuttgart
Universitätsstraße 38
D–70569 Stuttgart

Master's Thesis

An Extended Analysis of Difficulties and Regularities in Optical Flow Benchmarks

Stephan Albrecht

Course of Study:	Computer Science
Examiner:	Prof. Dr. -Ing. Andrés Bruhn
Supervisor:	Daniel Maurer, M.Sc.
Commenced:	May 15, 2017
Completed:	November 15, 2017
CR-Classification:	G.1.6, G.1.8, I.4.8

Abstract

A central problem in the field of computer vision is the extraction of movement from a sequence of images. This includes the determination of the displacement vector field between two subsequent frames. In the context of computer vision this displacement field is referred to as the optical flow. Until today many algorithms have been developed to solve the optical flow problem. In addition researchers have developed various kinds of benchmarks enabling the evaluation on performance and quality of these algorithms. The benchmarks contain real world data (KITTI [GLU12, MG15]), simple synthetic and real data (Middlebury [BSL$^+$11]) and even rendered movies with different rendering modes (MPI Sintel [BWSB12]). However, these benchmarks do not only differ in their creation, but also in the complexity of the scenes. The reason is the different focus on different challenges of the optical flow problem. Even though these benchmarks provide a good environment for comparison, only few studies provide an overall analysis on the difficulties and regularities of these optical flow test suites. These difficulties and regularities include illumination changes, large displacements and different types of movement. One of the few works addressing such an analysis is the master thesis of Hager [Hag17]. Hager analyzed the KITTI 2015, KITTI 2012, Middlebury and MPI Sintel benchmark on the aforementioned difficulties and regularities.

This thesis extends the work of Hager by presenting refined, as well as different methods and metrics to increase the interpretability of the obtained results in the different fields. Additionally, it provides a measure for researchers to help them to find image sequences containing a certain difficulty. A variational approach, based on brightness transfer functions, is introduced to measure illumination changes. The large displacement analysis is extended by a scale analysis in order to find large displacements of small objects. The movement type analysis is done using the order adaptive approach of Maurer et al. [MSB17]. The introduced metrics are tested on the training data of the benchmarks, with ground truth and computed flow, and compared to the results of Hager.

Kurzfassung

Ein zentrales Problem des Maschinensehens ist die Extraktion von Bewegungen aus einer Bildsequenz. Dies beinhaltet die Bestimmung des Verschiebungsvektorfeldes zwischen zwei aufeinander folgenden Bildern. Dieses Verschiebungsvektorfeld wird auch als Optischer Fluss bezeichnet. Bis heute wurden schon viele Algorithmen zur Lösung des Optischen Fluss Problems entwickelt. Gleichzeitig konzipierten Forscher unterschiedliche Benchmarks, gegen die die Algorithmen auf Performanz und Qualität evaluiert werden können. Die Benchmarks umfassen Daten aus der realen Welt (KITTI [GLU12, MG15]), einfache synthetische und reale Daten (Middlebury [BSL$^+$11]) und sogar gerenderte Filme mit verschiedenen Rendermodi (MPI Sintel [BWSB12]). Diese Benchmarks unterscheiden sich nicht nur in ihrer Erstellung, sondern auch in der Komplexität der Szenen. Der Grund dafür ist der Fokus auf unterschiedliche Schwierigkeiten des Optischen Fluss Problems. Obwohl diese Benchmarks sich sehr gut für den Vergleich diverser Algorithmen eignen, beschäftigen sich nur wenige Arbeiten mit einer kompletten Analyse der Optischen Fluss Benchmarks bezüglich Schwierigkeiten und Regularitäten. Diese Schwierigkeiten und Regularitäten beinhalten Beleuchtungsänderungen, große Bewegungen von kleinen Objekten und unterschiedliche Bewegungstypen. Eine dieser wenigen Arbeiten ist die Masterarbeit von Hager [Hag17]. Es werden die KITTI 2015, KITTI 2012, Middlebury und MPI Sintel Benchmark auf die zuvor erwähnten Schwierigkeiten und Regularitäten untersucht.

Diese Thesis erweitert die Arbeit von Hager mit überarbeiteten oder gar anderen Methoden und Metriken, um die Interpretierbarkeit der Resultate zu erhöhen. Es soll Forschern ein Messinstrument zur Verfügung gestellt werden, um ihnen bei der Suche nach bestimmten Schwierigkeiten in Testdatensätzen zu helfen. Ein überarbeiteter variationaler Ansatz, basierend auf Helligkeitstransferfunktionen, zur Messung von Beleuchtungsänderungen wird vorgestellt. Die Analyse von großen Verschiebungen wird mit einer Skalenanalyse erweitert, um große Versetzungen von kleinen Objekten zu detektieren und zu lokalisieren. Die Analyse von Bewegungstypen in einem Flussfeld wird mit dem ordnungsadaptiven Ansatz von Maurer et al. [MSB17] durchgeführt. Die vorgestellten Metriken werden mit Trainingsdaten der Benchmarks, mit gegebenem und berechnetem Fluss, evaluiert und mit den Ergebnissen von Hager verglichen.

Contents

1 Introduction

The desire to capture scenes on a photo dates all the way back to the 18th century [KS13]. Since then the interest in images and videos has become more important. Today mankind does not only use images or videos for their amusement, but also in areas of health care [YLL$^+$08], security [AMN13, BLLJ09] and transportation [GCT98]. A significant problem, which arises, is to find correspondences between image sequences, e.g. to locate and track objects. In terms of computer vision these correspondences between one image and another, are referred to as displacement fields or optical flow. Many methods ranging from early simple methods, like phase correlation and block based approaches, over to more complex approaches using differential methods, have been invented to solve the optical flow problem. Throughout the years, many optical flow benchmarks were introduced in order to test the quality of the developed methods.

1.1 Motivation

While these benchmarks provide a good base for comparison for the invented algorithms, only few efforts have been made to identify the difficulties and regularities of these benchmarks. Most of the time the different attributes of the methods, e.g. robustness w.r.t. illumination changes, are compared by just using a few image sequences or generated images with a specific difficulty [BFB94, MLBV10, GMN$^+$98]. A small number of works analyzing the differences of optical flow benchmarks, mostly focus on synthetic or real-world aspects [VRKM08, WBSB12, BSL$^+$11]. Vaudrey et al. [VRKM08], for example focus on how good the test benches are suited for the task of driver assistant systems. The significant difference of driver assistant systems to other tasks is, that robustness is more vital than accuracy. Till today's extend a large number of test benches, synthetic or real-world, focusing on accuracy have been created and made public. This strives not only for an analysis of the differences between these benchmarks, but also in general, to find out what the difficulties and regularities of optical flow benchmarks are.

Therefore, the goal is to create metrics to measure the difficulties and regularities. These metrics can be used to measure the complexity of benchmarks, or as parameters for methods determining the optical flow. Especially the differential methods proved to be very promising, having a great potential for optimization through assumption modeling. In general it applies, that the assumptions made about a problem, steer the complexity of solving it. In the case of optical flow these assumptions can be for example color constancy, gradient constancy or piece wise smooth flow fields. The challenge one faces, is to choose the right assumptions for a given image sequence. A model containing the color constancy assumption would perform poorly, if the scene contains a lot of illumination changes from one frame to the other. In that case, the gradient constancy assumption would be the better choice. One can only choose the right assumptions for solving the optical flow problem, if the information needed can be extracted from the images in advance, for example through the metrics introduced in this thesis. It is desirable to gain insight to the information in order to develop models adapting themselves automatically to the complexity of the problem. The motto is, don't use a sledgehammer to crack a nut. Another benefit is that the developed metrics can also help researchers find images sequences containing a certain difficulty, which they are looking for. The efforts, which have already been made towards developing these metrics to measure the difficulties and regularities, are presented in the next section.

1.2 Related Work

As mentioned before a lot of works analyze the differences of optical flow benchmarks mostly focusing on synthetic or real-world aspects or if the benchmarks contain certain complex tasks or scenarios. Most of the time, the works introducing a new benchmark discuss some difficulties, which the designed benchmark focuses on, like in the paper of Baker et al. [BSL$^+$11] introducing the Middlebury benchmark. A small comparison between the Middlebury and MPI Sintel benchmark can be found in the paper of Butler et al. [BWSB12]. Here the authors point out, that the MPI Sintel test suite contains long sequences, specular reflections, large motions, motion blur, defocus blur, and atmospheric effects, which are not covered by the Middlebury benchmark. In contrast, the Middlebury test suite contains sequences with nonrigid motion, high frame-rate videos, realistic synthetic sequences and modified stereo sequences of static scenes.

One of the few works discussing this issue of complications within different Benchmarks is the thesis "Difficulties and Regularities of Optical Flow Benchmarks" by Hager [Hag17]. The test suites taken into consideration are the KITTI 2012 and 2015, the Middlebury and MPI-Sintel test suite [GLU12, MG15, BSL$^+$11, BWSB12]. Different metrics enabling the measurement of difficulties and regularities, regarding illumination changes, different types of movement and large displacements, are introduced. Variational methods are used to approximate global and local illumination components to derive a metric, to measure the amount of illumination changes within a given image sequence. Additionally, another variational approach was used to analyze the flow fields, to find out how much of a certain movement type is contained in an image sequence. The last topic covered is the one of large displacements. Hager uses a certain threshold to distinguish between small and large displacements, which will be explained later. Finally, these components are analyzed, to obtain an overview of the difficulties and regularities of the aforementioned benchmarks.

1.3 Structure

The thesis is split into a foundations part, Chapter 2, and three main parts addressing the difficulties and regularities of optical flow benchmarks, namely: illumination changes, large displacements and different types of movement.

The first part of the analysis starting in Chapter 3 focuses on the difficulty of local illumination changes within a given image sequence. The results of the variational method used in [Hag17], when applied on identical images are discussed in Section 3.1. This will lead to a new modified version of the method presented in Section 3.2. Based on the new results, a slightly changed standard deviation with a fixed mean is proposed in Section 3.3, as an additional metric to those of [Hag17]. Section 3.4 concludes the first part of the evaluation of the new metric on the KITTI 2012/2015, Middlebury and MPI Sintel benchmark with both ground truth and computed flow fields.

The second part, Chapter 4, focuses on large displacements. The first section covers three approaches, which are used to find the scale of objects within a scene: a difference of Gaussians, a patch and a marching approach. Section 4.2 describes, how the large displacements are measured, and Section 4.3 presents the evaluation of the introduced metric.

An analysis of the different types of movements within an image sequence is done in Chapter 5, based on the order adaptive variational approach of Maurer et al. [MSB17]. The main concepts of the order adaptive approach, as well as the ones of Hagers approach are recapitulated in the Sections 5.1 and 5.2. The results of the order adaptive approach are presented and compared to the results of Hager in Section 5.3.

Chapter 6 concludes the thesis with a summary of the obtained results and future work.

2 Foundations

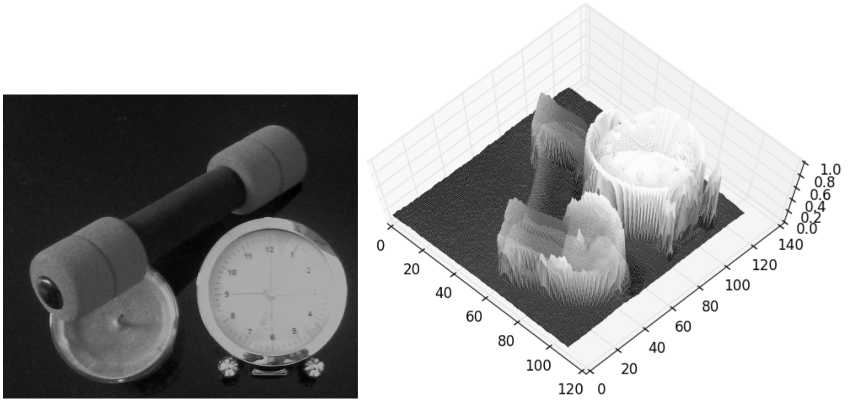
Let us start by introducing the main concepts used in this thesis. As mentioned before, this thesis is an extended analysis of difficulties and regularities of optical flow benchmarks. To be able to perform a detailed analysis, we will begin by taking a closer look at images and the optical flow problem in the first section. The next two sections introduce the considered benchmarks as well as the difficulties and regularities of optical flow benchmarks. The following section briefly summarizes the concepts of brightness transfer functions, which are used later on, to measure the difficulty of illumination changes. The next section of this chapter reviews differences of Gaussians and how they can be applied to obtain a scale-space representation for the later analysis of large displacements. Afterwards a short introduction of the EpicFlow method is given, which is later on used to interpolate sparse ground truth flow fields, in the chapter on movement categorization.

2.1 From Images to the Optic Flow

2.1.1 Images

The most common representations of an image are gray-scale or RGB-color images. Alternatively, an image can be represented as a function f, mapping positions of a rectangular domain $\Omega = (0, n) \times (0, m)$ to a co-domain of color $\mathbb{R}^3$ or gray $\mathbb{R}$ values. Ω is also called image domain or image plane [JHG99]. This can be written as $f : \mathbb{R}^2 \supset \Omega \to \mathbb{R}$ or $\mathbb{R}^3$ respectively. Figure 2.1 depicts an example of a gray-scale image with the corresponding representation as a function.

Dark positions correspond to low values of the function and bright ones to high values. For image sequences, the image $f(x, y, t)$, denoted as f^t, does not only depend on the positions $(x, y) \in \Omega$, but also on the time step t, with $t \in \mathbb{N}$. For RGB-color images the function $f : \Omega \to \mathbb{R}^3$ maps to a three-dimensional vector instead of a scalar value. As we now know how to represent images as a continuous function, let us move on to the optical flow problem.



(a) **(b)**

Figure 2.1: Example of an image representation as a 3D visualization. *From left to right:* (a) Gray-scale image. (b) Representation of the down-sampled version of (a) as a function $f(x, y)$ over a rectangular image domain Ω.

2.1.2 The Optical Flow Problem

A general problem in computer vision is to find corresponding pixels between two images. These correspondences are then used to track objects, make predictions about the movement or to make mappings between different types of images. The correspondences between images or frames are contained in the so-called displacement field or optical flow [HS81]. The calculation of the displacement field is referred to as the optical flow problem.

Let us take a closer look at the following example depicted in Figure 2.2. A car passes by a camera standing next to the road. The camera takes a picture at time step t and $t+1$. In the first frame the car is located at the position $(x, y)^\top$. In the second frame, the car has moved to a new position, denoted as $(x + u(x, y), y + v(x, y))^\top$.

The goal is to calculate the displacement vectors $(u(x, y), v(x, y))^\top$, short $(u, v)^\top$, for each position $(x, y)^\top \in \Omega$. To solve this problem, many methods have been developed [BBPW04, Hee87, LK$^+$81, AB85, BGW91, HS81]. One of the most famous approaches is the one of Horn and Schunck [HS81], using the idea of minimizing a cost function to obtain the desired flow field. Since the introduced concepts in this thesis are also based on the minimization of a cost function, let us go more into detail.

2.1.3 Variational Optical Flow Framework

The aforementioned cost functions are also called energy functionals. The difference between a function and a functional is that a function maps one or more input values to one value, whereas a functional maps one or more functions to one value. The functionals used in this

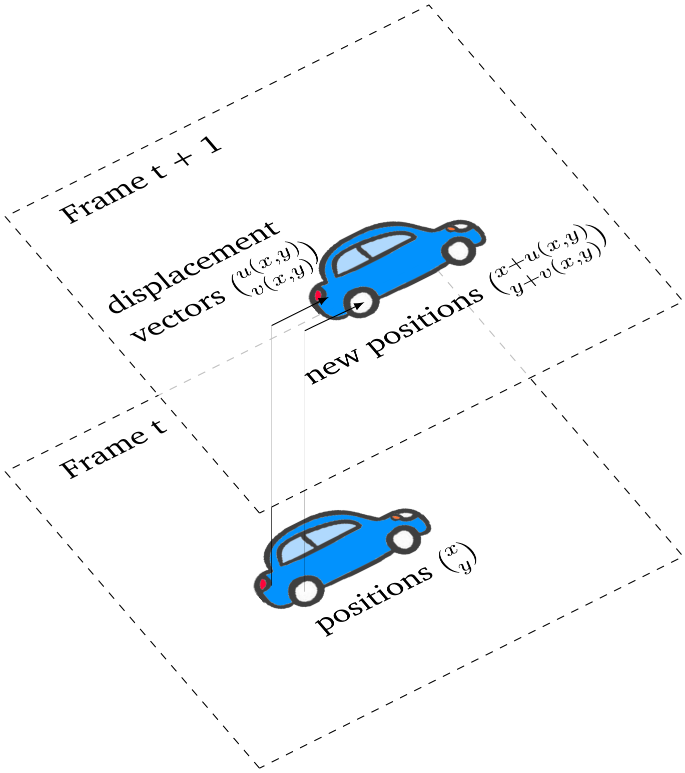

Frame t + 1
displacement vectors $\binom{u(x,y)}{v(x,y)}$
new positions $\binom{x+u(x,y)}{y+v(x,y)}$
Frame t
positions $\binom{x}{y}$

Figure 2.2: An example of a car, driving from left to right, captured in two frames with two exemplary displacement vectors.

thesis are denoted as E. The most well known functional is the integral, which builds the base for the variational optic flow framework:

$$E(u,v) = \int_\Omega F(x,y,u,v,u_x,u_y,v_x,v_y) \;\; dxdy \;\; . \tag{2.1}$$

The idea is, that the better the flow field u, v is approximated, the lower are the costs or energy. u_x, u_y, v_x and v_y denote the partial derivatives in x- and y-direction of u and v. To minimize a functional as in equation 2.1, one makes use of the Euler-Lagrange equations reading:

$$\begin{aligned} F_u - \partial_x F_{u_x} - \partial_y F_{u_y} &= 0 \;\; , \\ F_v - \partial_x F_{v_x} - \partial_y F_{v_y} &= 0 \;\; , \end{aligned} \tag{2.2}$$

with the Neumann boundary conditions

$$n^\top \begin{pmatrix} F_{u_x} \\ F_{u_y} \end{pmatrix} = 0, n^\top \begin{pmatrix} F_{v_x} \\ F_{v_y} \end{pmatrix} = 0 \;\; . \tag{2.3}$$

The vector n is the vector orthogonal to the image boundary.

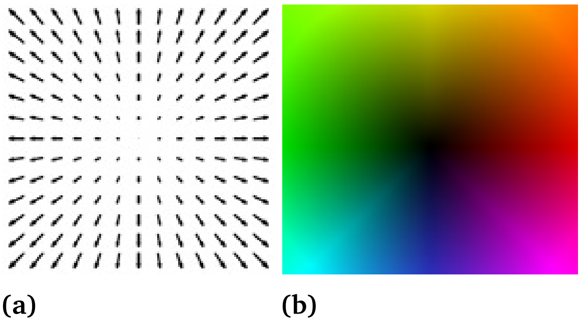


Figure 2.3: Example visualizations of flow fields. *From left to right:* (a) Sampled flow field. (b) Color coded flow field.

Typically F is composed of a data term D and a regularization term R, weighted by a factor α. This factor steers how smooth the resulting flow field is.

$$E(u, v) = \int_{\Omega} D(u, v) + \alpha \cdot R(u, v) \;\; dxdy \;\; . \tag{2.4}$$

The data term is used to make assumptions about the constancy of certain image features e.g the brightness constancy assumption. In contrast the regularization term makes assumptions regarding the flow to counter obscure solutions of the data term e.g. first order smoothness assumption. The first order smoothness assumption states, that the computed flow should be smooth. Mathematically this is expressed by the sum of gradients of u and v being close to zero. Higher orders of movement can be described by affine parametrizations of the flow, e.g. making it linear instead of constant.

A more detailed explanation of how optical flow fields can be computed can be found in [ZBW11]. The aforementioned assumptions need to be made, since one comes across different difficulties, when computing the optical flow. For example the aperture problem [BB95] or the occlusion of moving objects is countered by the smoothness assumption of the flow.

How other difficulties, like illumination changes or different movement types are countered, is explained in the later Section 2.3. Once the flow is computed, it can be displayed in several ways. The most common are sampled vector fields or color coded images as shown in Figure 2.3.

2.2 Review of the used Benchmarks

Like in the previous work of Hager, the following four benchmarks will be evaluated: the KITTI 2012 and 2015, the Middlebury and the MPI Sintel test suite. Let us review the specific attributes of each benchmark beginning with the KITTI benchmarks.

2.2.1 The KITTI 2012 and 2015 Benchmarks

The KITTI 2012 and 2015 test suites [GLU12, MG15] were created by Geiger et al. in the German city of Karlsruhe. Geiger and his team used the autonomous driving platform Annieway, equipped with a 360 degree Velodyne laser-scanner, as well as a stereo camera rig and GPS to establish these challenging real world benchmarks. The captured scenes vary from rural areas to highways with up to 15 cars and 30 pedestrians per image. The total amount of training frames are 194 in the 2012 set, and 200 in the 2015 set. The main difference between the 2012 and 2015 test suite is, that the 2012 benchmark does not contain scenes where other traffic participants are moving. In contrast, the 2015 sequences contain such sequences. For more accurate data the other moving vehicles are replaced by models. The sparse ground truth flow obtained by the laser scanner is on the one hand accurate for real world data, but on the other hand, the sparsity will cause some problems, when calculating the size of objects. Clearly these benchmarks comprise most of the difficulties, like illumination changes, object sizes and movement types, which one will come across, when developing optical flow techniques for autonomous driving.

2.2.2 The Middlebury Benchmark

The second benchmark used, is the Middlebury benchmark from Baker et al. [BSL$^+$11]. It is designed to cover four different aspects: non-rigid motion, realistic synthetic sequences, interpolation errors and modified stereo sequences of static scenes. The ground truth for non rigid motion is determined by tracking fluorescent textures. All of the eight scenes, used for the analysis are the ones, where training data is available.

In contrast to the KITTI benchmarks, the ground truth flow fields are more dense, but the amount of movement from frame to frame is not as large as in most of the KITTI sequences. Additionally the non-rigid scenes of the Middlebury benchmark were taken indoors on a movable setup. The purpose of this benchmark is different in many ways from the one of KITTI, which makes it an interesting candidate for a comparison.

2.2.3 The MPI Sintel Benchmark

The last test suite used in the later analysis, is the MPI Sintel from Butler et al. [BWSB12]. It is based on the synthetic short film Sintel and focuses on long-range motion, motion blur, multi-frame analysis and non-rigid motion. Additional to the aforementioned difficulties, the MPI Sintel test suite also addresses atmospheric challenges in different levels of rendering. The dense ground truth flow fields, together with a large amounts of training data of 1041 frames, from the naturalistic video sequences, form a challenging benchmark.

2.3 Difficulties and Regularities

The different benchmarks address different challenges. These challenges can be grouped into two main categories, difficulties and regularities.

The difficulties of optical flow benchmarks describe challenges, which intensify the task of finding the correspondences, e.g.: local-illumination changes like shadows or specular reflections, large displacements of small objects and others. Researchers have come up with many different models for the data and regularisation term of the variational optic flow framework to tackle these challenges. For example, if an image sequence contains a lot of illumination changes, one switches from the gray value constancy assumption to the gradient constancy assumption to gain better results.

Other problems, like the presence of local affine movement, can be handled by a different parametrisation of the flow [S^+94, NBK08]. This type of movement can be found in sequences taken by a car driving through the streets, as in the KITTI benchmarks. It may also be useful to switch between assumptions during the computation, to obtain the best motion model for the current frame or even the current pixel.

On the opposite regularities define properties of a task, which ease the solving process because additional assumptions can be made for simplification. In the case of optical flow determination, these could be global illumination/atmospheric changes, small displacements and global/camera movement, which may be constant or affine. Motion models containing global assumptions are easier to solve and therefore use less computational resources.

Some examples of the aforementioned difficulties and regularities are depicted in Figure 2.4. The first row depicts the regularity of global illumination changes, where the brightness changes from one frame to the other. An example for large displacements of small objects is found in row two, showing frame sections of the MPI Sintel test suite. The last two rows depict constant movement, a white car driving from left to right, and affine movement, a black car driving on the left. In addition, local illumination changes are present in these scenes, for example the shadows on the cars.

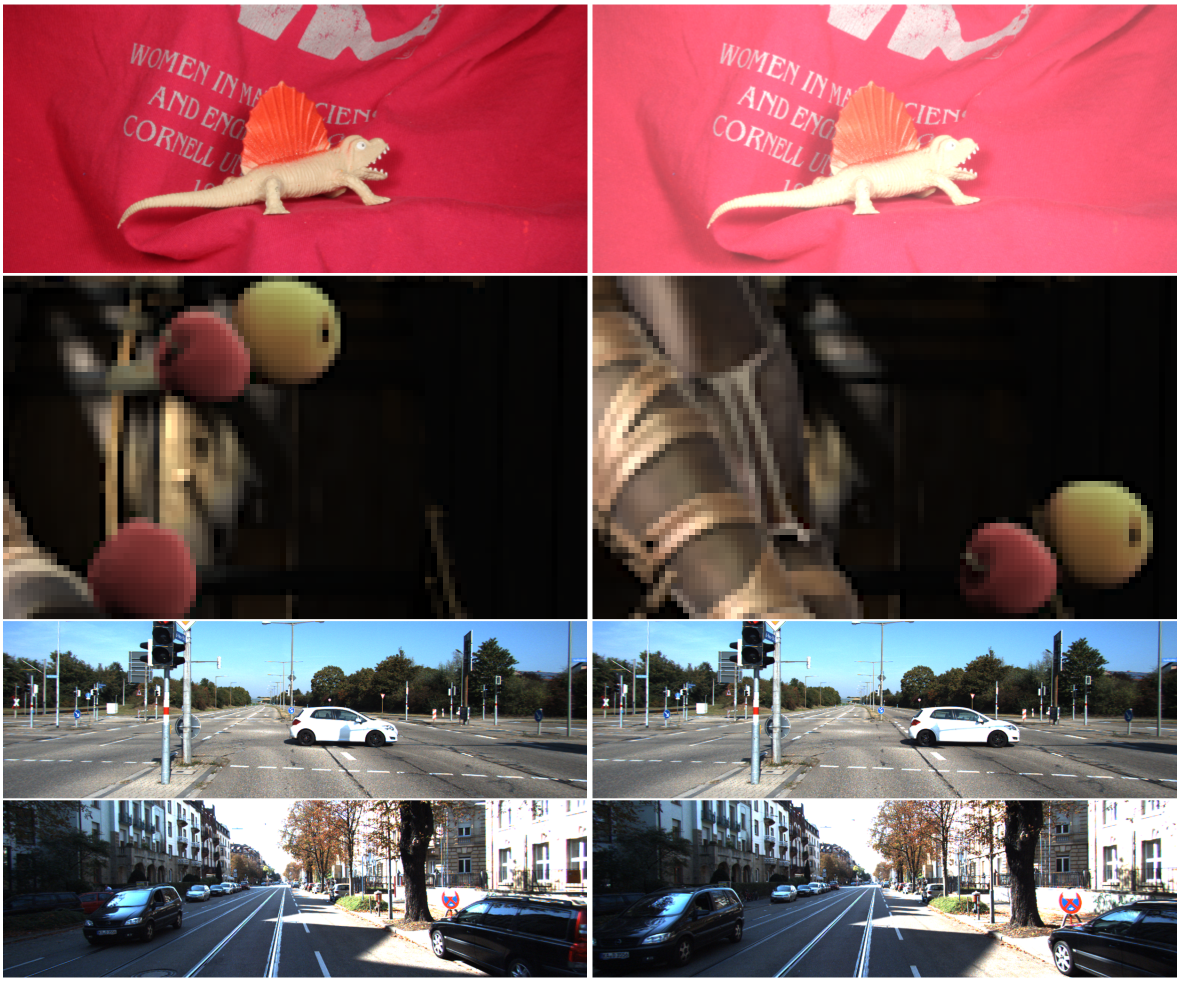


Figure 2.4: Examples of difficulties and regularities of optical flow benchmarks. *First row:* Global illumination changes within the same scene (Image section of an image of the Middlebury benchmark). *Second row:* Large displacements in MPI Sintel test suite. *Third and fourth row:* Images of the KITTI 2015 benchmark. *Third row:* Constant movement. *Fourth row:* Affine movement.

2.4 The Concept of Brightness Transfer Functions

This section briefly covers the basic concept of the brightness transfer functions used in this thesis. For further details see [GN06]. Originally brightness transfer functions have been used to describe the illumination change for two images of the same scene with different exposure times. Figure 2.5 shows exemplary the role of the brightness transfer functions.

These functions are based on the radiometric response function, which relates the actual measured brightness at a sensor to the image plane irradiance [LZ05]. To extract the radiometric response function out of a set of images, several constraints, in form of pixel pair brightness values, need to be found. To do so the following equation is set up:

$$g(f^t) = kg(f^{t+1}) \ , \tag{2.5}$$

with g being the inverse of the radiometric response function and k the exposure ratio $\frac{\text{expo\_}f^{t+1}}{\text{expo\_}f^t}$. All methods, which work without color rendition charts, use this equation to recover g and k. Each pair of corresponding pixels in an image pair gives one constraint. In order to obtain g and k, the iterative method of Mitsunaga and Nayar[MN99] is used. Since the focus is mapping the brightness values and not the irradiance values, we may reformulate the Equation 2.5 to

$$f^{t+1} = g^{-1}(kg(f^t)) = \Phi(f^t) \ , \tag{2.6}$$

with Φ being called brightness transfer function. How to obtain an explicit form of Φ will be shown later in Chapter 3.

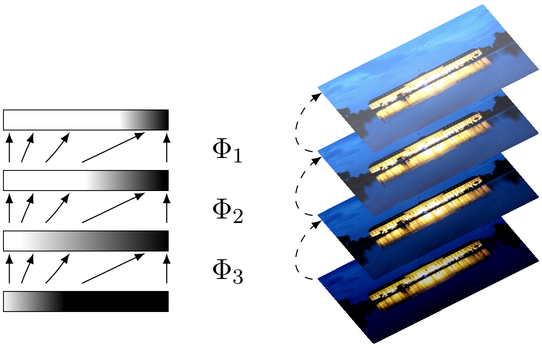


Figure 2.5: The role of the brightness transfer functions, showing the corresponding brightness from one image to the brightness of another image.

2.5 Difference of Gaussians and Their Applications

For the analysis of large displacements, it is necessary to detect the size of objects in the scene. Algorithms such as SIFT and SURF [Low99, BETVG08], are able to calculate the scale of a feature point by using a difference of Gaussians-pyramid. This technique is used for the evaluation of large displacements. Therefore a brief introduction to differences of Gaussians and their applications is given in this section.

2.5.1 The Gauss-Function

One of the most important functions in signal processing is the Gauss-function. It can be used in various ways, e.g. to smooth a signal or image or build a band-pass filter. The band-pass filter, created using the difference of Gaussians, will be used later on. The Gauss-function is a composition of the exponential function and a concave quadratic function. The Gauss-function is adapted to scale space and reads:

$$G(x,y,\sigma) = \frac{1}{2\pi\sigma^2}\mathrm{e}^{-\frac{(x-\mu)^2+(y-\mu)^2}{2\sigma^2}} \quad , \tag{2.7}$$

where x and y correspond to the image coordinates and σ to the standard deviation of the Gaussian with mean μ. An image can then be smoothed by convolving it with the Gaussian. For further information on convolution please see [S$^+$97].

2.5.2 Difference of Gaussians-Pyramids

When subtracting two Gaussians with different standard deviations as shown below,

$$D(x,y,\sigma) = (G(x,y,k\sigma) - G(x,y,\sigma)) * f(x,y) \quad , \tag{2.8}$$

one obtains a band-pass filter, called difference of Guassians (DoG). f denotes the original image, $*$ the convolution operator, and k a constant factor, with $k > 0$. The DoG is an approximation to the scale normalized Laplacian of Gaussians, as proved by Lindeberg et al. [Lin94]. When the DoG is applied to an image, the objects/pixels are obtained, which have been smoothed away by the larger Gaussian, but were contained in the image smoothed with the smaller Gaussian. In other words, a scale corresponding to the Gaussian standard deviation σ is obtained, where certain pixels vanish. Small objects or pixel groups already vanish when smoothing with small standard deviations, while larger objects require very large standard deviations. Obviously to get a scale value for each pixel belonging to a certain object, smoothing with different σ-values needs to be performed, creating the so called Gaussian-pyramid. The lowest level of the pyramid is a slightly smoothed version of the original image.

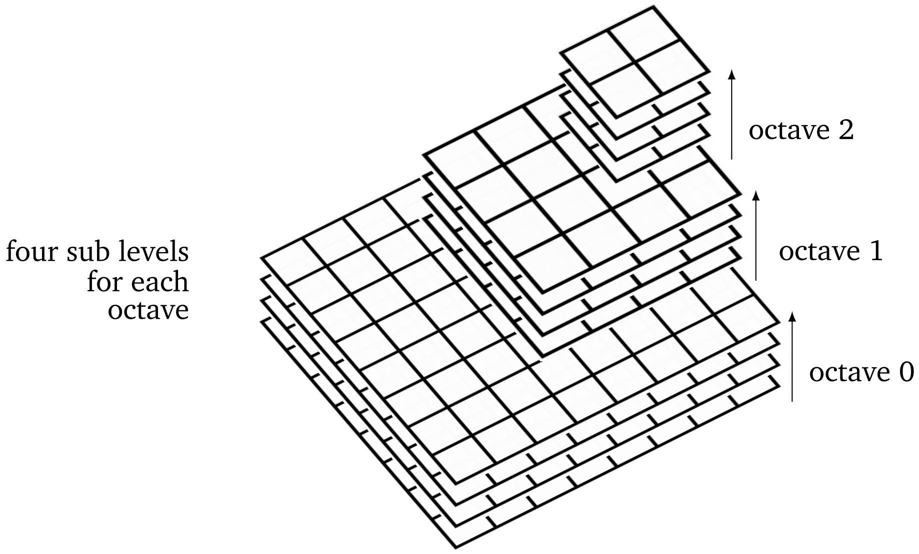


Figure 2.6: Example of a Gaussian-pyramid with three octaves and four sub levels for each octave.

This smoothing is done to get rid of potential outliers. The next level is created by smoothing the underlying layer and sub-sampling it to twice the size. This is done, until the number of defined layers is reached or until the layer-image has the size of one pixel, containing the mean color/gray-value. Alternatively the technique to build the DoG-pyramid can be modified as in the SIFT algorithm by Lowe et al., where each of the layers of the Gaussian-pyramid, denoted as octaves, consists of sub-layers.

Figure 2.6 shows an example structure of this specific Gaussian-pyramid as used in this thesis. Like in the simple Gaussian-pyramid, the bottom layer of the first octave is the presmoothed original image. Additionally practice has shown, that better results are obtained by double sizing the original image and by choosing four octaves with five sub-levels each. Let $o \in [0, \cdots, 3]$ be the octave index and $s \in [0, \cdots, S+3]$, with $S = 2$ the sub-level index. To construct the first octave, the initial image is convolved with the Gaussian with standard deviation $\sigma_{sub} = \sigma \cdot k^{((\text{index of third last sub-level}) \cdot o - 1 + s)}$ resulting in the five sub-levels. The next octave is then initialized with a sub-sampled version of the first initial image. These steps are repeated until the number of desired octaves has be reached. Table 2.1 shows the different standard deviations used in each sub-level for each octave, with $\sigma = 1$ and $k = 2^{1/S} = \sqrt{2}$.

Now, that the Gaussian-pyramid is built, it is quite simple to extract the DoG-pyramid by subtracting two subsequent layers in each octave. In order to find the scale-value for each pixel, the maximum value for the position of the pixel in the DoG-pyramid must be found. To do so, all octaves are up-sampled, via constant interpolation, to the original size. For each pixel all values in all of the $O \cdot (S+2)$ layers are compared, with O being the number of octaves.

sub-level	0	1	2	3	4
octave 0	0.70	1.00	1.41	2.00	2.82
octave 1	1.41	2.00	2.82	4.00	5.65
octave 2	2.82	4.00	5.65	8.00	11.3
octave 3	5.65	8.00	11.3	16.0	22.6

Table 2.1: σ_{sub} -values for each octave in the Gaussian-Pyramid, with $\sigma = 1$ and $k = \sqrt{2}$

Sticking to the example from above, the scale-values for the DoG-levels can be calculated according to the following formula:

$$\text{scale}(o, s, \sigma) = \sigma \cdot k^{((\text{index of third last sub-level}) \cdot o + s)} \quad , \tag{2.9}$$

with o being the index of the octave and σ the standard deviation of the Gaussian at base scale level. A scale value for each pixel can now be calculated. This information can be used as a descriptor like in SIFT and SURF, in order to enhance object detection and recognition. However, in this thesis the scale obtained by this method is used to analyze the optical flow benchmarks w.r.t large displacements in Chapter 4.

2.6 Interpolation of Sparse Flow Fields

For the evaluation of the movement type of a pixel, the movement information within a certain neighborhood is necessary. Problems arise, when this information is not available due to sparse flow fields as given in the KITTI benchmarks. Hence, the interpolation method EpicFlow of Revaud et al.[RWHS15] is used. This section briefly introduces the main concepts of this approach.

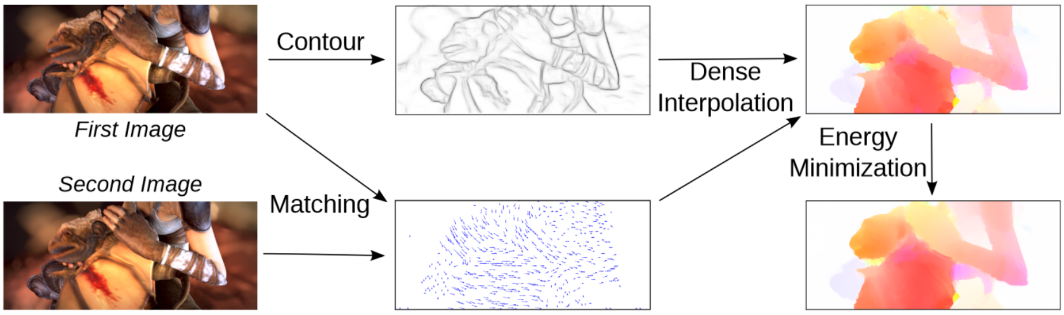


Figure 2.7: Overview of the EpicFlow method of Revaud et al.[RWHS15]

Figure 2.7 depicts the main steps of the EpicFlow algorithm. In the following we will look at all steps except the energy minimization, because only the dense interpolation is of interest. To interpolate a flow field the following inputs are required. The values which should be interpolated, a distance measure and a rule of calculation. Therefore the method needs the frame f^t and a list of matching pixels M, matching point form f^t to f^{t+1}. In order to be edge preserving also the contours of the t'th frame for the later distance calculation $D : f \times f \rightarrow \mathbb{R}^+$ are required. The interpolation rule for a pixel $\boldsymbol{p} \in f$ can be chosen to be the Nadaraya-Watson estimation [Was13] or a locally-weighted affine estimation. The Nadaraya-Watson estimation reads

$$F_{NW}(\boldsymbol{p}) = \frac{\sum_{(\boldsymbol{p}_m, \boldsymbol{p'}_m) \in M} k_D(\boldsymbol{p}_m, \boldsymbol{p}) \boldsymbol{p'}_m}{\sum_{(\boldsymbol{p}_m, \boldsymbol{p'}_m) \in M} k_D(\boldsymbol{p}_m, \boldsymbol{p})} \quad , \tag{2.10}$$

where $k_D(\boldsymbol{p}_m, \boldsymbol{p})$ denotes a Gaussian kernel $\exp(-aD(\boldsymbol{p}_m, \boldsymbol{p}))$ for a distance D with a parameter a. However, the interpolation rule used later on, is the locally-weighted affine interpolation, where $A_{\boldsymbol{p}}$ and $\boldsymbol{t}_{\boldsymbol{p}}$ are the parameters of the affine transformation.

$$F_{LA} = A_{\boldsymbol{p}} \boldsymbol{p} + \boldsymbol{t}_{\boldsymbol{p}}^{\top} \quad . \tag{2.11}$$

The parameters are obtained as a least square solution of a equation system, consisting of the two equations of the type:

$$0 = k_D(\boldsymbol{p}_m, \boldsymbol{p})(A_{\boldsymbol{p}} \boldsymbol{p}_m + \boldsymbol{t}_{\boldsymbol{p}}^{\top} - \boldsymbol{p'}_m) \quad , \tag{2.12}$$

for each match $(\boldsymbol{p}_m, \boldsymbol{p'}_m) \in M$. Since this equation system is over-determined Revaude et al. chose to restrict the matches for a pixel $\boldsymbol{p}$ to its K nearest neighbors according to the distance D. To calculate the distances Revaud et al. use the geodesic distance with respect to a cost map C:

$$D(\boldsymbol{p}, \boldsymbol{q}) = \inf_{\Gamma \in \mathcal{P}_{\boldsymbol{p},\boldsymbol{q}}} \int_{\Gamma} C(\boldsymbol{p}_{\boldsymbol{s}}) \; d\boldsymbol{p}_{\boldsymbol{s}} \quad . \tag{2.13}$$

Here $\mathcal{P}_{\boldsymbol{p},\boldsymbol{q}}$ is the set of all possible paths between $\boldsymbol{p}$ and $\boldsymbol{q}$. $C(\boldsymbol{p}_{\boldsymbol{s}})$ denotes the costs of crossing the pixel $\boldsymbol{p}_{\boldsymbol{s}}$ along the path Γ. The cost map C is obtained by using the contours to build motion boundaries. The contours are extracted using the structured edge detector SED [DZ13]. Therefore the costs between pixels on the same layer are lower than between pixels on different layers. For a fast approximation of C, a geodesic Voronoi Diagram, for clustering the contour image into cells containing one matched pixel, is used by Revaud et al. Dijkstra's algorithm is then used for the shortest path calculation as shown in Figure 2.8.

Using the cost map C and the Equations 2.10 and 2.13, the missing vectors of the ground truth flows of the KITTI 2012 and 2015 benchmark can be interpolated. An example of an interpolated flow field of the KITTI 2012 test suite is shown in Figure 2.9. The matches needed are given by the sparse ground truth. For further details on how the method works, please see [RWHS15].

Now that all necessary foundations have been introduced, let us start off with the first analysis concerning illumination changes.

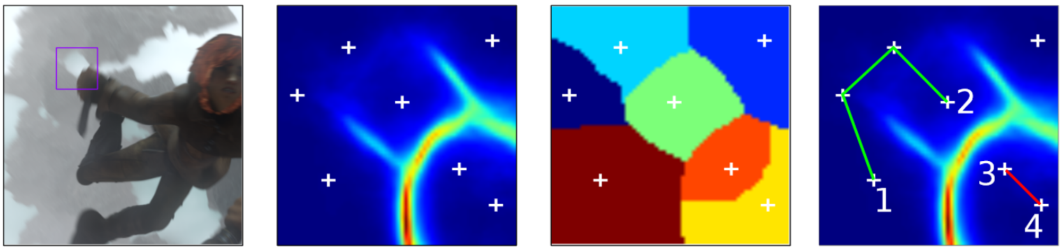


Figure 2.8: Example of the distance calculation of the EpicFlow method [RWHS15]. *From left to right:* Image from the MPI Sintel test suite. Image edges C of the zoomed area. The white crosses denote the matched positions $\{p_m\}$. Calculated Voronoi Diagram. Shortest paths of two neighboring matches.

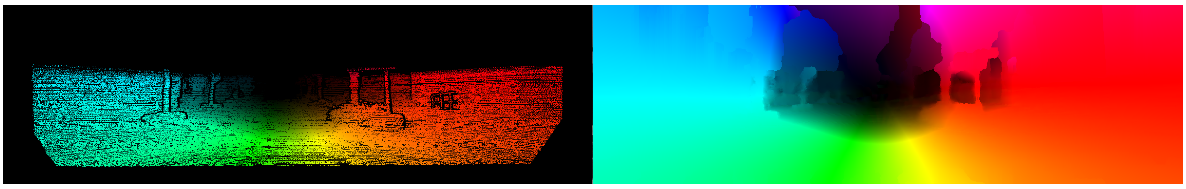


Figure 2.9: Example of an interpolated flow field of the KITTI 2012 benchmark using EpicFlow. *From left to right:* Ground truth flow and interpolated flow.

3 Illumination Changes

A significant difficulty in image processing are varying illumination effects. For example shadows on the street, specular reflections on windows or metal objects, the opening and closing of the camera lens lead to illumination changes. Hager uses brightness transfer functions, short BTF's, to make a statement about the difficulty of a benchmark suite. Because of high local illumination changes, the conclusion is, that the MPI Sintel benchmark is more difficult than the others. This statement is based on the measurements of the minimum, maximum, average, variance and standard deviation of the components of the BTF. The occurring problem is, that the minimum and maximum, as well as the average, do not provide a deep understanding of the brightness changes in the benchmark. Minimum and maximum values should be treated with caution. An image pair with low minimum or high maximum coefficients is not necessarily complicated, if they only occur sparsely. Furthermore, the average is subject to cancellation effects. In conclusion, only the combination of average, variance and standard deviation makes it possible to give evidence on the complexity of an image sequence.

In this chapter, the previously mentioned results are verified using a modified version of the approach of Hager. First, the robust variational approach for local illumination changes is recapitulated and evaluated. Making use of the findings from the measurements, a more suited method, for determining if illumination changes take place or not, is developed. Based on the behavior of the refined method, a standard deviation with a fixed average is introduced as an additional metric in order to analyze the image sequences regarding illumination changes more deeply.

3.1 Review of the Variational Approach for Local Illumination Changes

The aforementioned variational approach is based on the parameterized BTF's proposed by Grossberg and Nayar [GN04]. The BTF in general reads

$$\Phi(\boldsymbol{c}, f) = \overline{\phi}(f) + \sum_{i=1}^{n} c_i \cdot \phi_i(f) \quad , \tag{3.1}$$

with $\overline{\phi}(f)$ being the mean of f. It maps the gray or color values of an input frame to transformed gray or color values, using the coefficients $\boldsymbol{c} = (c_1, \cdots, c_n)^\top$ and n basis functions ϕ_i. In addition, the affine model of Negahdaripur [NY93] is used choosing

$$\overline{\phi}(f) = 0, \phi_1(f) = f, \phi_2(f) = 1 \ . \tag{3.2}$$

This choice of the basis functions leads to the following equation for the relation between the first and the second frame at the position (x, y) at time t. For simplicity, the component descriptions are shortened as follows: $c_1(x, y, t) = c_1$, $c_2(x, y, t) = c_2$, $f(x, y, t) = f^t$ and $f(x + u, y + v, t + 1) = f^{t+1}$ with (u, v) being the given ground truth flow or computed flow.

$$c_1 \cdot f^t + c_2 = f^{t+1} \ . \tag{3.3}$$

To approximate the coefficients the following energy functional, with an additional γ-function, is used. This function turns off the data term when no valid flow is present. It is needed because both the KITTI and Middlebury benchmark contain ground truth flow fields with some vectors not being valid, e.g. no laser measurement.

$$\begin{aligned} E(c_1, c_2) &= \int_\Omega \gamma \Psi\big((c_{1_{x,y,t}} \cdot f_{x,y,t} + c_{2_{x,y,t}} - f_{x+u,y+v,t+1})^2\big) + \alpha \Psi\big(|\nabla c_{1_{x,y,t}}|^2 + |\nabla c_{2_{x,y,t}}|^2\big) \ , \\ \gamma &= \begin{cases} true(1) & \text{if the flow is valid} \\ false(0) & \text{else} \end{cases} \ . \end{aligned} \tag{3.4}$$

To solve for the coefficients c_1 and c_2 the energy functional is discretized and minimized using a sub-quadratic penaliser function $\Psi(s^2) = 2 \cdot \sqrt{s^2 + \lambda^2}$, with $\lambda > 0$.

3.1.1 Coefficient Evaluation on Identical Image Pairs

Interpreting the results of the multiplicative and additive components of sequential images is a challenging task, since the correct solution for the given image sequence is unknown. To be able to judge if the above method performs as expected, it is evaluated on identical image pairs in this section. Setting $f^{t+1} = f^t$ implies the trivial solution that c_1 should be set to one and c_2 to zero. For each of the coefficients c_1 and c_2, small to no deviations from one and zero are expected. The measurements where only made at those positions where the ground truth flow is valid.

Figure 3.1 shows the measurements of the KITTI 2012 benchmark. Note the small deviations of the multiplicative components from one and even larger deviations of the additive components

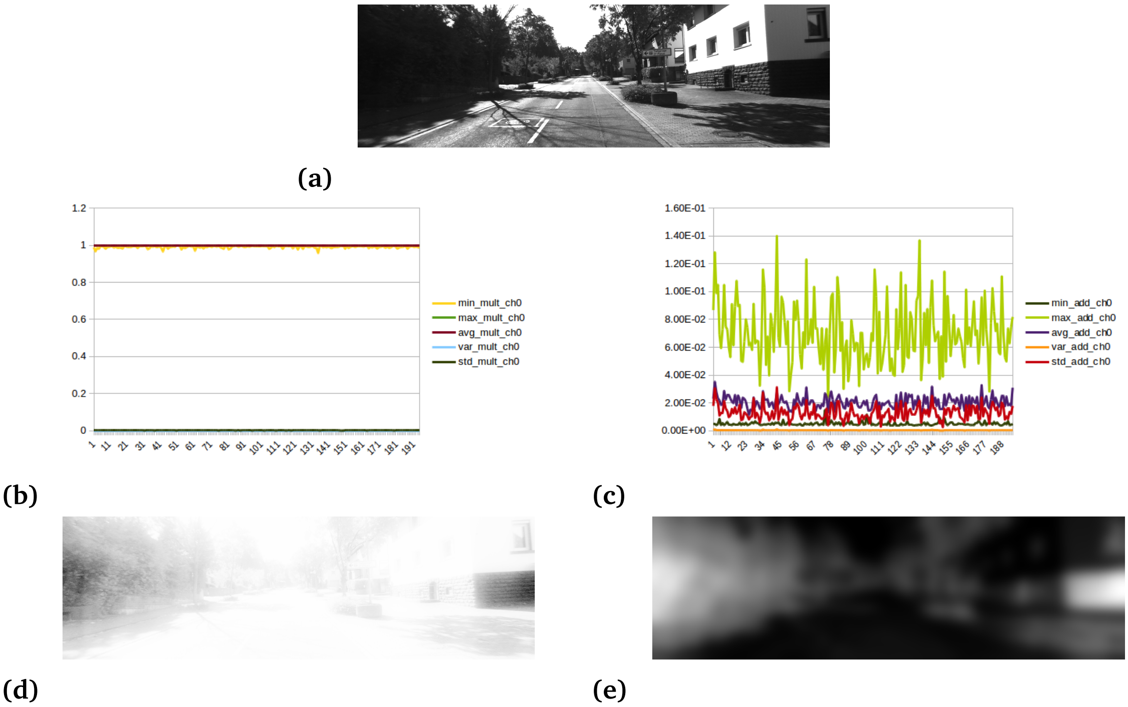


Figure 3.1: Coefficient evaluation on the KITTI 2012 benchmark with the method of Hager ($\alpha = 0.1$). *First row:* (a) First frame. *Second row, from left to right:* (b) Multiplicative component statistics. (c) Additive component statistics. (x axis = frame number) *Second row, from left to right:* (d) Image of the multiplicative components of the first frame. (e) Image of the additive components of the first frame.

from zero. This behavior is explained by the initialization of the components with zero. During computation, the multiplicative components are raised to a value close to one, leading to a rise of the additive components, such that the condition of the BTF's hold. The smoothness term causes the additive and multiplicative components to compete against each other, visualized in the component images (3.1d, 3.1e). Note that the interval of the values is re-scaled to zero and 255 in the images. In order to be as smooth as possible, the white areas try to dominate the black areas and vice versa.

Unfortunately, this behavior complicates the process to make a statement about illumination changes. It is not observable, if there is a change in illumination or not. The measurements of the other benchmarks fortify the observation of the effects made in the KITTI 2012 benchmark.

As mentioned before the goal is to create a refined model, in order to better explain the coefficient values. Furthermore, it is desired, that the new model suits the expectations of the value of the coefficients, when applied on identical images.

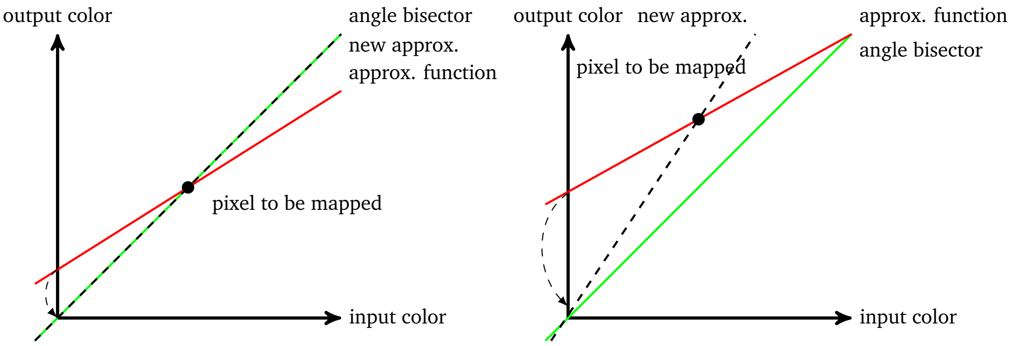


Figure 3.2: Examples of linear regressions through one point. Red denotes the method of Hager, dotted the intuition of the modified version.

3.2 Improving the Method

As shown in the previous section, there is a need for a modified version of the method of Hager. The linear regression performed in each point only has one data point corresponding to the current pixel. Since more than one point is needed to define or fit a unique line, assumptions are needed about all linear regressions in the neighborhood of a pixel. In Hager's method, these assumptions include the smoothness of the multiplicative component and additive component of the linear regressions. In contrast to this method a different approach is made. The previous assumption, that the additive components should be smooth is replaced by the assumption, that the additive components should be rather small. This is motivated by assuming, that there are only tiny to no illumination changes in an image sequence, resulting in the angle bisector for the linear regression.

The impact of the modification is depicted in Figure 3.2. The figure shows two different scenarios of the linear regression. The first scenario, on the left hand side, visualizes the case of no illumination change. It shows the fitting of the linear regression (red line) in the point located at the position of the input pixel value and transformed value. The steepness of the approximation is lower than the one of the angle bisector (green line) due to the initialization with zero. This initialization issue is corrected by the new method, by pulling the additive component close to the origin (dotted line). This steepens the function, increasing the gradient to a value near one.

The second scenario, on the right hand side, shows the regression when small illumination changes are present. Again, we want to achieve that the method approximates the functions as close as possible to the angle bisector, meaning that the additive component is zero and the multiplicative component is one. Analog to the first scenario, a better fit is obtained by pulling down the additive component towards zero.

What remains is the transfer of these insights to a refined method. Instead of penalizing the gradient magnitude, only the magnitude itself is penalized. Since Ψ is an approximation of the magnitude, c_2^2 is only used in the new term with an additional weight parameter β.

$$E(c_1, c_2) = \int_\Omega \gamma\Psi((c_1 \cdot f^t + c_2 - f^{t+1})^2) + \alpha\Psi(|\nabla c_1|^2) + \beta\Psi(c_2^2) \;\; dxdy \;\; . \tag{3.5}$$

The next step, for the minimization of this energy functional, is to set up the Euler-Lagrange equations. Note the absence of the smoothness terms in the second line of the equation 3.6. In fact, by neglecting the neighborhood of the additive component, we obtain a weighted sparsity term:

$$\begin{aligned} F_{c_1} - \partial_x F_{c_{1_x}} - \partial_y F_{c_{1_y}} &= 0 \;\; , \\ F_{c_2} &= 0 \;\; . \end{aligned} \tag{3.6}$$

Obviously, the boundary conditions are also reduced to one equation

$$n^\top \begin{pmatrix} F_{c_{1_x}} \\ F_{c_{1_y}} \end{pmatrix} = 0 \;\; , \tag{3.7}$$

since the additive components are no more subject to diffusion. The partial derivatives of F needed for the calculation of the Euler-Lagrange equations are shown below:

$$\begin{aligned} F_{c_1} &= \gamma\Psi'((c_1 \cdot f^t + c_2 - f^{t+1})^2) \cdot 2(c_1 \cdot f^t + c_2 - f^{t+1}) \cdot f_t \;\; , \\ F_{c_{1_x}} &= \alpha\Psi'(|\nabla c_1|^2) \cdot 2c_{1_x} \;\; , \\ F_{c_{1_y}} &= \alpha\Psi'(|\nabla c_1|^2) \cdot 2c_{1_y} \;\; , \\ F_{c_2} &= \gamma\Psi'((c_1 \cdot f^t + c_2 - f^{t+1})^2) \cdot 2(c_1 \cdot f^t + c_2 - f^{t+1}) + \beta\Psi'(c_2^2) \cdot 2c_2 \;\; . \end{aligned} \tag{3.8}$$

Plugging these in Equation 3.6 and dividing by 2, yields the two equations

$$\begin{aligned} \gamma\Psi'((c_1 \cdot f^t + c_2 - f^{t+1})^2) \cdot (c_1 \cdot f^t + c_2 - f^{t+1}) \cdot f^t - \alpha\text{div}(\Psi'(|\nabla c_1|^2) \cdot \nabla c_1) &= 0 \;\; , \\ \gamma\Psi'((c_1 \cdot f^t + c_2 - f^{t+1})^2) \cdot (c_1 \cdot f^t + c_2 - f^{t+1}) + \beta\Psi'(c_2^2) \cdot c_2 &= 0 \;\; . \end{aligned} \tag{3.9}$$

For a more compact write up we introduce the following abbreviations:

$$\begin{aligned} \Psi'((c_1 \cdot f^t + c_2 - f^{t+1})^2) &= {\Psi'}_{d_{i,j}} \ , \\ \Psi'(c_2^2) &= {\Psi'}_{d2_{i,j}} \ , \\ \Psi'(|\nabla c_1|^2) &= {\Psi'}_{s_{i,j}} \ . \end{aligned} \tag{3.10}$$

Next the integral and functions are discretized, e.g. the continuous function f becomes a discrete function $f_{i,j}$ with (i,j) being the pixel position. The derivatives of c_1 are approximated by central differences assuming that the pixel width and height is the same $h_x = h_y = h$, leads to the two equations for each pixel position $(i,j) \in \Omega$:

$$\begin{aligned} \gamma {\Psi'}_{d_{i,j}} \cdot (c_{1_{i,j}} \cdot f^t_{i,j} + c_{2_{i,j}} - f^{t+1}_{i,j}) \cdot f^t_{i,j} - \alpha \sum_{(\tilde{i},\tilde{j}) \in N(i,j)} \frac{{\Psi'}_{s_{\tilde{i},\tilde{j}}} + {\Psi'}_{s_{i,j}}}{2} \left(\frac{c_{1_{\tilde{i},\tilde{j}}} - c_{1_{i,j}}}{h^2} \right) &= 0 \ , \\ \gamma {\Psi'}_{d_{i,j}} \cdot (c_{1_{i,j}} \cdot f^t_{i,j} + c_{2_{i,j}} - f^{t+1}_{i,j}) + \beta {\Psi'}_{d2_{i,j}} \cdot c_{2_{i,j}} &= 0 \ , \end{aligned} \tag{3.11}$$

with $N(i,j)$ being the neighbors of the pixel (i,j). To determine c_1 and c_2 the fixed point iteration k, shown in 3.12, is solved by using the lagged non-linearity method [CM99, DV97].

$$\begin{aligned} c^{k+1}_{1_{i,j}} &= \frac{\gamma {\Psi'}_{d_{i,j}} f^t_{i,j} f^{t+1}_{i,j} - \gamma {\Psi'}_{d_{i,j}} f^t_{i,j} c^k_{2_{i,j}} + \alpha \sum_{(\tilde{i},\tilde{j}) \in N(i,j)} \frac{{\Psi'}_{s_{\tilde{i},\tilde{j}}} + {\Psi'}_{s_{i,j}}}{2} \left(\frac{c^k_{1_{\tilde{i},\tilde{j}}}}{h^2} \right)}{\gamma {\Psi'}_{d_{i,j}} {f^t_{i,j}}^2 + \alpha \sum_{(\tilde{i},\tilde{j}) \in N(i,j)} \frac{|N(i,j)|}{h^2}} \ , \\ c^{k+1}_{2_{i,j}} &= \gamma \cdot \frac{{\Psi'}_{d_{i,j}} f^{t+1}_{i,j} - {\Psi'}_{d_{i,j}} c^k_{1_{i,j}} f^t_{i,j}}{{\Psi'}_{d_{i,j}} + \beta {\Psi'}_{d2_{i,j}}} \ . \end{aligned} \tag{3.12}$$

Let us check, how this modification improves the approximation of the coefficients when applied on identical image pairs.

3.3 First Results of the Refined Method on Identical Image Pairs

As mentioned before the energy functional is modified in such a way, that when used on identical images, the multiplicative coefficients should result in one and the additive components should result in zero. The aim is to create a clear border, if illumination changes take place. Therefore, the parameter α is set to 0.1 and β to 0.2. 1500 iterations are performed by the lagged non-linearity method, updating the Ψ primes every 150 steps. As before only the areas where the flow is valid where taken into account. Let us start off with the Red channel statistics of the KITTI 2015 test suite Figures 3.3d, 3.3e, 3.3i and 3.3j.

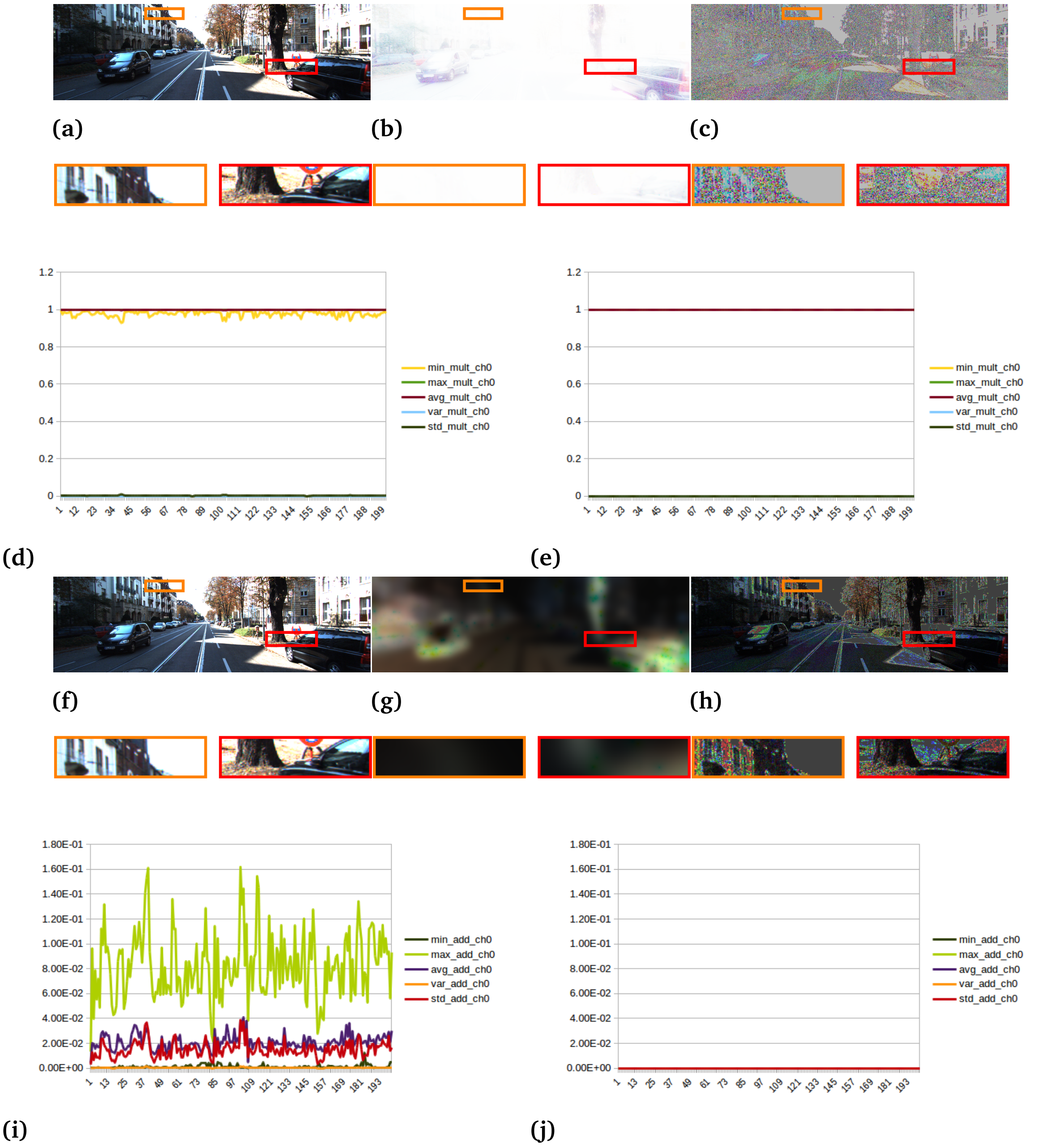


Figure 3.3: Coefficient evaluation on the KITTI 2015 benchmark, with a zoom box of the houses on the left and of the bottom of the tree on the right. *First row, from left to right:* (a) First frame. Multiplicative components of the first frame, with (b) being the original version and (c) the improved version. *Second row, from left to right:* Multiplicative component statistics Red channel of the (d) original method and of the (e) modified method. *Third row, from left to right:* (f) First frame. Additive components of the first frame, with (g) being the original version and (h) the improved version. *Fourth row, from left to right:* Additive component statistics of the (i) original method and of the (j) modified method. (Original method: $\alpha = 0.1$, modified method: $\alpha = 0.1$, $\beta = 0.2$, x-axis = frame number).

3.3.1 Evaluation on the KITTI Benchmarks

Figures 3.3d, 3.3e, 3.3i and 3.3j show the elimination of the small deviations of the multiplicative components from one. Same holds for the larger deviations of the additive components from zero. Taking a look at Figures 3.3a, 3.3b, 3.3c, 3.3f, 3.3g and 3.3h, reveal the improvement of penalising the magnitude of the additive components. It has to be taken into account, that the interval of $[0, 255]$ is adjusted to the minimum and maximum of the coefficients in each color channel for visualization purposes. Comparing the multiplicative component image of the original method (b) with the one of the new method (c) shows, how slight the deviations from one are in the multiplicative component image of the new method. Same holds for the additive component images. Areas, where this can be seen best, are the buildings on the left hand side or next to the car located on the bottom right.

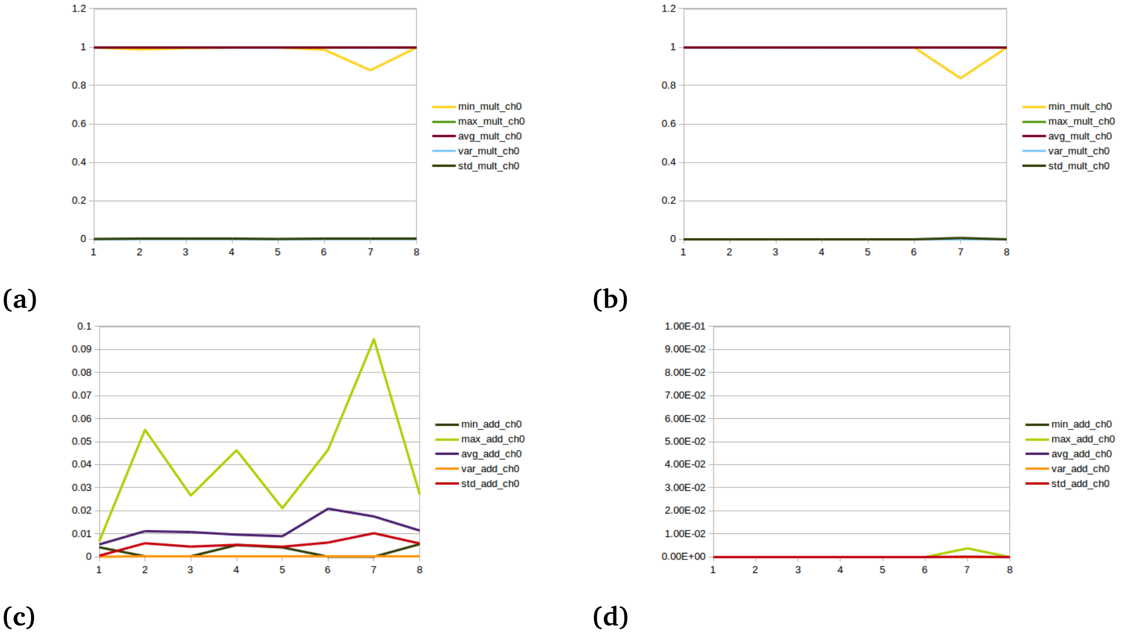
Analogous observations can be made in the additive component images with small deviations from zero. Similar results were achieved in the remaining two channels, as well as in the KITTI 2012 benchmark.

3.3.2 Evaluation on the Middlebury and MPI Sintel Benchmark

In contrast to the KITTI test suites, irregularities appear in the seventh frame set of the Middlebury benchmark, shown in Figure 3.4.

The first image pair having such an irregularity is part of the Urban3 image sequence of the Middlebury Benchmark. Here the minimum value of the multiplicative components reaches a lower value than 0.9 with both methods and this has an effect on the additive components being raised away from zero. The explanation for the occurrence of these values lies within the input image. Figure 3.5 row one, shows the input image, the multiplicative and additive component images, together with the zoom of the upper right corner.

The color values in this corner are zero, leaving infinite solutions for our equation system. Due to the fact, that both coefficients are initialized with zero, it is already an optimal solution in this region accept at the border to the non zero region. Both data and smoothness term are fulfilled. The large size of the zero region slows down the inpainting effect of the smoothness term of the multiplicative components. Same holds for the clean MPI Sintel frames. An example can be found in the Ambush5 scene Frame 33 and 44, where zero values can be found in the region of the hair tips of the women and in the beard of the man depicted in Figure 3.5.



(a) **(b)**

(c) **(d)**

Figure 3.4: Coefficient evaluation on the Middlebury benchmark of the Red channel. *First row, from left to right:* Multiplicative component statistics of the original method (a) and of the modified method (b). *Second row, from left to right:* Additive component statistics of the original method (c) and of the modified method (d). Original method: $\alpha = 0.1$, modified method: $\alpha = 0.1$, $\beta = 0.2$. X-axis = frame number.

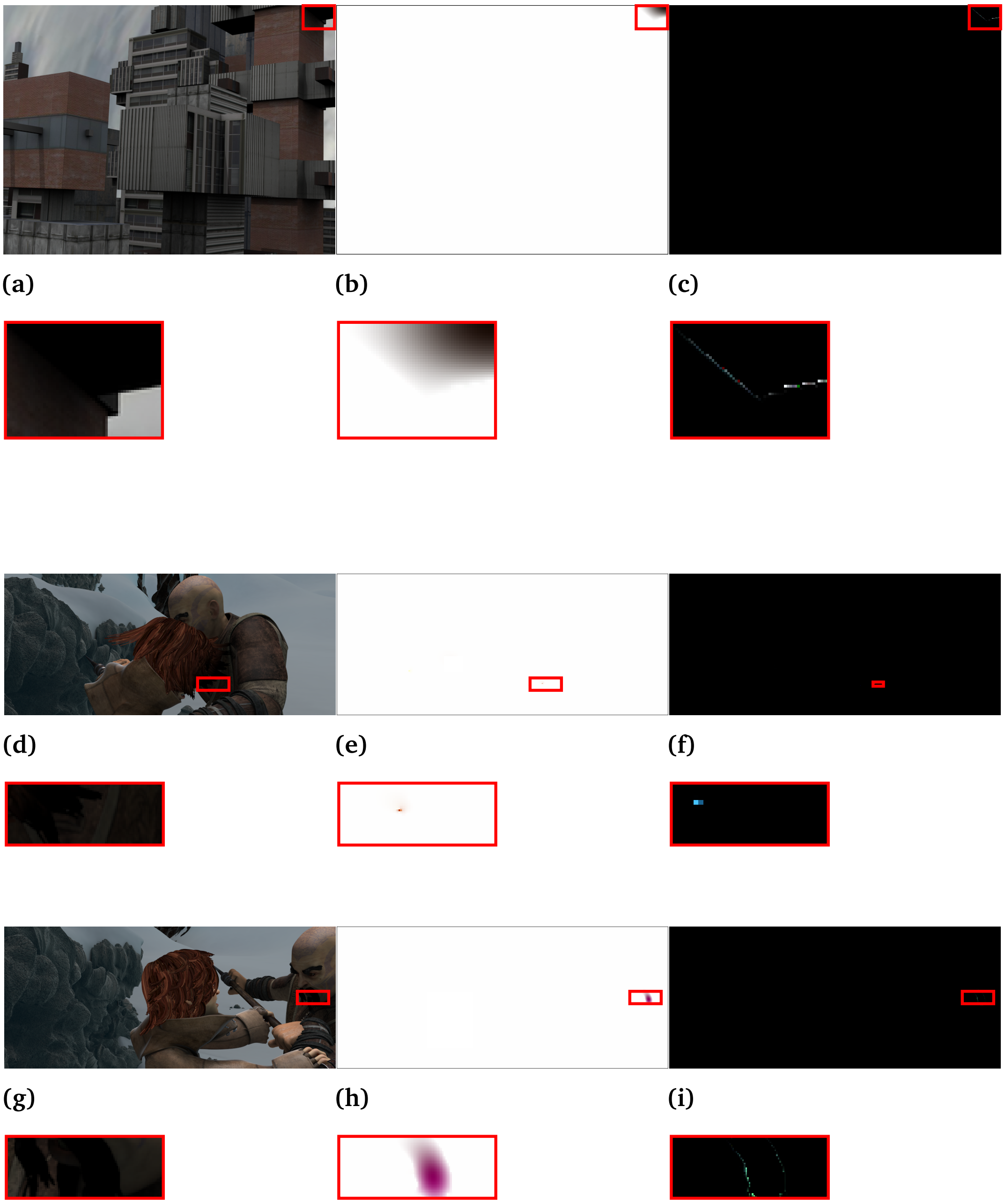


Figure 3.5: Coefficient evaluation on the Middlebury and MPI Sintel benchmark, with zoom-boxes of regions of interest. *First row, from left to right:* (a) Frame ten of the Urban3 sequence. (b) Multiplicative component image. (c) Additive component image. *Second row, from left to right:* (d) Frame 33 of the Ambush 5 sequence. (e) Multiplicative component image. (f) Additive component image. *Third row, from left to right:* (g) Frame 44 of the Ambush 5 sequence. (h) Multiplicative component image. (i) Additive component image. ($\alpha = 0.1$ and $\beta = 0.2$ for all).

3.3.3 Adding Saturated Region Handling

Not only zero regions in the input frames cause difficulties, but also regions, where the color values are fully saturated, having a value of 255. Due to the fact, that only values between zero and 255 can be measured, we obtain plateaus in these areas worsening the linear regression in the neighborhood. To counter this, the function γ is extended. Now the function turns off the data term, whenever a pixel is encountered, with the values of zero or 255 in either of the channels of the first or second frame and also, if no valid flow is available. The energy functional is rewritten as follows:

$$\begin{aligned} E(c_1, c_2) &= \int_\Omega \gamma \Psi((c_1 \cdot f^t + c_2 - f^{t+1})^2) + \alpha \Psi(|\nabla c_1|^2) + \beta \Psi(c_2^2) \;\; dxdy \;\; , \\ \gamma &= \begin{cases} true(1) & \text{if the flow is valid and color value is not saturated or zero} \\ false(0) & \text{else} \end{cases} \quad . \end{aligned} \tag{3.13}$$

Setting up the Euler-Lagrange equations and discretizing them as before, results in the following iterative scheme:

$$\begin{aligned} c_{1_{i,j}}^{k+1} &= \frac{\gamma \Psi'_{d_{i,j}} f_{i,j}^t f_{i,j}^{t+1} - \gamma \Psi'_{d_{i,j}} f_{i,j}^t c_{2_{i,j}}^k + \alpha \sum_{(\tilde{i},\tilde{j}) \in N(i,j)} \frac{\Psi'_{s_{\tilde{i},\tilde{j}}} + \Psi'_{s_{i,j}}}{2} \left(\frac{c_{1_{\tilde{i},\tilde{j}}}^k}{h^2} \right)}{\gamma \Psi'_{d_{i,j}} {f_{i,j}^t}^2 + \alpha \sum_{(\tilde{i},\tilde{j}) \in N(i,j)} \frac{|N(i,j)|}{h^2}} \\ c_{2_{i,j}}^{k+1} &= \gamma \cdot \frac{\Psi'_{d_{i,j}} f_{i,j}^{t+1} - \Psi'_{d_{i,j}} c_{1_{i,j}}^k f_{i,j}^t}{\Psi'_{d_{i,j}} + \beta \Psi'_{d2_{i,j}}} \quad . \end{aligned} \tag{3.14}$$

Since the saturated pixels are treated as if the ground truth flow is not valid, they will not be taken into account in the statistics of the sequential images.

The above results have shown, that if no illumination change takes place in a scene, the multiplicative components will be assigned to one and the additive components will be assigned to zero. This aspect of the method enables to say, that if deviations of the multiplicative components from one and deviations of the additive components from zero are measured, illumination changes from one frame to the other are present. The higher the deviations from one and zero are, the more complicated the illumination changes are.

This leads to the following metric, based on the standard deviation for the calculation of illumination changes:

$$
\begin{aligned}
\text{STDM} &= \sqrt{\frac{1}{n}\sum\nolimits_{(i,j)\in\tilde{\Omega}}({c_1}_{i,j}-1)^2}\ , \\
\text{STDA} &= \sqrt{\frac{1}{n}\sum\nolimits_{(i,j)\in\tilde{\Omega}}({c_2}_{i,j})^2}\ .
\end{aligned}
\tag{3.15}
$$

$\tilde{\Omega}$ denotes the pixels, which have valid ground truth flow and do not belong to a saturated region. In the following, this new metric is used to analyze the illumination changes of the sequences of each benchmark.

3.4 Illumination Change Evaluation

In the sections above a new method was developed together with a new metric to quantify illumination changes. The first part of the section evaluates the performance of the introduced concepts. We will see, that the new robust method approximates the brightness change better than the old version. The second part evaluates the illumination changes found in each benchmark.

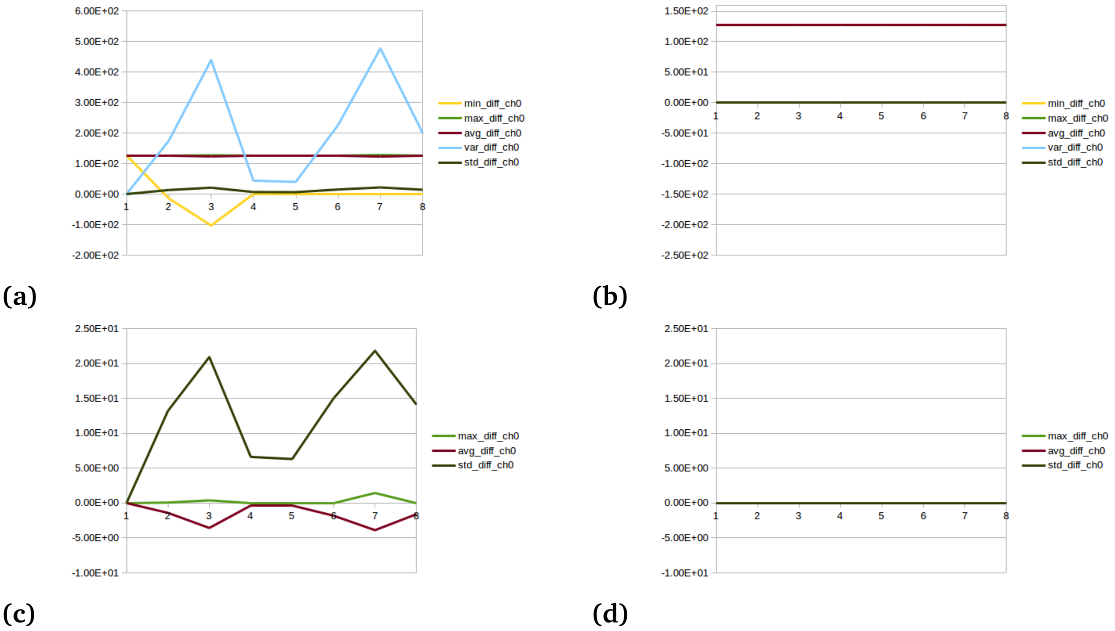
3.4.1 Difference Evaluation

First, let us only consider the approximation of the multiplicative and additive components. To prove, that the novel method improves the approximation of the components, the difference (diff) of the second frame and the transformed first frame is calculated in each pixel, as shown in the following Equation.

$$\text{diff}_{i,j} = f^{t+1}_{i+u,j+v} - ({c_1}_{i,j} \cdot f^{t}_{i,j} + {c_2}_{i,j}) + 127. \tag{3.16}$$

First, the differences are evaluated with the coefficients approximated by the method of Hager. Afterwards, they are measured with the novel technique. Taking a look at the statistics of the Red channel of the benchmarks, presented in Figure 3.6 and 3.7, reveals the improvements achieved by the new method.

In the left column Figure 3.6 of the Middlebury statistics, we can observe, that the minimum and maximum values showing severe deviations of the color values from 127 when evaluated with the old method. This can be seen best in the left image of the second row (c). The second row of Figure 3.6 only shows the maximum and average, both shifted by -127, and



(a) **(b)**

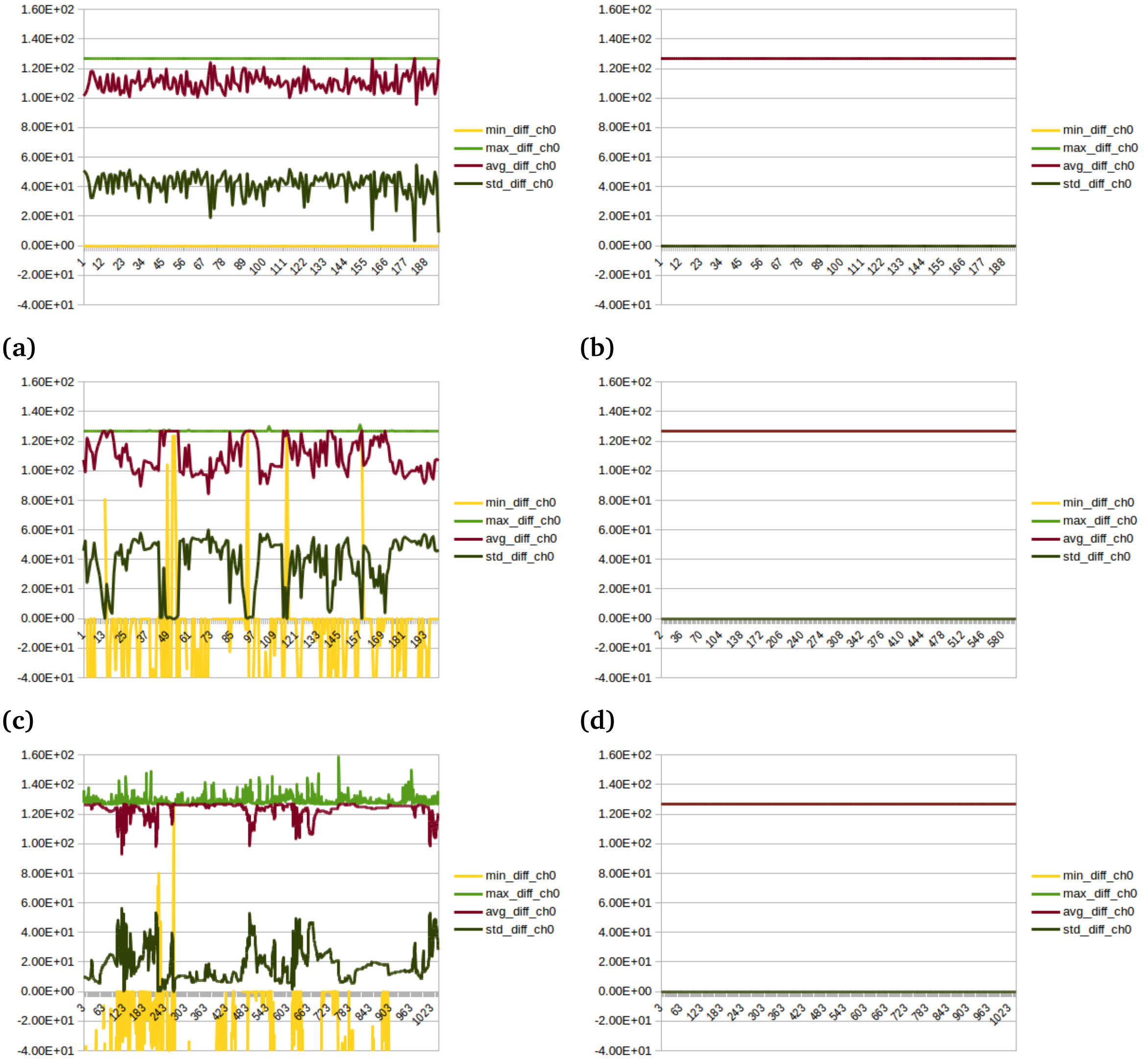
(c) **(d)**

Figure 3.6: Difference evaluation on the Middlebury benchmark channel 0. *First row. from left to right:* (a) Differences when the illumination change is approximated with the original method ($\alpha = 0.1$). (b) Differences when the illumination change is approximated with the modified method ($\alpha = 0.1$, $\beta = 0.2$). *Second row. from left to right:* (c, d) Same as the first row but without the shift by 127, min-values and variance for a better visualization of the deviations. X-axis = frame number.

the standard deviation, in order to better see smaller deviations. The peaks in the sub figure belong to the third (Grove3) and seventh (Urban3) sequence. One reason for this behavior is, that these scenes are synthetic and contain a lot of saturated values, which are taken into account in the old method leading to wrong transformed color values.

Figure 3.7 depicts the difference statistics of the other benchmarks. The first row shows the results of the KITTI 2012 test suite. Here the values of the average difference fluctuate around 110. Same holds for the KITTI 2015 benchmark in row two. In comparison, the average difference values of the MPI Sintel benchmark show some outlier regions as well as some areas, where the average is close to 127. Recall, a value of 127 stands for no difference between the color values of the second frame and the transformed first frame. The strong fluctuations in both KITTI 2012 and 2015 can be explained by the sparse ground truth flow fields. The more dense the flow field is, the more values are taken into account. Therefore, a more stable average is observed in row three of the statistics of the MPI Sintel test suite.

In comparison, the differences calculated with the components, approximated by the new method, show no deviations from 127. The statistics in the right column of Figure 3.6 and 3.7 show, that similar results were obtained across all benchmarks. All averages lie perfectly



(a) **(b)**

(c) **(d)**

(e) **(f)**

Figure 3.7: Difference evaluation on the Red channel of the other benchmarks. *First column, from top to bottom:* Illumination change approximated with the original method ($\alpha = 0.1$). (a) KITTI 2012, (c) KITTI 2015, (e) MPI Sintel. *Second column, from top to bottom:* Illumination change approximated with the modified method ($\alpha = 0.1$, $\beta = 0.2$). (b) KITTI 2012, (d) KITTI 2015, (f) MPI Sintel. X-axis = frame number.

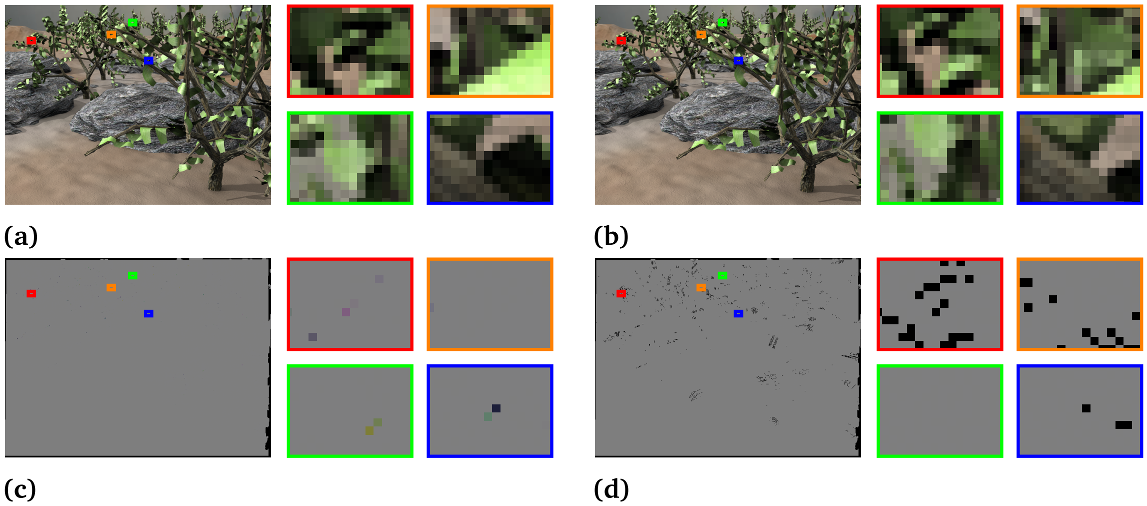


Figure 3.8: Difference evaluation on the Middlebury benchmark Grove3 sequence. *From top to bottom:* (a) Frame 10. (b) Frame 11. (c) Differences with the method of Hager ($\alpha = 0.1$). (d) Differences with the novel method ($\alpha = 0.1$, $\beta = 0.2$).

on 127 and the standard deviation shows no fluctuations. This means, that all illumination changes, which take place, are resembled in the coefficients. If we would have had fluctuations in the difference, like in the old method, it would mean, that we would only consider a part of the total illumination change or approximate it badly. This proves that it makes sense only to take those pixels into account, for the later metric calculation, where the flow is known, and which do not belong to saturated or zero regions.

To enforce this statement, let us look at Figure 3.8 showing the difference images of the Middlebury benchmark Grove3 scene between frame 10 and 11. Black pixels in the difference images denote no flow, zero or saturated regions. Let us first focus on the areas surrounded by the red and orange boxes. These parts of the image show the effect of the γ function canceling out those parts, where no ground truth information is available or which are saturated (c, d). Regarding the green box in the difference image of the old method (c), both the light green and dark yellow difference pixel vanish, when approximated with the new method (d). Same holds for the turquoise pixel in the blue box. Since the pixels vanish into gray, it shows, that the color was better transformed from the first to the second frame.

Next, we will look at the results, when applying the novel metric introduced in Section 3.3 to measure illumination changes.

3.4.2 STDM and STDA Evaluation

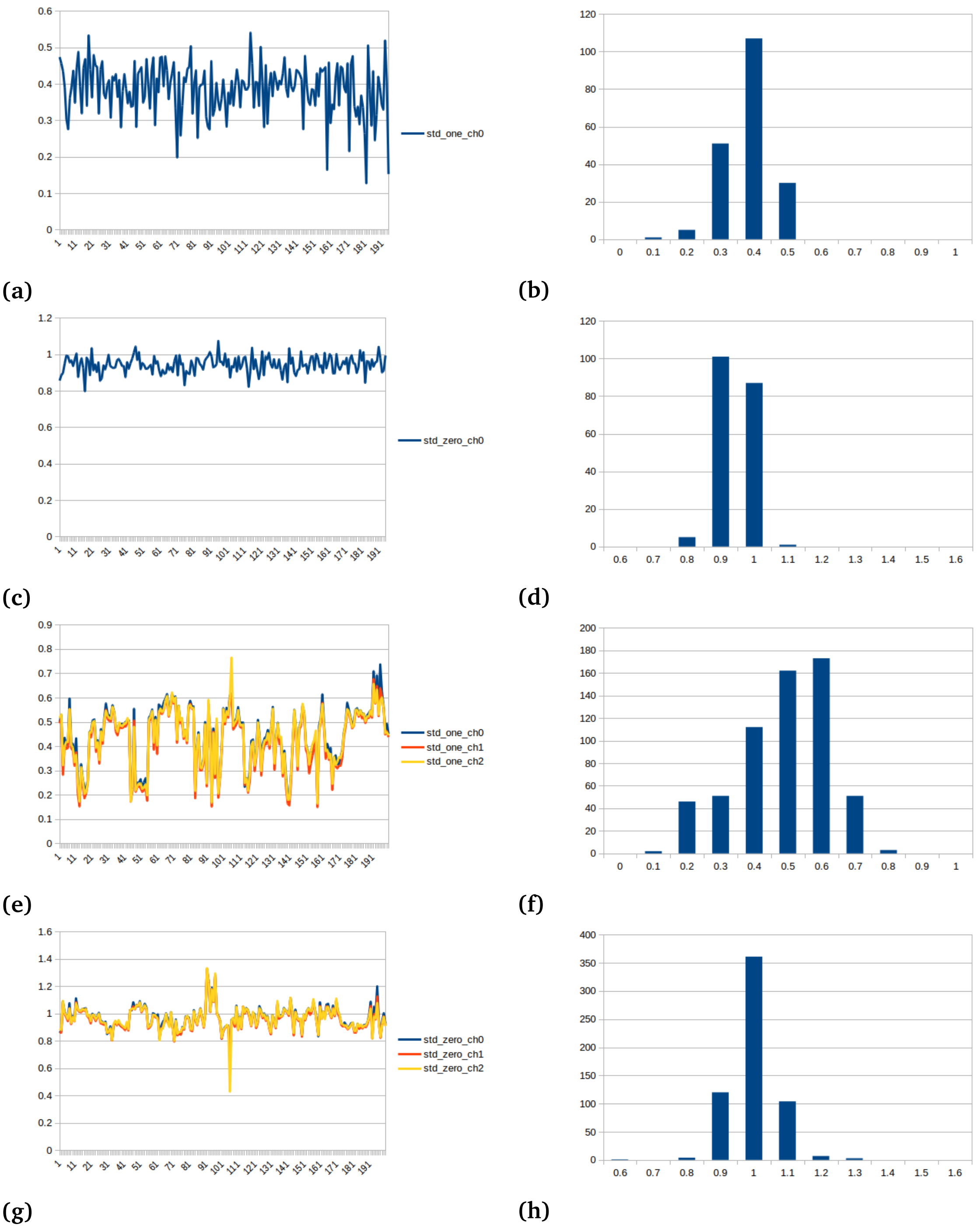
The following Figures 3.9 and 3.10 show the standard deviation from one/zero of the multiplicative/additive components for each image sequence in the different benchmarks. We will begin by discussing the results of the KITTI 2012, KITTI 2015, Middlebury and MPI Sintel benchmark separately, followed by a comparison over all benchmarks via scatter-plots. The computed flow is obtained by the first order approach of Zimmer et al. [ZBW+09] and will also be used in the other evaluation chapters later on.

Figure 3.9 visualizes the STDM and STDA evaluation on the KITTI 2012 benchmark. The further the standard deviations are away from zero, the more the multiplicative or additive coefficients deviate from one or zero respectively. Here the highest deviations lie at a value of about 0.55. Compared to the other benchmarks this value lies in the mid-table. Since the majority of the other standard deviation values lie at 0.3 to 0.4, means, that the total benchmark seems to have moderate local illumination changes. Also the additive component deviations do not vary more than 1.1 away from zero, but also not less than 0.8. The benchmark mainly consists of global illumination changes, with a few sequences or regions containing local illumination changes. Causes for these results could be the impact of the gray scale images or the sparse ground truth, such that only a few pixels fall into regions with strong illumination changes.

The KITTI 2015 benchmark contains more local illumination changes than the older KITTI 2012 test suite. This is due to the higher values of the standard deviation from one of the multiplicative components, depicted in Figure 3.9 row three. The amount of global illumination changes is in average the same, accept for a couple of outliers. Depicted in the histograms, it seems, that the scenes from the KITTI 2015 test suite do not vary as much form the ones of the KITTI 2012 benchmark. Also the difference of gray and color images does not play a decisive role.

Looking at the values of the Middlebury benchmark, shown in Figure 3.10 row one and two, reveals, that this benchmark is the one with the lowest amount of illumination changes. This is due to the fact, that either some of the scenes were taken indoors, or that in the case of synthetically images, the majority of the lighting is ambient light. This is reflected by the low multiplicative component deviations together with the additive component deviations of about 1 for all sequences.

The benchmark with the widest range of deviations is the clean MPI Sintel benchmark, Figure 3.10 row three and four. The values of the multiplicative component deviations range from about 0.05 to 8.5 and the additive component deviations range from about 0.82 to 1.5. Clearly, this is due to the variety and amount of scenes in the training data set, and that they rendered the scenes with as much naturalistic lighting as possible. In conclusion, this benchmark contains some of the most complex scenes with regards to illumination changes. An interesting observation is, that for all benchmarks, the majority of the standard deviations of the additive



(a)

(b)

(c)

(d)

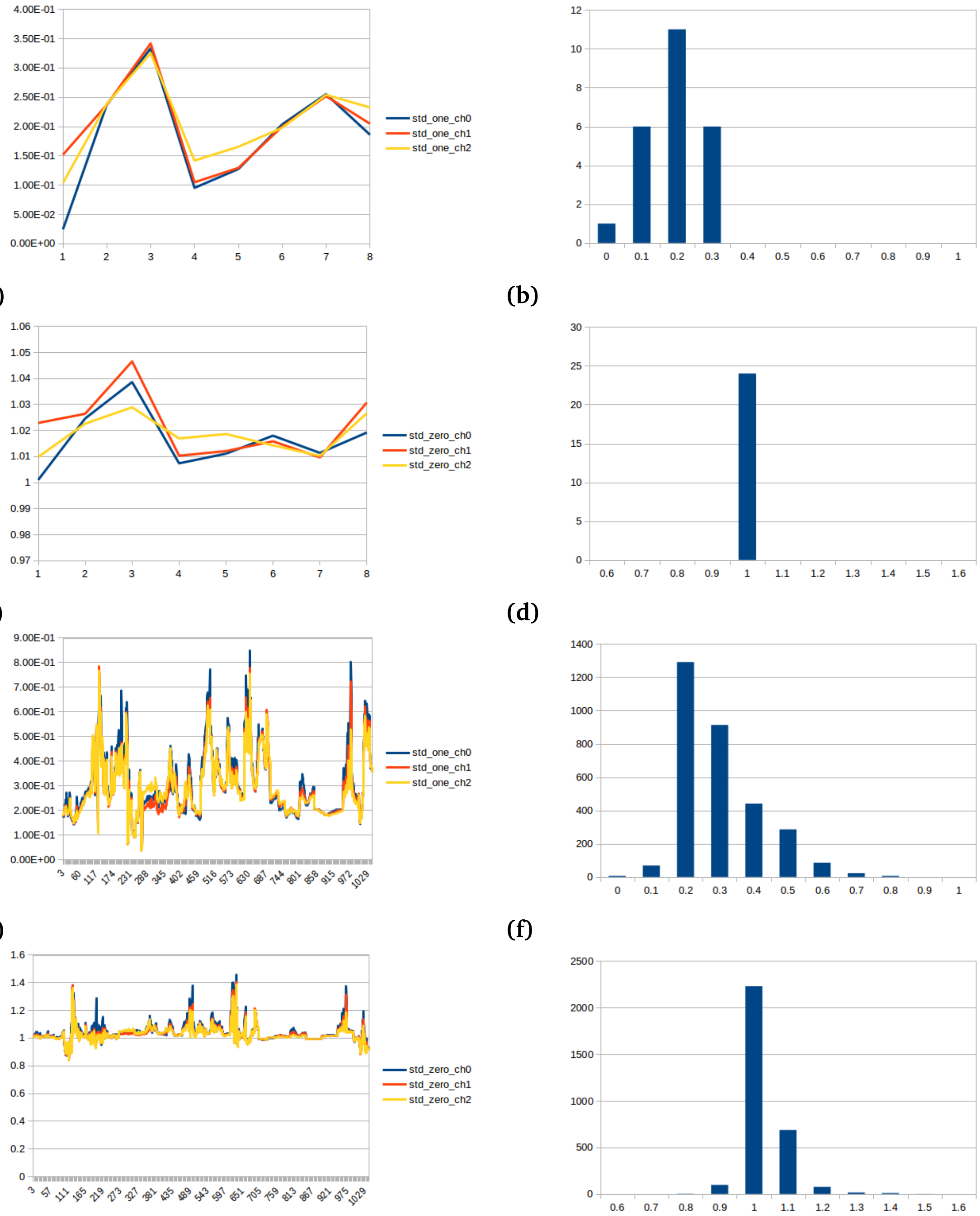
(e)

(f)

(g)

(h)

Figure 3.9: STDM and STDA evaluation on the KITTI 2012 and 2015 benchmark (x-axis = frame number) with corresponding histograms. *First row, from left to right:* (a) STDM (2012). (b) Histogram of (a). *Second row. from left to right:* (c) STDA (2012). (d) Histogram of (c). *Third row, from left to right:* (e) STDM (2015). (f) Histogram of (e). *Fourth row, from left to right:* (g) STDA (2015). (h) Histogram of (g).



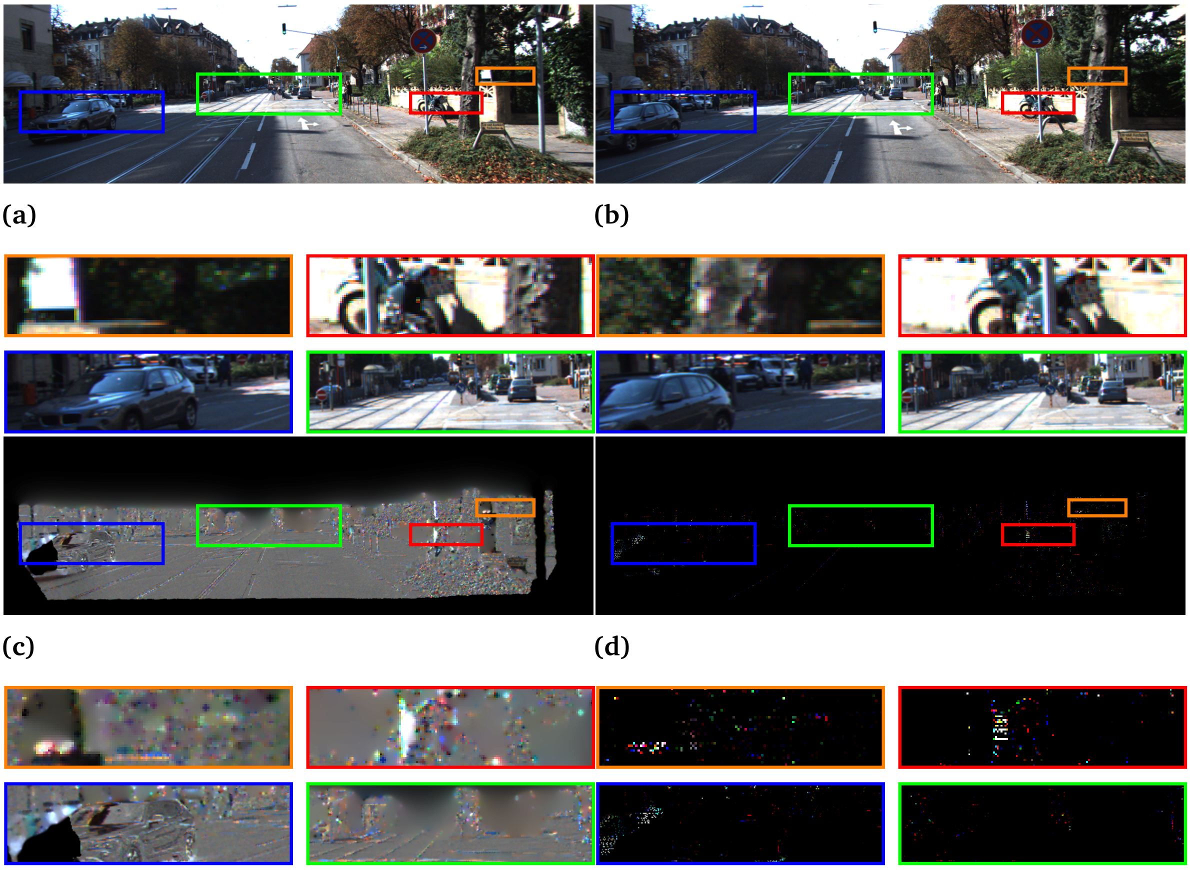
(a) **(b)**

(c) **(d)**

(e) **(f)**

(g) **(h)**

Figure 3.10: STDM and STDA evaluation on the Middlebury and MPI Sintel benchmark (x-axis = frame number) with corresponding histograms in the right column. Middlebury: *First row, from left to right:* (a) STDM. (b) Histogram of (a). *Second row. from left to right:* (c) STDA. (d) Histogram of (c). MPI Sintel: *Third row, from left to right:* (e) STDM. (f) Histogram of (e). *Fourth row. from left to right:* (g) STDA. (h) Histogram of (g).



(a)

(b)

(c)

(d)

Figure 3.11: Coefficient evaluation on the KITTI 2015 benchmark Sequence 2, with a zoom box of the houses on the left and of the bottom of the tree on the right. *First row, from left to right:* First and second frame. *Second row, from left to right:* Normalized multiplicative component image and normalized additive component image ($\alpha = 0.1$, $\beta = 0.2$).

components from zero lie around 1 or 1.1. Let us take a look where these values come from. Figure 3.11 depicts the multiplicative and additive components of the second sequence of the KITTI 2015 benchmark. The high values of the additive component are mostly in those regions, where objects were occluded by other ones (red and orange zoom box). A large amount of low additive components are found throughout the sequence in the red and blue channel (green and blue zoom box). Especially in the areas of the signs and traffic lights. So a lot of small compensations are made by the additive components, leading to the aforementioned distributions.

But which benchmark is now more complex regarding illumination changes?

To answer this question, we need a more suited diagram to make comparisons. The scatterplot, top sub-figure of Figure 3.12, shows the distribution of the image sequences of each benchmark according to their standard deviations of the multiplicative/additive components from one/zero. In contrast to the histograms, we can directly see the combination of both standard deviations from one and zero. A point located at the origin would mean that there is no illumination change in the corresponding image sequence. The further a point is away from the origin, the more complex illumination changes are contained. An interesting feature to see is the cone like shape, stretching from left to right with the tip being close to the point (0,1). It seems that even if the multiplicative component does not vary, the additive component is used to correct illumination outliers. The large circles in the figure denote the means of the distributions. Many sequences are gathered around $(0.35, 1)$. The most complex sequence belonging to the MPI Sintel test suite is located at $(0.845, 1.457)$. Obviously, the MPI Sintel benchmark has more chances to contain a complex scene than the Middlebury benchmark, due to a large amount of image sequences. The comparison therefore was done on the means of the distributions. First of all, the differences to the origin were calculated for each mean, shown in the table below.

	KITTI 2015	KITTI 2012	MPI Sintel	Middlebury
Mean distances	1.061	1.021	1.074	1.037
Mean STDM-value	0.435	0.386	0.295	0.197

Table 3.1: Distances of the scatter-plot means to the origin and STDM-values of the mean, calculated with ground truth flow

Since the small values of the additive components are not as meaningful as the multiplicative component values, one should either do a weighting of the achieved standard deviation values or only take the multiplicative component into account. In this case only the multiplicative component values are regarded to obtain a complexity order of the benchmarks.

3.4.3 Preliminary Conclusion

Finally, we can say, that regarding illumination changes computed via ground truth, the KITTI 2015 test suite is the most complex one followed by the KITTI 2012 benchmark. Fewer illumination changes were found in the MPI Sintel and the Middlebury benchmark.

What remains to investigate is, if the same results are achieved, when approximating the coefficients with calculated flow instead of the ground truth flow.

The bottom sub-figure of Figure 3.12 presents the distributions of the deviations, when the coefficients are approximated with computed flow. Like before, a cone like structure can be observed in the plot. All test suites have sequences with almost no illumination change. Because all benchmarks have points located on the far left at a level of one of the standard deviation from zero. In contrast to the approximation with the ground truth, higher values of the standard deviations from one/zero are achieved. In particular, the KITTI 2012 and MPI

Sintel test suite differ the most from the calculations with the ground truth. This can be seen best in the scatter-plots shown in Figure 3.13. Note that the KITTI benchmarks now have a dense flow field, meaning, that the illumination change is calculated for a lot more pixels than in the case, when using the sparse ground truth flow. Additionally, the means are shifted, but not as much, such that the order of complexity stays the same.

	KITTI 2015	KITTI 2012	MPI Sintel	Middlebury
Ground truth flow	0.435	0.386	0.295	0.197
Computed flow	0.486	0.461	0.334	0.208

Table 3.2: STDM values of the mean.

Even though we have small deviations in the ranking values of the benchmarks, an overall statement can be made. The KITTI 2015 and KITTI 2012 benchmarks belong to a more complex category, with respect to illumination changes, than the MPI Sintel and Middlebury benchmark on average. Both scatter-plots have shown, that the most difficult sequences regarding illumination changes can be found in the MPI Sintel test suite. Next, we will leave the area of illumination changes and instead focus on the difficulty of large displacements.

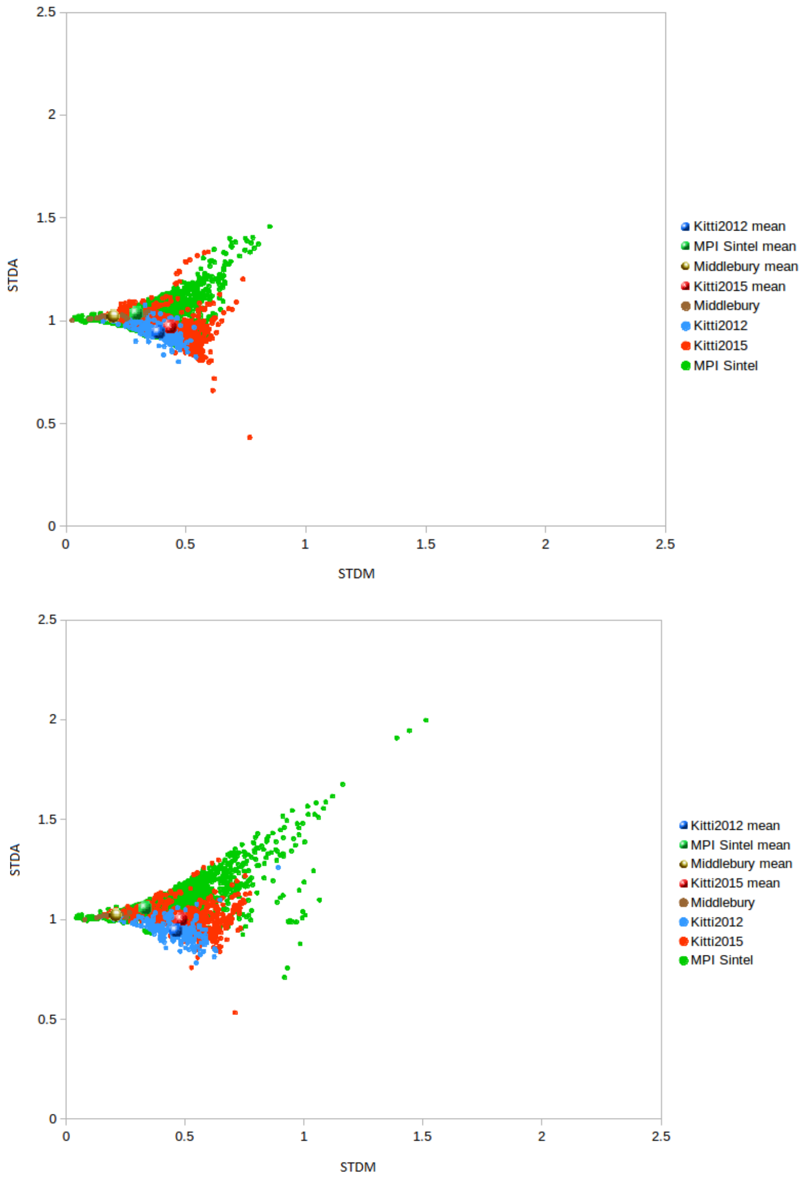
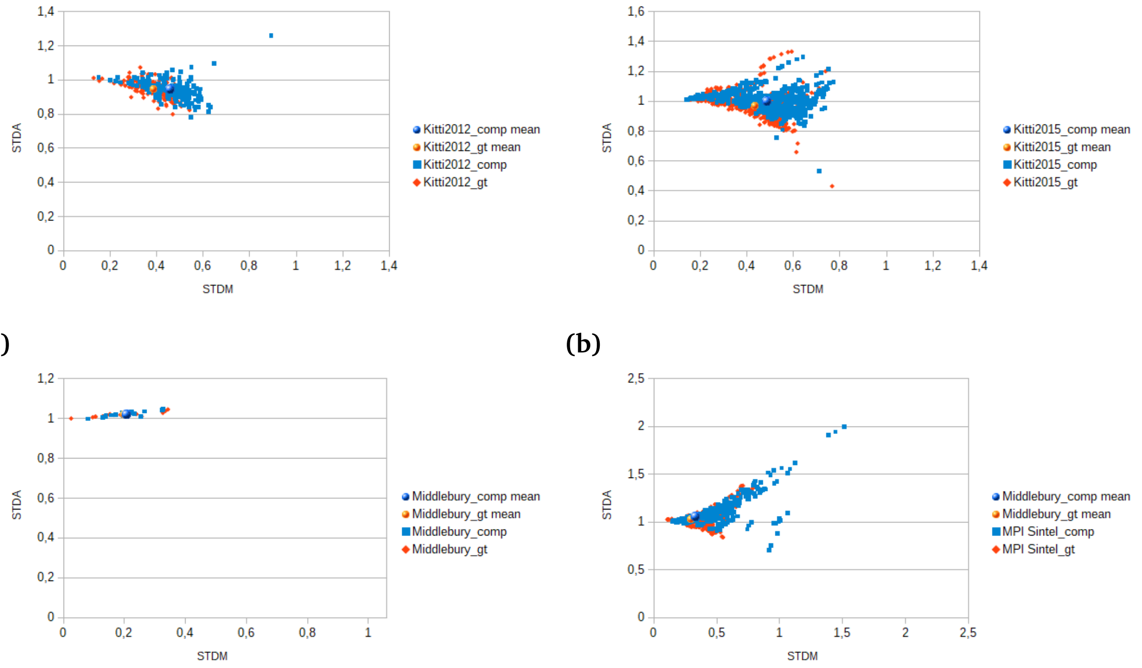


Figure 3.12: Scatter-plot of the STDA and STDM evaluation of the benchmarks. *Top:* Calculated with ground truth flow. *Bottom:* Calculated with computed flow (first order Zimmer et al.).



(a)

(b)

(c)

(d)

Figure 3.13: Separate scatter-plots of the STDA and STDM evaluation for each of the benchmarks calculated with ground truth flow and computed flow. *First row, from left to right:* (a) KITTI 2012. (b) KITTI 2015. *Second row, from left to right:* (c) Middelbury. (d) MPI Sintel.

4 Large Displacements

As known from literature and practice, one of the toughest challenges in optical flow estimation are large displacements. The first question, which comes to mind, when reading the key word large displacements is: “What does the author mean with large?” Five, ten or even a hundred pixels? In contrast to the analysis done by Hager, not only the magnitude of the each vector will be taken into account but also the size of the object it belongs to. This is done, because the size of the displacement should be considered relative to its size. Therefore, this chapter discusses the attempts, which were made, in order to extract the amount of large displacements for each of the aforementioned benchmarks. Two main tasks arise. The first, being the most difficult one, is the determination of the object sizes. The second, more simpler task, is the calculation of a movement/object-size ratio. Let us begin with the scale calculations.

4.1 Finding the Scale of a Pixel

As mentioned before, the first task is to calculate the sizes of moving objects in an image. Since the objects vary form cars in the KITTI test suite to dragons in the MPI Sintel benchmark, algorithms like the R-CNN designed for small object detection [CLTX16] are not an option, because there is no training data for all possible objects. In contrast to calculate the sizes of all objects, one simply tries to obtain a value for each pixel, which corresponds to the scale of the object it belongs to. Hence, three different approaches were tested. The first one uses differences of Gaussians. The second is a neighborhood-approach, using a fixed patch size to calculate the number of similar vectors in the flow field within the patch. The third and last one is a marching algorithm.

The task of calculating the scale-value for certain image features is not new. Familiar algorithms such as SIFT or SURF [Low99, BETVG08] use the Gaussian-scale-space, as explained in Chapter 2, in order to find key-points. However, these algorithms have one disadvantage. Due to the fact, that the interests lie on the detection of local maxima in scale and space and not on the maximum scale detection for all pixels, one cannot use these algorithms directly to calculate the scale of all pixels in the image. In addition, one rather wants to calculate the sizes of the objects on the flow instead of on the RGB-frames. This is done, because if lots of small objects are located near each other and move in the same direction with similar speed, they can be identified as one large object. An example is shown in Figure 4.1.

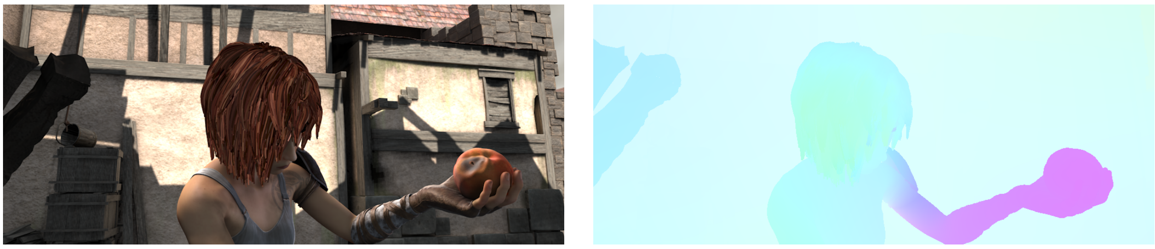


Figure 4.1: Frame 14 and the displacement field between frame 14 and 15 for a sequence of the MPI Sintel benchmark.

In this case the 14th frame as well as the displacement field between the 14th and 15th frame are displayed. Let us focus on the hand holding the apple. We know, that there are two objects in this region. The hand and the apple, each having a different size. What we also can observe is, that when the hand moves the apple moves as well, which means that we have a larger object consisting of both apple and hand. If we would use the RGB-frame we would obtain scales for the hand and the apple instead of the combined object. To encounter this, one can simply use the flow field for the size detection, since objects moving in the same direction have similar displacement vectors.

4.1.1 A Difference of Gaussians Approach

The first approach to obtain scale values from the flow field uses the differences of Gaussians, short DoG. The flow field, having the dimensions $n \times m$, is divided into two component arrays, each having the same dimension as the flow field. For each of these, a difference of Gaussians-pyramid is created with 7 octaves, 6 sub levels and without sub sampling. The pyramids are processed by running over each pixel and searching for the maximum value in the DoG-pyramid. The level containing the max value corresponds to the scale of the pixel. What remains is to combine the scale values for both arrays by taking the maximum scale value for each pixel. Although this approach sounds very reasonable, tests have shown, that some major drawbacks exist. For this reason the second neighborhood based-approach and the third marching approach are used later on.

First let us look at the DoG result when trying to obtain a scale value for a square of different sizes. For simplicity, the u components of the flow, movement in x-direction of the square is set to 255 and the v component, movement in y-direction, to zero. Figure 4.2 shows the different scales in the second column from the left, calculated using the DoG method for each of the squares on the left hand side. The standard deviation σ used to build the pyramids was set to 0.2. To increase the visibility, the image is plotted by taking the log of the scale value. Note the large blurred boundaries as well as the dark edges surrounding the square in the scale image.

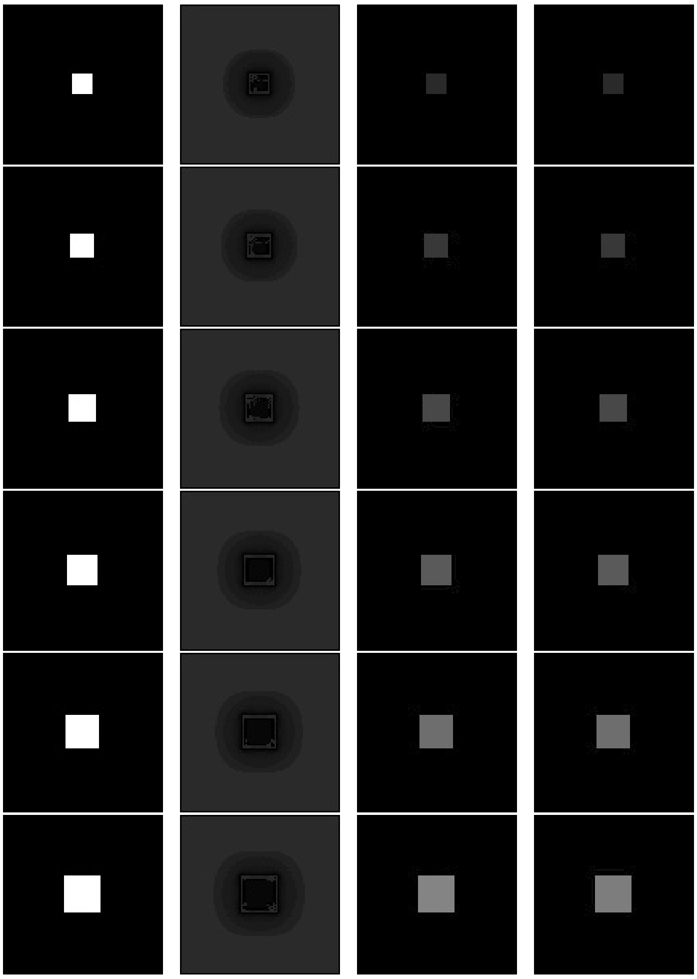


Figure 4.2: Scale evaluation on different sizes of squares using the difference of Gaussians, the neighborhood/patch approach and a marching approach. *First column, from top to bottom:* Different square sizes (12×12, 14×14,$\cdots$, 20×20, total image size $= 100 \times 100$). *Second column, from top to bottom:* Difference of Gaussians-scale image ($\sigma = 0.2$, number of octaves $= 7$, number of sub levels $= 6$, no sub sampling). *Third column, from top to bottom:* Patch-scale image (patch size $= 51 \times 51$, $m = 0.5$, $a = 20$). *Fourth column, from top to bottom:* Marching-scale image (neighborhood size $= 7 \times 7$, $m = 0.5$, $a = 20$).

Obviously, this is due to the fact, that the maximum difference at the pixel located directly next to the square is achieved in the first iteration of blurring. This means, that this pixel will always have the smallest scale, which is actually not true since the adjacent object, in this case the background, may be really large. Additionally, this blurring effect also influences the scale of the surrounding pixels as can be seen in the figure. The area of false scale values is so large, that the impact on the later calculated ratio would be too great. Also note, that the inner part of the square should be getting brighter and brighter. This change is very small, since the blurring kernel does not get large enough to reduce the image to its mean value. The larger the kernel, the larger is the region of outliers. The results of this approach did not improve, even by choosing different σ's, octave and sub-level sizes, computing with sub-sampling and using different scale adjustments. For this reason a second attempt was made to calculate the scale values using a completely different method.

4.1.2 A Patch Method for Scale Calculation

The second method used to calculate the scale values is neighborhood based. The fundamental idea is to use a patch of a certain size and count the number of vectors, which are similar to the central one. To calculate if two vectors $\boldsymbol{v_1}$ and $\boldsymbol{v_2}$ are similar, the following function is used:

$$\text{sim}(m, a, \boldsymbol{v_1}, \boldsymbol{v_2}) = \begin{cases} true & \text{if } ||\boldsymbol{v_1}| - |\boldsymbol{v_2}|| \leq m * |\boldsymbol{v_1}| \text{ and } \text{arctan2}(\boldsymbol{v_1}, \boldsymbol{v_2}) \leq a \\ false & \text{else} \end{cases} . \tag{4.1}$$

$m \in [0, \cdots, 1]$ denotes how much the magnitude of the vectors may vary and $a \in [0, \cdots, 180]$ denotes the maximum angle between the vectors in order to consider them to be similar. Next, the number of vectors in the patch, which are similar, are counted and divided by the patch size to get the scale value of a pixel. The improved results for the square images can be seen on the right hand side of Figure 4.2, where a patch of size 51×51 was used together with $m = 0.5$ and $a = 20$. In contrast to the DoG approach, sharp contours are achieved as well as a smooth increasing of the scale values. Note that the maximum scale value is dependent on the patch size. A small disadvantage of this approach is, that the scale is always smaller at the borders, if the object doesn't fit in quarter of the patch size.

4.1.3 A Marching Method for Scale Calculation

To counter the disadvantage of the patch approach of being dependent on the size of the patch, we will look at a alternative marching method. The idea of the marching method is similar to the one of the patch approach with the difference, that we keep track of the pixels which are similar in order to exceed the boundaries of the patch. In addition, a neighborhood size can be chosen to overcome the problematic of sparse flow fields.

We start off by labeling all positions of the flow as not visited and push them in a set. Then the following algorithm is performed:

Algorithm 4.1 Marching algorithm

```
while the set is not empty do
    push a position from the non labeled set to the stack;
    counter = 0;
    while the stack is not empty do
        current = pop a position from the stack;
        counter ++;
        mark it as visited;
        for all neighbors of current in a certain patch size do
            if not visited & are similar then
                push from the non labeled set to the stack;
            else
                do nothing;
            end if
        end for
    end while
    set the scale value for the positions to the counter value;
end while
```

The most right column of Figure 4.2 shows the results of this approach on the square example. Even though we gain the property of measuring larger objects than the patch size and reducing the run-time complexity to $O(1)$, we encounter the drawback of objects containing completely different vectors. For example: A horizontal line of 1000 vectors, facing in the same direction, begins with a tiny vector on the left hand side and ends with a very long vector on the right. The difference in magnitude of an adjacent vector still suffices the similarity condition. This would mean, that we obtain a large scale for all positions of the line and therefore a wide variety of movement-size ratios. This then leads to the misinterpretation, that only a part of the object is a large displacement and the other part not. For the later evaluation both the patch and the marching approach are used. Now, that we are able to determine the scale values, we can move forward to determine the movement-size ratio.

4.2 Calculating the Ratio Between Movement and Scale

To determine the movement-size ratio for each pixel, used to determine large displacements, one simply divides the flow vector by the scale value for each position. Figure 4.3 shows the

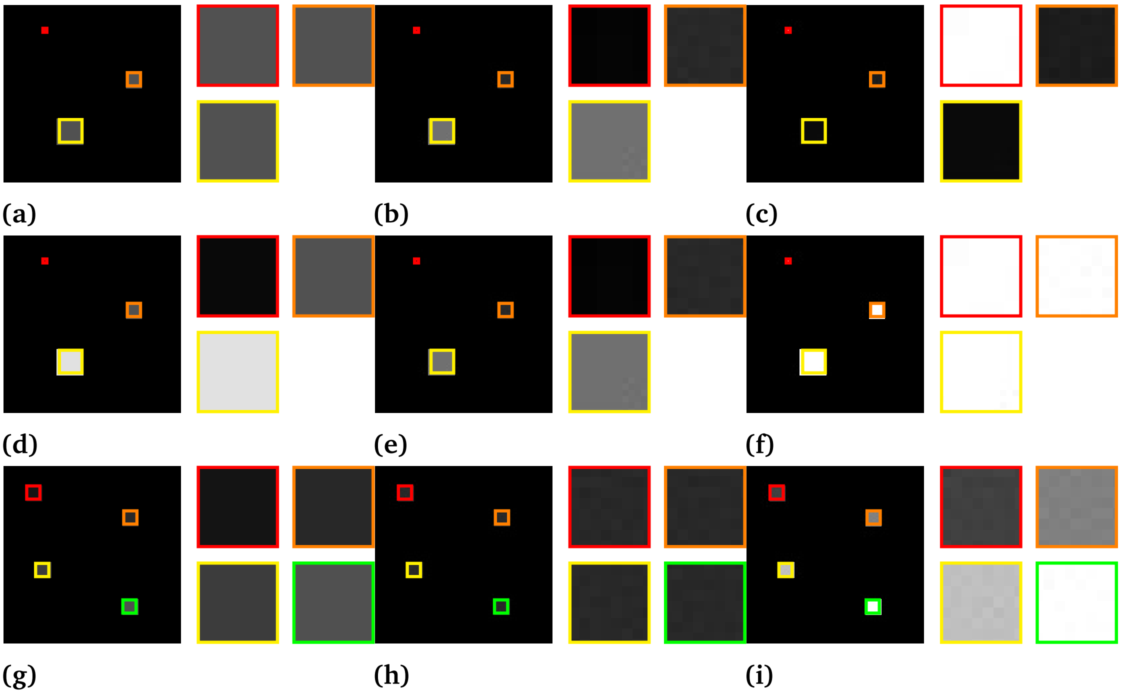


Figure 4.3: Validation of the ratio calculation on different movement speeds and square sizes. The parameters are set to $m = 0.5$. $a = 20$. *First row, from left to right:* (a) Flow field magnitude of three squares having the different sizes (3×3, 9×9 ,15×15) and each having the same flow speed of 81 in x direction. (b) Extracted normalized scale. (c) Calculated and re-scaled ratio image. *Second row, from left to right:* (d) Flow field magnitude of three squares having different sizes (3×3, 9×9,15×15) and each having a different flow speed (9,81 and 225 in x direction). (e) Extracted normalized scale. (f) Calculated and re-scaled ratio image. *Third row, from left to right:* (g) Flow field magnitude of four squares each having the same size (9×9), but different speeds (20 (red), 40 (orange), 60 (yellow) and 80 (green) in x direction). (h) Extracted normalized scale. (i) Calculated and re-scaled ratio image.

evaluation of the approach of the calculated scales and ratios for different movement speeds and sizes of squares.

The first row in the figure demonstrates the impact of the size of an object on the ratio, if all objects are traveling at the same speed. The brighter the object is in the last column, the larger is the displacement relative to the size. The second row shows the correct behavior of the method, when all objects move at the speed of their own size. In this case all squares are equally colored as one would expect. The third row depicts, what happens to the ratio, if the size stays the same but the velocities differ. Again the outcome is as expected. The fastest one is the brightest followed by the slower ones in descending order of the brightness.

Note for the recognition of the scale via the patch approach, the size of pixel can only be as large as the size of the patch and may lead to misinterpretation, if the patch is chosen too small. Tests have shown, that a reasonable size for a patch lies at 51×51, with an acceptable computation duration. What remains is to find a suitable threshold for the detection of large displacements.

For simplicity it is assumed, that the objects lie within a bounding square. In order to prevent long thin objects to be approximated by a large bounding square, the area of the square is set to the area of the object/pixel-count. Given the shape of a square, it is obvious, that the threshold can be designed in terms of multiples of the edge length of an approximated square. For the later analysis we will use the following formula for the large displacement evaluation:

$$\mathrm{ld}(r, sc, k) = \begin{cases} true & \text{if } r \geq \frac{k}{sqrt(sc)} \quad k \in \mathbb{Z} \\ false & \text{else} \end{cases}, \tag{4.2}$$

with r being the ratio, sc the scale value and k the parameter for the multiples of the edge length. Now that all things are set, let us move forward to the evaluation of the different benchmarks.

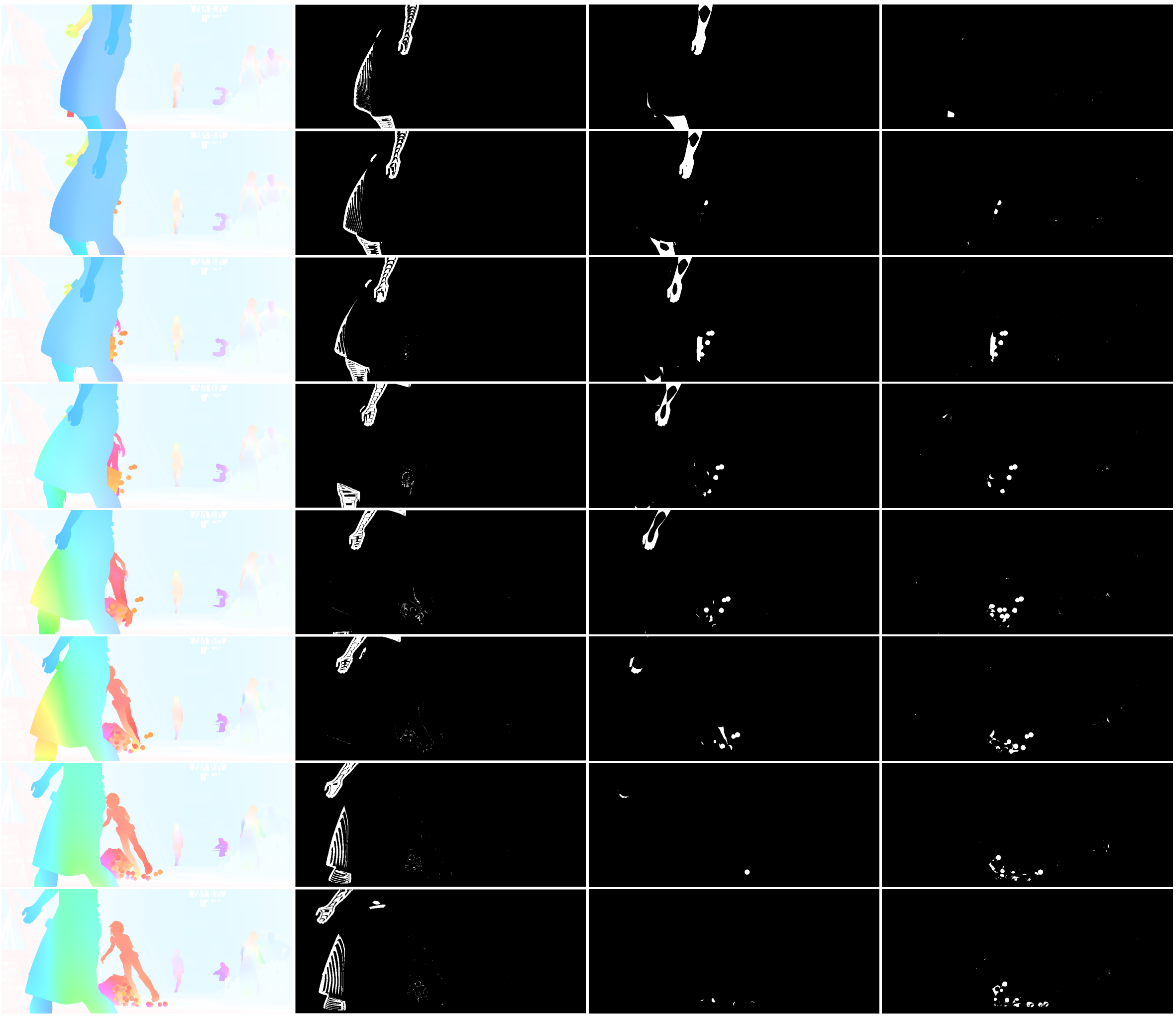


Figure 4.4: Large displacement evaluation of the Market2 scene of the MPI Sintel benchmark (frame 8 to 15), with the movement being greater than $\sqrt{\text{scale value}}$. *First column, from top to bottom:* Ground truth flow. *Second column, from top to bottom:* Large displacements using DoG with scale value adaption (scale value times 20). The threshold cannot be interpreted in the same way as when using the patch or marching approach. *Third column, from top to bottom:* Large displacement detection using the patch approach (patch size $= 51 \times 51$, $m = 0.5$, $a = 10$). *Fourth column, from top to bottom:* Large displacement detection using the marching approach (neighborhood size $= 7 \times 7$, $m = 0.1$, $a = 5$).

4.3 Large Displacement Evaluation

The previous section introduced the concepts of identifying the scale of objects within a scene as well as the definition of large displacements. In the following, the threshold for the large displacements was calculated with the parameter k being set to one. This means, a large displacement is measured, if the magnitude of the vector is greater or equal to the square root of the scale of the object to which the vector belongs.

4.3.1 A First Attempt

To verify, if one is a good value for k, a first attempt to measure large displacements was made using the Market2 scene of the MPI Sintel benchmark. This scene was chosen because of the variety of moving objects. Especially the crate containing the apples is of interest, since it is accelerated quickly causing the small apples to fall out with a great speed, making it a good scenario to look for large displacements. Figure 4.4 shows the evaluation of this first attempt. It is expected, that the apples, which have small scale and are falling down very fast, are recognized as large displacements, in contrast to the person walking in the foreground. Let us take a closer look at the results obtained with the patch approach.

As we can see in the middle column of the figure, the apples are clearly recognized as large displacements, but also the limbs of the person walking in the foreground. The back and forth movement of the hand is recognized as a large displacement, since it is regarded as a separate object. The smaller movements in the background, especially the person picking up the barrel, are discarded as expected.

Comparing the results of the patch approach to the ones achieved with the marching approach, one can clearly see, that the limbs of the person in the foreground are not recognized as well as the leg of the person tripping over the apples. In contrast to the patch method some large displacements of the size of a few pixels are also detected in the background. For example on the right hand side of the person picking up the barrel.

Despite the fact that one could now tune the parameter k of the threshold to only recognize the apples, it is left at the value of one for the evaluation of the other benchmarks.

4.3.2 Comparison Between the Benchmarks Using the Patch Approach

The following tables present the amount of large displacements in percent. The first table, Table 4.1, shows the results achieved using ground truth flow and the second, Table 4.2, the results using computed flow. Note that the values from row to row do not vary. This could have several reasons with the first being, that the patch is not large enough to capture vectors, were the angle deviates more that 5 degrees and still fulfills the magnitude condition.

Angel (a)/ Magnitude (m)		50%	30%	10%
20 °	K12:	36.03	36.53	45.72
	K15:	39.94	41.37	**50.61**
	Mid:	00.00	00.00	00.02
	Sin:	07.97	08.17	09.10
10 °	K12:	36.03	36.53	45.72
	K15:	39.94	41.37	**50.61**
	Mid:	00.00	00.00	00.02
	Sin:	07.97	08.17	09.10
5 °	K12:	36.03	36.53	45.72
	K15:	39.94	41.37	**50.61**
	Mid:	00.00	00.00	00.02
	Sin:	07.97	08.17	09.10

Table 4.1: Amount of large displacements ($k = 1$, patch size $= 51 \times 51$) found in the benchmarks using *ground truth flow*, the patch approach and different similarity parameter values.

Angel (a)/ Magnitude (m)		50%	30%	10%
20 °	K12:	25.47	25.91	**32.68**
	K15:	19.86	20.71	27.87
	Mid:	00.00	00.00	00.15
	Sin:	07.12	07.22	08.07
10 °	K12:	25.47	25.91	**32.68**
	K15:	19.86	20.71	27.87
	Mid:	00.00	00.00	00.15
	Sin:	07.12	07.22	08.07
5 °	K12:	25.47	25.91	**32.68**
	K15:	19.86	20.71	27.87
	Mid:	00.00	00.00	00.15
	Sin:	07.12	07.22	08.07

Table 4.2: Amount of large displacements ($k = 1$, patch size $= 51 \times 51$) found in the benchmarks using *computed flow*, the patch approach and different similarity parameter values.

Another reason could be, that the difference of the magnitude of the vectors of the objects and the background is so big, that the angle does not play a role anymore.

Looking at both tables column wise, an increase of the amount of large displacements in all benchmarks can be seen. This is due to the decrease of the scale value for each vector, since fewer similar vectors are found within the patch of each vector. The side effect of this observation is, that it can be said, that a lot of affine movements take place in those benchmarks, were the values differ a lot from column to column. This can be seen in particular in the last row of the first table. Here, the angle may only differ by five degrees, meaning that the movement described by the similar vectors goes in the same direction. Therefore, if the similar vectors of the patch move in the same direction and their number increases the more deviation from the magnitude is allowed, implies, that these still belong to the same object with non constant movement. This can be seen best in the cases of the KITTI 2012 and 2015 test suite.

Comparing the amount of large displacements in the last column within each benchmark, measured with ground truth flow, leads to the following order. First place goes to the KITTI 2015 benchmark, with an average amount of about 50 percent of large displacements, closely followed by the KITTI 2012 test suite. The last places go to the MPI Sintel benchmark, with about nine percent, and the Middlebury benchmark with 0.02 percent. The large difference of the first two and last two places can be again explained by the sparse ground truth flow fields of both KITTI benchmarks and by the fast camera movement when driving. Figure 4.5 of the KITTI 2015 test suite exemplary depicts the large displacements found in both KITTI benchmarks.

The most amount of large displacements is found on the sides of the frames due to the fast driving along the road. One can also make out the boundary, at which the vector magnitude suffices to exceed the threshold. This type of fast camera movement is not found in the other two benchmarks. Only few large displacements are found in the Middlebury benchmark, while

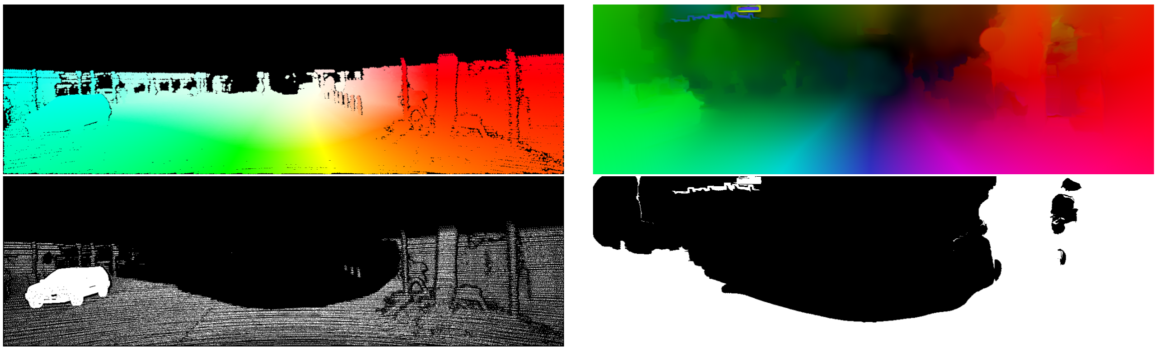


Figure 4.5: Visualization of the used flows and the large displacements found in Frame 0 of the KITTI 2015 benchmark. *First row, from left to right:* Ground truth flow and computed flow. *Second row, from left to right:* Large displacements found using the ground truth flow and computed flow.(patch size $= 51 \times 51$, $k = 1, m = 10, a = 5$).

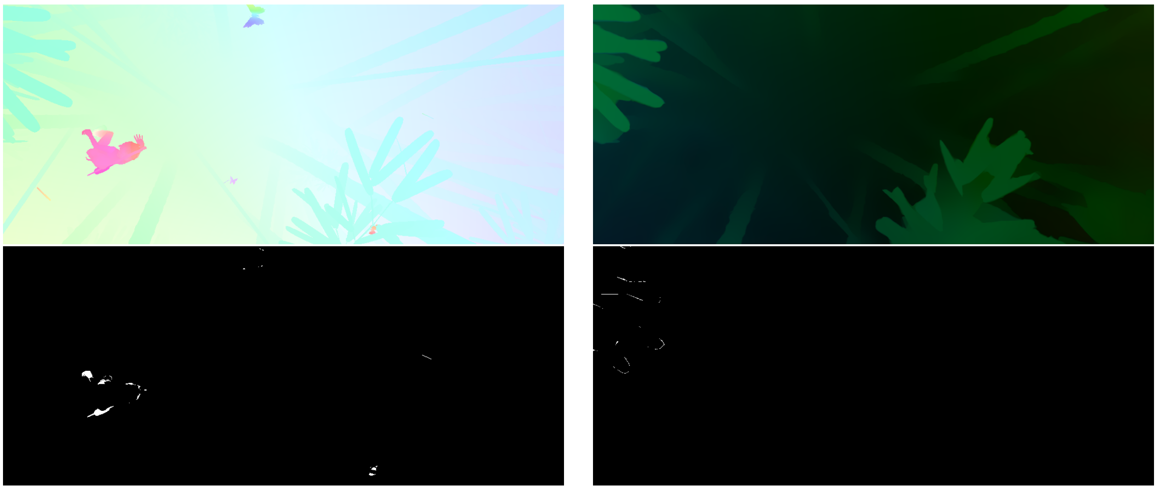


Figure 4.6: Visualization of the used flows and the large displacements found in Frame 19 of the Bamboo1 scene of the MPI Sintel test suite. *First row, from left to right:* Ground truth flow and computed flow. *Second row, from left to right:* Large displacements found using the ground truth flow and computed flow.(patch size $= 51 \times 51$, $k = 1, m = 10, a = 5$).

smaller amounts of large displacements are found in the MPI Sintel test suite, as shown in Figure 4.4. The reason lies in the small movements of the camera around or through the almost static scenes of the Middlebury benchmark.

A similar order is obtained, when using computed flow to measure the amount of large diplacements, with the difference that the KITTI 2012 and 2015 test suit switch places. Additionally, we can observe, that the values of the KITTI benchmarks are a lot lower in all parameter constellations. This shows the impact of the sparse ground truth flow fields on the scale values and therefore also on the large displacement values of both KITTI test suites. The reduction of the MPI Sintel values can be explained by looking at the following Figure 4.6 of the Bamboo1 scene. The person walking through the shrubs can be clearly seen in the ground truth flow but not in the computed flow. Only the leaves in the foreground are precisely visible in the computed flow field. Therefore, it can be said, that the used method to compute the flow may not be the best in recognizing the movement of small objects, which are in addition partially occluded. Similar observations were made in the Ambush and Market scenes, just to name a few. Looking at the values of the Middlebury benchmark, one can say, that the expectations were met by not having any large displacements in the scenes.

To show the impact of the patch size on the KITTI benchmarks one computation was made with a patch of size 151×151 on the KITTI 2012 benchmark, using computed flow. The amount of large displacements found using the larger patch, with the parameters of the similarity function set to 10 percent and 5 degrees, resulted in 24 percent instead of 45 percent. The larger the patch size gets, the less large displacements are found along the sides of the KITTI benchmarks.

This effect is illustrated in Figure 4.7. The reason for only doing one computation with a patch of this big size was the long computation time.

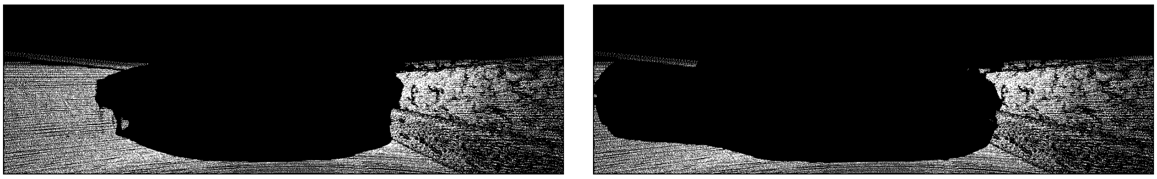


Figure 4.7: Visualization of the large displacements found in Frame 0 of the KITTI 2012 benchmark. *From left to right:* Calculation with a patch of size 51×51 and of 151×151 ($k = 1, m = 10, a = 5$).

4.3.3 Comparison Between the Benchmarks Using the Marching Approach

The following measurements with the marching approach were done using a neighborhood size of 7×7 to overcome the missing vectors in the KITTI ground truth flows. Since only a small neighborhood is taken into account, the parameters of the similarity function were set to $a = 5$ and $m = 0.1$. It is expected, that the KITTI benchmarks contain a lot less large displacements in comparison when measured with the patch ansatz.

Method/ Benchmark	KITTI 2015	KITTI 2012	MPI Sintel	Middlebury
Patch	**50.6 %**	45.7 %	9.10 %	0.02 %
Marching	**5.82 %**	2.51 %	0.33 %	0.07 %

Table 4.3: Amount of large displacements ($k = 1$, patch size $= 51 \times 51$, neighborhood size $= 7 \times 7$, $a = 5$, $m = 0.1$) found in the benchmarks using ground truth flow.

Method/ Benchmark	KITTI 2015	KITTI 2012	MPI Sintel	Middlebury
Patch	27.8 %	**32.6 %**	8.07 %	0.15 %
Marching	0.44 %	0.09 %	0.20%	**0.47 %**

Table 4.4: Amount of large displacements ($k = 1$, patch size $= 51 \times 51$, neighborhood size $= 7 \times 7$, $a = 5$, $m = 0.1$) found in the benchmarks using computed flow.

Obviously, the Middlebury benchmark achieves the lowest value of 0.07% when using ground truth flow for the evaluation. This is due to the fact, that the majority of the scenes are obtained by photographing a static scene from a different angle. Therefore these scenes do not contain large displacements.

Looking at Table 4.4, one can see, that the Middelbury benchmark apparently contains the highest amount of large displacements, when evaluated using computed flow compared to the

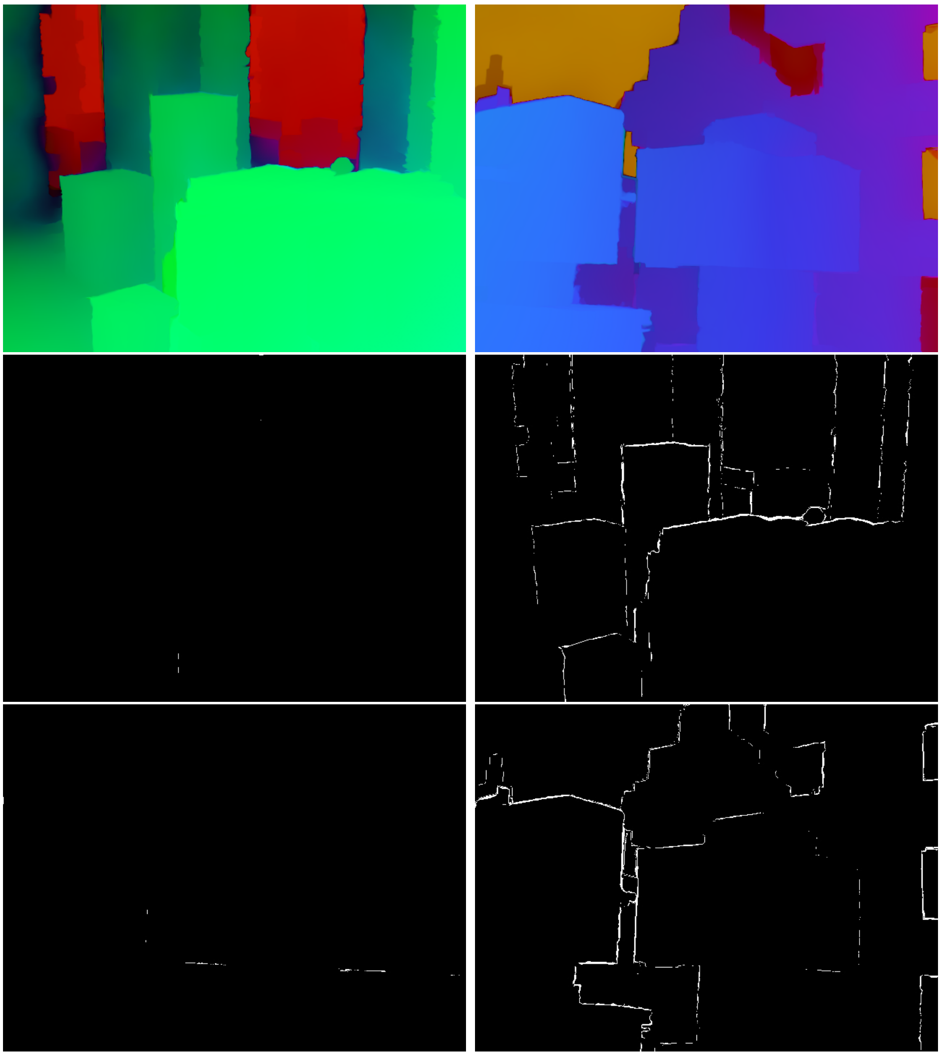


Figure 4.8: Visualization of the large displacements found in the Urban2 and Urban3 scene of the Middlebury benchmark. *First row from left to right:* Computed flow of the Urban2 and Urban3 sequence. *Second and third row from left to right:* Using ground truth and computed flow.

evaluation with ground truth flow. This increase is explained by the Figure 4.8 depicting the large displacements found in the Urban2 and Urban3 sequence.

Note the large displacements on the edges of the objects. This behavior results form the outliers of the computed flow at the edges.

The KITTI benchmarks, as expected, achieve a lot lower values than when using the patch approach. This is exemplary shown in Figure 4.9 depicting the displacement maps of the first and second frame of the KITTI 2015 benchmark. The right image of the figure also shows the

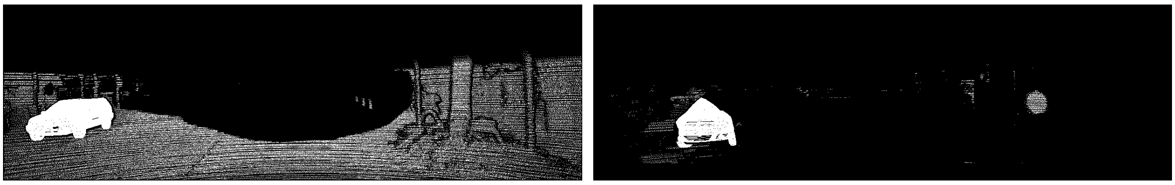


Figure 4.9: Visualization of the large displacements using ground truth flow found between the first frame and second frame of the KITTI 2015 benchmark. *From left to right:* Patch and marching approach.

disadvantage of the marching algorithm. The car on the left hand side of the large displacement map is split into the front part being a large displacement and a the back part, which is not recognized as a large displacement even though both parts belong to the same physical object. However, the main goal is achieved by the marching approach by giving the streets and houses a large scale such that these are not considered as large displacements. In comparison to the other benchmarks the KITTI 2015 test suite achieves the highest amount of large displacements when using ground truth flow and the second highest amount using computed flow. This decrease from ground truth to computed flow is explained by the pixels taken into account for the statistics. Since about 20% of the total pixels, most of them in the sky region of the ground truth, are not taken into account due to not containing flow information, it is obvious that this results in a higher average amount of large displacements. In contrast, the statistics of the computed flow takes these 20% of the pixels with small to no movement into account and therefore achieves a lower average large displacement value.

The MPI Sintel benchmark contains a moderate amount of large displacements. While the other benchmarks vary extremely in the values from ground truth to computed flow, the MPI Sintel values almost stay the same. These small values occur since most of the time the sequences include two larger objects, which interact in the foreground and smaller objects in the background, which do not move a lot. Figure 4.10 shows some sequences where this is the case.

4.3.4 Intermediate Conclusion

As an intermediate conclusion it can be said, that regarding large displacements the KITTI test suites are the most complex ones according to the measurements made. But one should keep in mind, that this is due to the affine movement of large obstacles passing by the car at a very high speed. The MPI Sintel test suite has fewer but more intuitive large displacements as seen in Figure 4.4. The Middlebury benchmark has the least amount of large displacements. The most of them are caused by motion discontinuities or occlusion. The next chapter deals with an in depth analysis of the different movement types, constant and affine, which we have already come across during the analysis of large displacements.

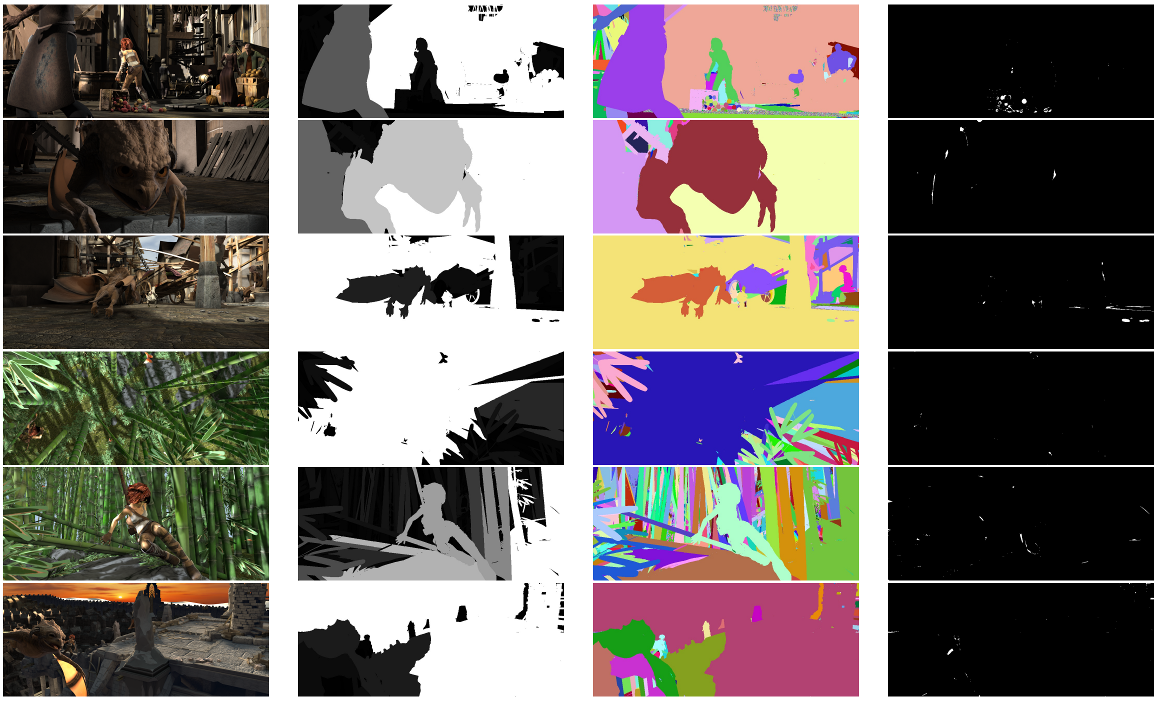


Figure 4.10: Visualization of some large displacements using ground truth flow found in the MPI Sintel benchmark. *From left to right:* First frame, scale-image, segment-image and large displacement-image. *Form top to bottom:* Market2 scene Frame 19, Market5 scene Frame 5, Market6 scene Frame 2, Bamboo1 scene Frame 6, Bamboo2 scene Frame 42 and Temple2 scene Frame 17.

5 Movement Categorization

The last topic covered in this thesis is a comparison between the categorization of movement in first or second order made by Hager and the categorization made by the order-adaptive regularization for variational optical flow by Maurer et al.[MSB17]. Since both types are the most common ones in the majority of optical flow problems, higher order movements are not considered. As already explained in the foundations chapter 2, first order movement takes place, when the flow describing the movement is constant within a region. In contrast, second order movement is present, if the flow within a region is affine. For example if the object moves towards the camera. First let us start by recapitulating and comparing the methods mathematically to show that both models are ideal to compare.

5.1 Review of the Movement Categorization Models of Hager

As mentioned before the goal is to categorize the vectors or the complete flow fields into first or second order movement. In the case of local movement categorization, the category depends on the surrounding neighborhood. Therefore it is essential to have dense flow fields. Hager handled the challenge of the sparse flow fields, found in the KITTI 2012/2015 and Middelbury benchmarks, by reconstructing the flow using the filling in effect of two different variational methods. One using a first order smoothness term and the other a second order smoothness term. The two reconstructed flows are then evaluated for a global and a local categorization. So let us go more into detail on the reconstruction.

First, we start off with the traditional framework for optical flow computation or in this case inpainting:

$$E(u,v) = D(u,v) + \alpha \cdot R(u,v) \ \ , \tag{5.1}$$

consisting of the data term D and the weighted regularization term R, with the weighting parameter α. Since the reconstructed flow should be as similar to the ground truth flow (u_{gt}, v_{gt}) as possible, Hager chooses the data term to be:

$$D(u,v) = \int_\Omega \Psi((u^t - u^t_{gt})^2) + \Psi((v^t_{i,j} - v^t_{gt})^2) \ \ dxdy \ \ , \tag{5.2}$$

with $\Psi = 2 \cdot \sqrt{s^2 + \lambda^2}$ being the sub-quadratic penaliser function like in the illumination chapter. The regularization term is chosen according to the analyzed movement, which will be shown later. As shown in the illumination chapter the functional is solved by setting up the Euler-Lagrange equations. Afterwards, these equations are reformulated according to the Jacobi iterative scheme and then solved using the lagged non linearity method. For further details on how the iterative schemes are set up please see [Hag17] Chapter 4.4 Movement.

5.1.1 Global Constant and Affine Model

As seen in the illumination chapter, different parametrisations of the brightnes transfer functions like constant or affine are possible. Same holds for the flow (u, v). To analyze the movement types globally, Hager compares the weights A_1 and B_1 with the ones of the affine movement. If A_1 and A_2 of the affine model are close to zero and A_3 close to A_1 of the constant model, the movement is globally constant. Otherwise, it is globally affine. The constant and affine parametrisation using the weights A_* and B_*reads:

$$\begin{aligned} \text{constant: } \begin{pmatrix} u \\ v \end{pmatrix} &= \begin{pmatrix} A_1 \\ B_1 \end{pmatrix} = \begin{pmatrix} u^t \\ v^t \end{pmatrix} \ , \\ \text{affine: } \begin{pmatrix} u \\ v \end{pmatrix} &= \begin{pmatrix} A_1 \cdot \hat{x} + A_2 \cdot \hat{y} + A_3 \\ B_1 \cdot \hat{x} + B_2 \cdot \hat{y} + B_3 \end{pmatrix} = \begin{pmatrix} u^t \\ v^t \end{pmatrix} \ , \end{aligned} \tag{5.3}$$

where $\hat{x} = \rho(x - x_0)$ and $\hat{y} = \rho(y - y_0)$ with $\rho = 1$ and x_0 being half of the image width and y_0 half of the image height. Using Equation 5.2 and 5.1 and setting the regularization term to zero for global flow calculation, results in:

$$\begin{aligned} \text{constant: } E(A_1, B_1) &= \int_\Omega \Psi(A_1 - u_{gt}^t)^2) + \Psi((B_1 - v_{gt}^t)^2) \ dxdy \\ \text{affine: } E(A_1, A_2, A_3, B_1, B_2, B_3) &= \int_\Omega \Psi(A_1 \cdot \hat{x} + A_2 \cdot \hat{y} + A_3 - u_{gt}^t)^2) \\ &+ \ \Psi((B_1 \cdot \hat{x} + B_2 \cdot \hat{y} + B_3 - v_{gt}^t)^2) \ dxdy \ , \end{aligned} \tag{5.4}$$

5.1.2 Local Constant and Affine Model

For the local movement types it is required to reconstruct the flows in order to obtain dense flow fields for the later categorization. Hager therefore reconstructs two flow fields. The first flow is reconstructed using a first order smoothness term penalized with the sub-quadratic penalizer-function from before:

$$R_{\text{const}}(u,v) = \int_\Omega \Psi(|\nabla u^t|^2) + \Psi(|\nabla v^t|^2) \;\; dxdy \;\; , \tag{5.5}$$

Using the data term from Equation 5.2 with the additional γ-function from before, which turns off the data term, where no ground truth flow is available and plugging it in the framework results in:

$$E(u,v) = \int_\Omega \gamma(\Psi((u^t - u^t_{gt})^2) + \Psi((v^t - v^t_{gt})^2)) + \alpha \cdot (\Psi(|\nabla u^t|^2) + \Psi(|\nabla v^t|^2)) \;\; dxdy \;\; . \tag{5.6}$$

For the second affine flow reconstruction, Hager chose to use the robust Frobenius norm of the Hessian of the u and v components:

$$R_{\text{affine}}(u,v) = \int_\Omega \Psi(||H(u^t)||^2_F) + \Psi(||H(v^t)||^2_F) \;\; dxdy \;\; , \tag{5.7}$$

leading to the following energy functional for reconstruction:

$$E(u,v) = \int_\Omega \gamma(\Psi((u^t - u^t_{gt})^2) + \Psi((v^t - v^t_{gt})^2)) + \alpha \cdot (\Psi(||H(u^t)||^2_F) + \psi(||H(v^t)||^2_F)) \;\; dxdy \;\; . \tag{5.8}$$

Once the flows are reconstructed two thresholds, $T_{\nabla a}$ and $T_{H(a)}$ are used, with $a \in \{u,v\}$, to distinguish between first order, second order and higher order movement. The first threshold applied on the gradients of the flow field separates constant movement from higher order movement. The second threshold separates movement lower or equal to second order from higher order movements. Hence, three cases for each direction u and v are obtained.

- $|\nabla a|^2 \leq T_{\nabla a} \rightarrow$ a is of type first order/ constant movement.
- $|\nabla a|^2 > T_{\nabla a}$ and $||H(a)||^2_F \leq T_{H(a)} \rightarrow$ a is of type second order/ affine movement.
- $|\nabla a|^2 > T_{\nabla a}$ and $||H(a)||^2_F > T_{H(a)} \rightarrow$ a is of type higher than second order movement.

Hager uses half of the median of the gradient or hessian values as the respective threshold.

5.2 Review of the Order-Adaptive Regularisation for Variational Optical Flow

The work of Maurer et al. [MSB17] discusses a variational approach for optic flow calculation with automatic regularization order determination. The comparison of both the results of Maurers and the results of Hagers approach is of interest. Before comparing the approaches, the details on how the order is determined automatically in the energy functional of Maurer et al. is discussed, starting with the traditional framework for optical flow computation:

$$E(u,v) = D(u,v) + \alpha \cdot R(u,v) \ , \tag{5.9}$$

consisting of the data term D and the weighted regularization term R. The assumptions used in the data term are the brightness and gradient constancy assumption. More interesting is the modeling of the regularization. Maurer et al. introduce the combining of two smoothness terms: One of first order and the other of second order, with a weighting parameter c. The choice for the first order smoothness term fell on the anisotropic complementary regulariser of Zimmer et al. [ZBW11]

$$R_1(u,v) = \int_\Omega S_1(u,v) \ dxdy = \int_\Omega \sum_{l=1}^{2} \Psi_l((\boldsymbol{r_l}^\top \nabla u)^2 + (\boldsymbol{r_l}^\top \nabla v)^2) \ dxdy \ . \tag{5.10}$$

$\boldsymbol{r_1}(x,y)$ and $\boldsymbol{r_2}(x,y)$, with $(x,y)^\top \in \Omega$, are two orthonormal eigenvectors of the regularisation tensor [ZBW11]. The penalisation function $\Psi_1(s^2)$ is chosen to be the edge-enhancing Perona-Malik penaliser [PM90] and $\Psi_2(s^2)$ the edge-preserving Charbonnier penaliser [CBFAB97]:

$$\begin{aligned} \Psi_1(s^2) &= \epsilon^2 \log(1 + s^2/\epsilon^2) \\ \Psi_2(s^2) &= 2\epsilon^2 \sqrt{1 + s^2/\epsilon^2} \ , \end{aligned} \tag{5.11}$$

with $\epsilon > 0$. As the second order smoothness term, the anisotropic coupling model of Hafner et al. [HSW15] is chosen:

$$\begin{aligned} R_2(u,v) &= \int_\Omega \inf_{a,b} \{S_2(u,v,\boldsymbol{a},\boldsymbol{b}) + \beta \cdot S_{\text{aux}}(\boldsymbol{a},\boldsymbol{b})\} \ dxdy \\ S_2(u,v,\boldsymbol{a},\boldsymbol{b}) &= \sum_{l=1}^{2} \Psi_l((\boldsymbol{r_l}^\top(\nabla u - \boldsymbol{a}))^2 + (\boldsymbol{r_l}^\top(\nabla v - \boldsymbol{b}))^2) \\ S_{\text{aux}}(\boldsymbol{a},\boldsymbol{b}) &= \sum_{l=1}^{2} \Psi_l\Big(\sum_{k=1}^{2} (\boldsymbol{r_k}^\top \mathcal{J} a \boldsymbol{r_l})^2 + (\boldsymbol{r_k}^\top \mathcal{J} b \boldsymbol{r_l})^2\Big) \ . \end{aligned} \tag{5.12}$$

The model consists of the coupling term S_2, connecting the gradients of u and v to the auxiliary functions $\boldsymbol{a}(x,y)$ and $\boldsymbol{b}(x,y)$. The smoothness term S_{aux} enforces the smoothness on $\boldsymbol{a}$ and $\boldsymbol{b}$. $\mathcal{J}a$ and $\mathcal{J}b$ denote the Jacobian with respect to a and b. This model is chosen, because the energy of both terms is ideal to compare, due to the similarity between the first and second-order model. Another benefit is, that if the auxiliary functions approach the values of the gradients of u and v, $\boldsymbol{a} = \nabla u$ and $\boldsymbol{b} = \nabla v$, the following regularization term for $R_2(u, v)$ is obtained:

$$\begin{aligned} R_2(u,v) = \int_\Omega & \underbrace{\sum_{l=1}^{2} \Psi_l\big((\boldsymbol{r_l}^\top(\nabla u - \nabla u))^2 + (\boldsymbol{r_l}^\top(\nabla v - \nabla v))^2\big)}_{=0} \\ & + \beta \cdot \sum_{l=1}^{2} \Psi_l \Big(\sum_{k=1}^{2} (\boldsymbol{r_k}^\top \mathcal{J}(\nabla u)\boldsymbol{r_l})^2 + (\boldsymbol{r_k}^\top \mathcal{J}(\nabla v)\boldsymbol{r_l})^2 \Big) \; dxdy \end{aligned} \tag{5.13}$$

Since the energy of S_2 is zero, this can be simplified and rewritten to:

$$\begin{aligned} R_2(u,v) &= \int_\Omega \beta \cdot \sum_{l=1}^{2} \Psi_l \Big(\sum_{k=1}^{2} (\boldsymbol{r_k}^\top H(u)\boldsymbol{r_l})^2 + (\boldsymbol{r_k}^\top H(v)\boldsymbol{r_l})^2 \Big) \; dxdy \\ &= \int_\Omega \beta \cdot \Big(\Psi_1 \Big(\sum_{k=1}^{2} (\boldsymbol{r_k}^\top H(u)\boldsymbol{r_1})^2 + (\boldsymbol{r_k}^\top H(v)\boldsymbol{r_1})^2 \Big) \\ &+ \Psi_2 \Big(\sum_{k=1}^{2} (\boldsymbol{r_k}^\top H(u)\boldsymbol{r_2})^2 + (\boldsymbol{r_k}^\top H(v)\boldsymbol{r_2})^2 \Big) \Big) \; dxdy \\ &= \int_\Omega \beta \cdot \Big(\Psi_1 \Big((\boldsymbol{r_1}^\top H(u)\boldsymbol{r_1})^2 + (\boldsymbol{r_1}^\top H(v)\boldsymbol{r_1})^2 + (\boldsymbol{r_2}^\top H(u)\boldsymbol{r_1})^2 + (\boldsymbol{r_2}^\top H(v)\boldsymbol{r_1})^2 \Big) \\ &+ \Psi_2 \Big((\boldsymbol{r_1}^\top H(u)\boldsymbol{r_2})^2 + (\boldsymbol{r_1}^\top H(v)\boldsymbol{r_2})^2 + (\boldsymbol{r_2}^\top H(u)\boldsymbol{r_2})^2 + (\boldsymbol{r_2}^\top H(v)\boldsymbol{r_2})^2 \Big) \Big) \; dxdy \end{aligned} \tag{5.14}$$

If no penaliser is used, $\Psi(s^2) = s^2$, the equation can be rewritten as:

$$R_2(u,v) = \int_\Omega \beta \cdot \Big(u_{xx}^2 + v_{xx}^2 + u_{xy}^2 + v_{xy}^2 + u_{xy}^2 + v_{xy}^2 + u_{yy}^2 + v_{yy}^2 \Big) \; dxdy \;\; , \tag{5.15}$$

since $\boldsymbol{r_1}$ and $\boldsymbol{r_2}$ form and orthonormal basis. Comparing this regularisation term to the affine regulariser of Hager also with $\Psi(s^2) = s^2$:

$$\begin{aligned} R_{\text{affine}}(u,v) &= \int_\Omega \Psi(||H(u^t)||_F^2) + \Psi(||H(v^t)||_F^2) \; dxdy \;\; , \\ &= \int_\Omega (u_{xx}^2 + 2u_{xy}^2 + u_{yy}^2 + v_{xx}^2 + 2v_{xy}^2 + v_{yy}^2) \; dxdy \;\; , \end{aligned} \tag{5.16}$$

one can see the main difference is, that Hager penalises the components separately, in contrast to Maurer et al. who uses a directional penalisation. If no penaliser is used, both regularisers are identical for the case of $a = \nabla u$ and $b = \nabla v$.

With this as a basis, Maurer et al. introduces four adaptive schemes. A global, a local, a non-local and a region based scheme. In order to achieve meaningful results, the following will be focused only on the global and local adaptive scheme.

5.2.1 Global Scheme

The simplest scheme introduced by Maurer et al. is the global order adaptive scheme. The regularization term consist of a weighted combination of the previous presented smoothness terms and reads

$$\begin{aligned} R_{\text{global}}(u,v,c) = \int_\Omega \inf_{a,b} \{ c \cdot S_1(u,v) + (1-c) \cdot (S_2(u,v,\boldsymbol{a},\boldsymbol{b}) + T) \\ + \beta \cdot S_{\text{aux}}(\boldsymbol{a},\boldsymbol{b}) + \lambda \cdot \phi(c) \} \ \ dxdy \ \ . \end{aligned} \tag{5.17}$$

Hereby the $c \in [0,1]$ plays the role of the global weighting parameter, with its costs described in the selection term $\phi(c)$, at which we will look at later on. The T are additional activation costs, since S_2 should only be preferred if a minimum average benefit is achieved in comparison to taking S_1. Hence, the choice of T controls the impact of small fluctuations. The higher T is chosen, the less overfitting takes place. To incorporate this behavior Maurer et al. formulated the calculation of the parameter c in terms of a sigmoid function, with the steering parameter λ, reading:

$$\begin{aligned} c &= \frac{1}{1+e^{-\Delta/\lambda}} \\ \Delta &= T + \frac{1}{|\Omega|} \int_\Omega S_2(u,v,\boldsymbol{a},\boldsymbol{b}) - S_1(u,v) \ \ dxdy \ \ . \end{aligned} \tag{5.18}$$

What remains is to show the choice of the selection term $\phi(c)$. From the derivative $\frac{\partial R_{\text{global}}}{\partial c} = 0$ follows.:

$$0 = \int_\Omega S_1(u,v) - S_2(u,v,\boldsymbol{a},\boldsymbol{b}) + \phi'(c) \ \ dxdy \ \ . \tag{5.19}$$

Rewriting the equations from 5.18 such that $\int_\Omega S_2 - S_1 \ \ dxdy$ can be replaced in the above Equation 5.19,

$$\int_\Omega S_2(u,v,\boldsymbol{a},\boldsymbol{b}) - S_1(u,v) \ \ dxdy = |\Omega| \left(\lambda \cdot \ln \left(\frac{1}{c} - 1 \right) + T \right) \ \ , \tag{5.20}$$

leads to the following term for $\phi'(c)$:

$$\phi'(c) \quad = \quad -\lambda \cdot \ln\left(\frac{1}{c} - 1\right) - T \ . \tag{5.21}$$

Integrating with T as the integration constant enables to drop λ. The selection term $\phi(c)$ changes to:

$$\begin{aligned} \phi(c) &= \ln(1-c) - c \cdot \left(\frac{1}{c} - 1\right) \ , \\ &= (1-c) \cdot \ln(1-c) + c \cdot \ln(c) \ . \end{aligned} \tag{5.22}$$

For a more detailed derivation please see [MSB17]. Since the flow is given later on, $S_1(u,v)$ becomes constant. R_{global} can them be reformulated to:

$$R_{\text{global}}(\boldsymbol{a}, \boldsymbol{b}, c) = \int_\Omega \inf_{a,b} \left\{(1-c) \cdot (S_2(\boldsymbol{a}, \boldsymbol{b}) + T) + \beta \cdot S_{\text{aux}}(\boldsymbol{a}, \boldsymbol{b}) + \lambda \cdot \phi(c)\right\} \ dxdy \ . \tag{5.23}$$

5.2.2 Local Scheme

The main difference between the global and local method is, that global value c is changed to a function c_{local} mapping each position $(x, y) \in \Omega$ to a value in the interval of $[0, 1]$. Therefore the following formula is used to calculate the value of c_{local} at a specific position, where only the benefit at the current position is taken into account.

$$\begin{aligned} c_{\text{local}} &= \frac{1}{1 + e^{-\Delta/\lambda}} \\ \Delta &= T + S_2(u, v, \boldsymbol{a}, \boldsymbol{b}) - S_1(u, v) \quad . \end{aligned} \tag{5.24}$$

The regulariser for a given flow then reads:

$$R_{\text{local}}(\boldsymbol{a}, \boldsymbol{b}, c_{\text{local}}) = \int_\Omega \inf_{a,b} \left\{(1-c_{\text{local}}) \cdot (S_2(\boldsymbol{a}, \boldsymbol{b}) + T) + \beta \cdot S_{aux}(\boldsymbol{a}, \boldsymbol{b}) + \lambda \cdot \phi(c_{\text{local}})\right\} \ dxdy \ . \tag{5.25}$$

5.3 Evaluation of the Order-Adaptive Schemes

Since dense flow fields are required for the determination of the movement order, interpolated ground truth flow is used for the KITTI benchmark. The interpolation is done using the EpicFlow algorithm as described in Section 2.6. Figure 5.1 shows the ground truth and interpolated ground truth of the first frame of both KITTI benchmarks.

As mentioned before, we already have the flow field and are only interested in the determination of c and the auxiliary functions. Therefore, the approach is simply initialized with the flow field and u and v are held fixed. The values for T were chosen to be different powers of ten, while the other parameters were kept the same, $\beta = 10^{-2}$ and $\lambda = 10^{-5}$. The statistics are evaluated at those points, where the original ground truth is given. Figure 5.2 additionally shows the histograms of the amount of local second order movement contained in the benchmarks for the T value of 10^{-2}, to visualize the distribution of scenes definitely containing affine motion.

Looking at Table 5.1, the highest amount of global second oder movement is achieved by the KITTI 2012 benchmark, with the highest set value of the activation costs T. It is followed by

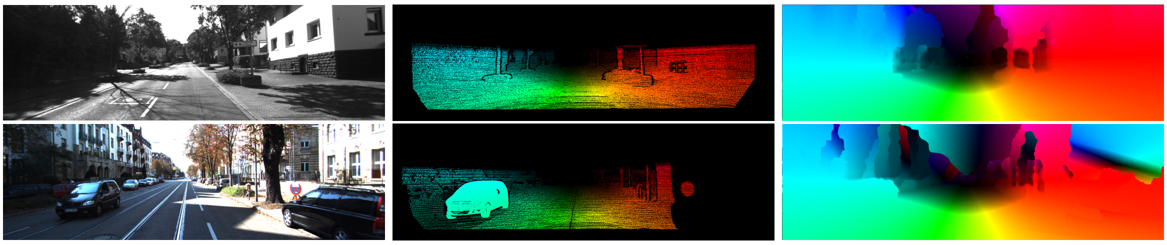


Figure 5.1: Visualization of the interpolated ground truth flow fields of the first frames of the KITTI benchmarks. *From left to right:* First frame, ground truth and interpolated ground truth using EpicFlow.

T/ Benchmark	KITTI 2015	KITTI 2012	MPI Sintel	Middlebury
10^{-2}	80.5 %	**93.8 %**	18.3 %	0.00 %
10^{-3}	87.0 %	**97.9 %**	28.7 %	0.00 %
10^{-4}	94.5 %	**99.4 %**	61.9 %	12.5 %

Table 5.1: Global movement categorization (amount of second order, ground truth).

T/ Benchmark	KITTI 2015	KITTI 2012	MPI Sintel	Middlebury
10^{-2}	57.3 %	**67.3 %**	13.6 %	4.46 %
10^{-3}	67.9 %	**75.1 %**	21.0 %	6.71 %
10^{-4}	72.3 %	**80.2 %**	40.4 %	19.4 %

Table 5.2: Local movement categorization (amount of second order, ground truth).

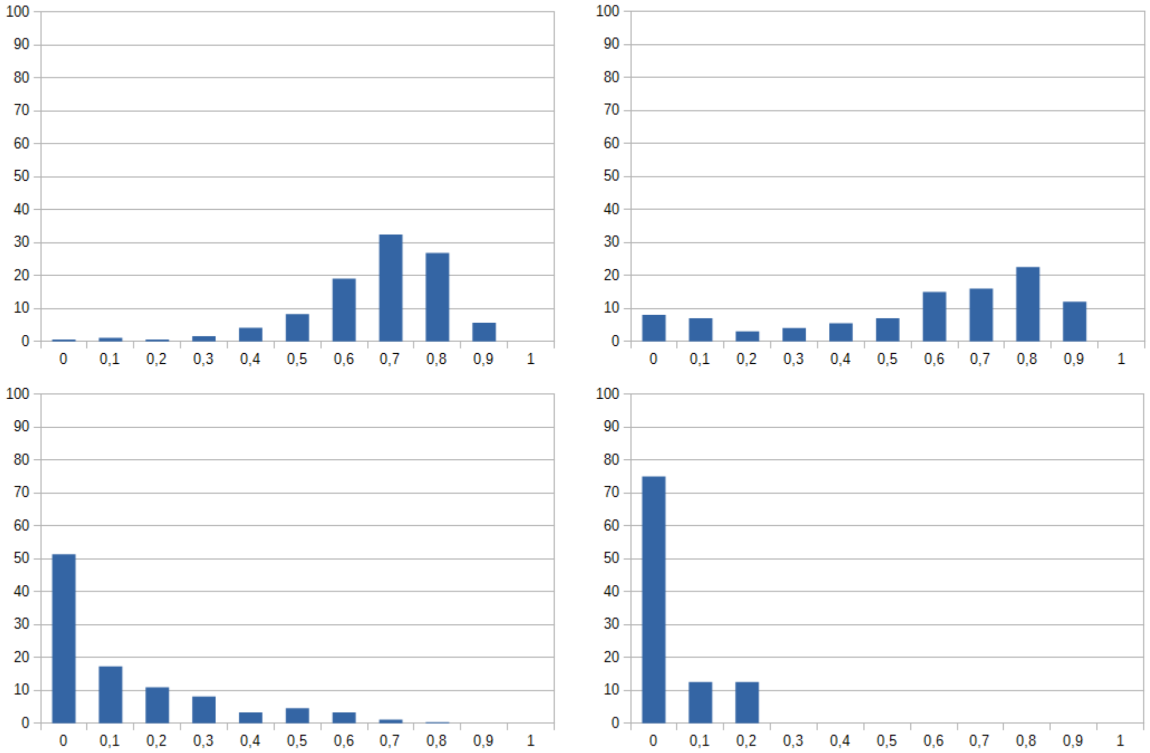


Figure 5.2: Normalized histograms of the amount of local second order movement contained in the benchmarks for the T value of 10^{-2}. *First row from left to right:* KITTI 2012 and 2015 test suite. *Second row from left to right:* MPI Sintel and Middlebury benchmark. (x-axis = aomunt in decimal percent, y-axis = frames in percent).

the KITTI 2015 test suite with a value of 80.5 %. These high results are due to the interpolation method and should be treated with caution. The MPI Sintel benchmark achieves a moderate value of 18.3 % of global second order movement. The Middlebury benchmarks contains no global second order movement as one would expect, due to the small displacements between the frames.

However, if we look at the achieved values in Table 5.2, we can observe that the Middlebury benchmark contains some affine movements. These are mostly found in the Urban scenes, where the camera moves into the scene and not from left to right. In comparison the MPI Sintel test suite contains a lot more affine movements than the Middlebury benchmark. Especially the close up scenes with a lot of action, as the Ambush5, contribute a lot second order movement to the statistics. However the histogram of the MPI Sintel test suite reveals, that the majority of the scenes contains constant movement.

Concerning the KITTI test suites, very high values are obtained by the analysis of local affine movement throughout the benchmarks, Figure 5.2 row one. Taking a look at Figure 5.1 one can see why these values are that high. Constant movement can only be found in the middle part of the image, when evaluating the flow field visually. Additionally the entire upper part is not taken into account, where one would expect the constant movement of the sky. As mentioned before only those pixels are taken into account, where ground truth flow is available. All of these factors cause the high values achieved by the KITTI test suites. Also note the difference between the KITTI 2015 and 2012 test suite. The 2015 benchmark contain less second order movements, since it contains scenes, where the car carrying the camera setup is not moving but standing at an intersection. In contrast, the KITTI 2012 test suite only contains scenes where the car is moving. All in all it can be said, that the KITTI benchmarks contain the most amount of second order movement, since the distance to the MPI Sintel test suite is very large. The last place goes to the Middlebury benchmark.

Although we use a first order model to calculate the computed flow, it is evaluated in the following tables.

T/ Benchmark	KITTI 2015	KITTI 2012	MPI Sintel	Middlebury
10^{-2}	**100 %**	**100 %**	16.5 %	0.00 %
10^{-3}	**100 %**	**100 %**	29.5 %	0.00 %
10^{-4}	**100 %**	**100 %**	66.8 %	12.5 %

Table 5.3: Global movement categorization (amount of second order, computed flow).

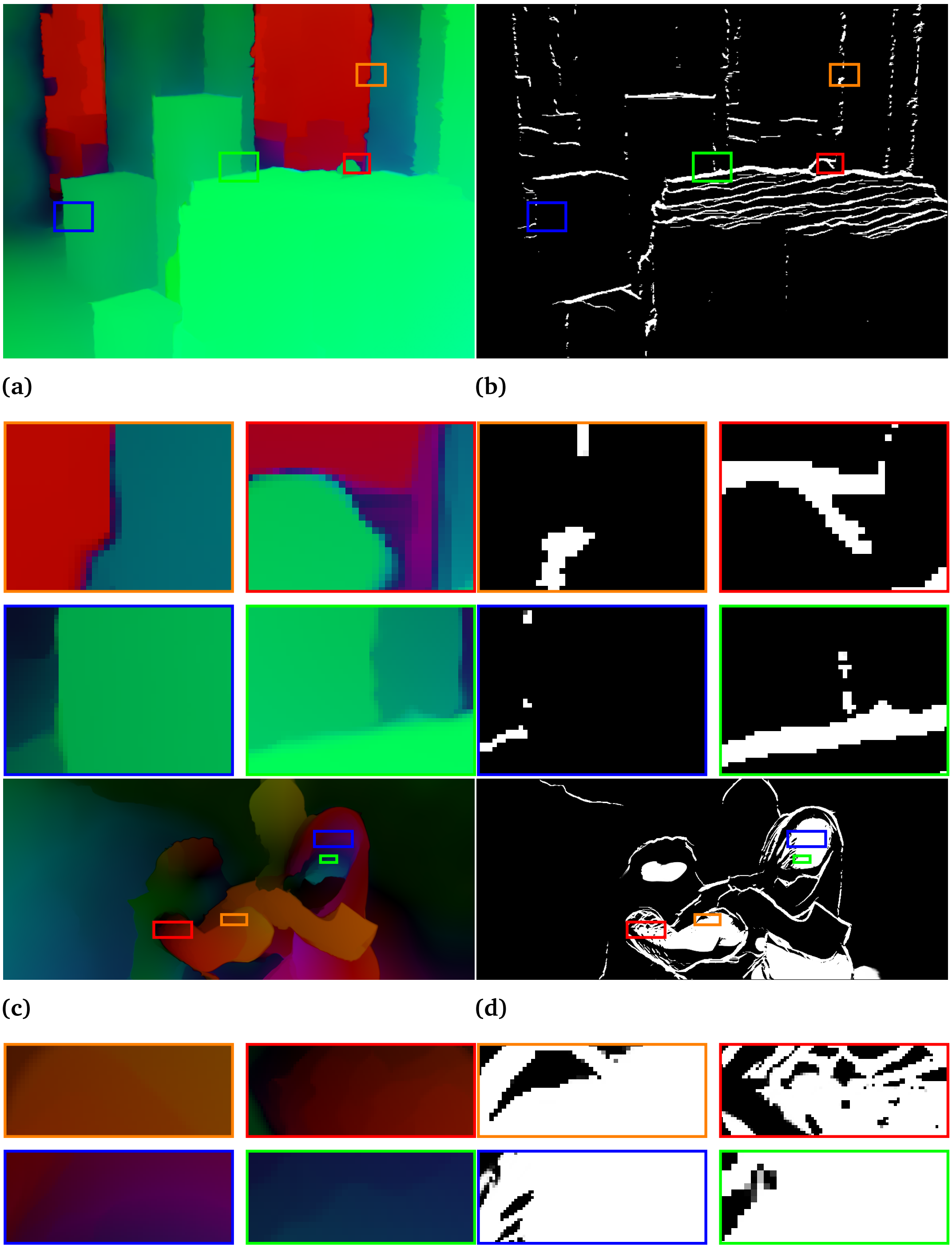
T/ Benchmark	KITTI 2015	KITTI 2012	MPI Sintel	Middlebury
10^{-2}	0.00 %	0.00 %	**15.8 %**	3.33 %
10^{-3}	0.00 %	0.00 %	**25.0%**	9.41 %
10^{-4}	0.00 %	0.00 %	**50.1 %**	29.0 %

Table 5.4: Local movement categorization (amount of second order, computed flow).

Looking at the local movement categorization evaluation on the KITTI test suites, no second order movement is found. In contrast, the global movement evaluation resulted in 100% second order movement. These results are reasonable, since a first order model was used for the computation resulting in local constant flow fields. However, the global movement remains second order.

The MPI Sintel and Middlebury benchmark achieve higher values as when using the ground truth flow for the evaluation. The first explanation for this behavior are the outliers produced at the motion boundaries and at occluded regions. An example is depicted in Figure 5.3 row one. The second explanation, shown in row two of the figure is, that the smoothness parameter of the computed flow is chosen very small. This leads to lots of of small constant movement areas in the flow field, but since the the auxiliary functions are also smoothed the total area is classified as affine movement.

In comparison, Hager evaluated that the MPI Sintel benchmark contains more points containing affine movement. Looking at the amount of local constant movement found in all the benchmarks Hager obtained values of 35 to 40 %. These values match with the ones of the KITTI benchmarks being 42.7% for 2015 and 32.7% for 2012. However large differences are found when comparing the results to the ones of the MPI Sintel and Middlebury benchmark.



(a)

(b)

(c)

(d)

Figure 5.3: Examples of second order movement found in the Middlebury and MPI Sintel test suite using computed flow ($T = 10^{-2}$). *First row, from left to right:* Computed flow of the Urban2 scene and corresponding categorization map. *Second row, from left to right:* Computed flow of the Ambush5 scene between Frames 8 and 9 and corresponding categorization map.

6 Conclusion and Future Work

6.1 Conclusion

This thesis has introduced metrics to determine illumination changes, large displacements and movement orders. In the case of illumination changes it has been shown, that a variational approach based on brightness transfer functions, enforcing small values of the additive components, leads to better interpretable results. The large displacement evaluation was extended by a scale analysis using three different methods. As a result, the patch and marching approach to determine the scale value of a pixel performed best. The scale values were later on used in the large displacement evaluation. Finally, the movement was categorized using the order adaptive approach of Maurer et al. and compared to the results of Hagers approach. The following tables summarize the results of the measurements made with the novel metrics.

The amount of illumination changes measured using the standard deviation of the multiplicative component from one of the approximated brightness transfer function, resulted in about 45%. The highest standard deviation was achieved on the KITTI 2015 test suite for both ground truth and computed flow. However, the most complex scenes regarding illumination changes are contained in the MPI Sintel benchmark. The least amount of illumination changes were found in the Middlebury benchmark with a value of about 20% for both flows.

Benchmark/ Difficulty	Illumination changes (STD)		Large displacements (amount)		Movement categorization (amount of second order)	
	mult. component	add. component	scale via patch	scale via marching	c map global	c map local
KITTI 2015	**0.43**	0.96	**50.6%**	**5.82%**	80.5%	57.3%
KITTI 2012	0.38	0.94	45.7%	2.51%	**93.8%**	**67.3%**
MPI Sintel	0.29	**1.03**	9.10%	0.33%	18.3%	13.6%
Middlebury	0.19	1.01	0.02%	0.07%	0.00%	4.46%

Table 6.1: Comparison of the benchmarks using ground truth flow.

Benchmark/ Difficulty	Illumination changes (STD)		Large displacements (amount)		Movement categorization (amount of second order)	
	mult. component	add. component	scale via patch	scale via marching	c map global	c map local
KITTI 2015	**0.48**	1.00	27.8%	0.44%	**100%**	0.00%
KITTI 2012	0.46	0.94	**32.6%**	0.09%	**100%**	0.00%
MPI Sintel	0.33	**1.05**	8.07%	0.20%	16.5%	**15.8%**
Middlebury	0.20	1.02	0.15%	**0.47%**	0.00%	12.5%

Table 6.2: Comparison of the benchmarks using computed flow.

Regarding large displacements of small objects in Table 6.1, the same complexity order of the benchmarks was obtained as in the illumination evaluation. The comparison with the results achieved with the computed flow showed, how much impact the sparse ground truth flow of the KITTI benchmark had on the measurements. The very high amount of large displacement measured in the computed flow of the Middlebury test suite was due to outliers at motion borders. Combining the insights of patch, marching approach and both flows, revealed, that both MPI Sintel and KITTI 2015 have the most amount of large displacements. In contrast, the KITTI 2012 and Middlebury benchmark contain the least amount of large displacements.

The movement categorization resulted in very high values for the KITTI Vision benchmarks due to the interpolation with EpicFlow. In comparison to the other benchmarks both KITTI Vision benchmarks contain almost only second order movement accept for a couple of scenes in the 2015 test suite. Small amounts of second order movement were also found in the scenes of the MPI Sintel benchmark. The least amount was measured in the Middlebury benchmark.

Looking at all difficulties the KITTI Vision benchmarks resulted on average as the most complex ones, achieving very high values in all categories. The MPI Sintel test suite shows moderate complexity on average and the Middlebury test suite is the simplest one compared to the other benchmarks.

6.2 Future Work

The following suggests different ideas for further refinement of the presented techniques and other evaluation possibilities.

Concerning illumination changes the standard deviation of the multiplicative component and the standard deviation of the additive component from zero was analyzed. Since the multiplicative component is responsible for shadows or brighter areas, it would be very interesting to extract these regions. This could be achieved for example by using [WT05, LY12]. These results could then be compared to the multiplicative component images. Additionally, further refinement of the variational approach with different penalizer functions or adding additional assumptions in the regularization could lead to further improvements of determining the brightness transfer function coefficients.

A difficulty that has not been regarded so far are the amount of occluded regions. A statistic of how much occlusion is present in a benchmark could lead to further insight on how reliable the Metrics work, since it was shown that these regions cause problems during the evaluation, e.g. the scale calculation jumping to other objects due to outliers.

The second difficulty discussed in this thesis were large displacements. In this case the main problem was finding a method to extract the scale of an object pixel wise. Although two different approaches have been introduced, a more robust method is of great interest. Therefore a difference of Gaussians approach could be used with other adaptions or a preceding structure texture decomposition at different levels, even though the first attempt did not succeed. Another possibility would be to determine the scale not only using the flow, as done in this thesis, but to use a combination of both frame and flow to obtain a feasible scale value for each pixel. This would also deal with the problematic of rotating objects, where the similarity strongly varies in a small region causing difficulties in the patch approach.

Because of the sparse ground truth flows given in the KITTI benchmarks the metrics needed special handling most of the time. Obviously, the analysis works best on dense flow fields. These metrics could be evaluated using different methods for the determination of the flow field. It could be worth to evaluate if approaches, who achieve similar average angular and end point errors, lead to similar results when evaluating the metrics.

Bibliography

[AB85] E. H. Adelson, J. R. Bergen. Spatiotemporal energy models for the perception of motion. *Journal of the Optical Society of America A*, 2(2):284–299, 1985. (Cited on page 12)

[AMN13] S. Aslani, H. Mahdavi-Nasab. Optical flow based moving object detection and tracking for traffic surveillance. *World Academy of Science, Engineering and Technology, International Journal of Electrical, Computer, Energetic, Electronic and Communication Engineering*, 7(9):1252–1256, 2013. (Cited on page 7)

[BB95] S. S. Beauchemin, J. L. Barron. The computation of optical flow. *ACM Computing Surveys (CSUR)*, 27(3):433–466, 1995. (Cited on page 14)

[BBPW04] T. Brox, A. Bruhn, N. Papenberg, J. Weickert. High accuracy optical flow estimation based on a theory for warping. In *Proc. of the European Conference on Computer Vision*, pp. 25–36. 2004. (Cited on page 12)

[BETVG08] H. Bay, A. Ess, T. Tuytelaars, L. Van Gool. Speeded-up robust features (SURF). *Computer Vision and Image Understanding*, 110(3):346–359, 2008. (Cited on pages 19 and 49)

[BFB94] J. L. Barron, D. J. Fleet, S. S. Beauchemin. Performance of optical flow techniques. *International Journal of Computer Vision*, 12(1):43–77, 1994. (Cited on page 7)

[BGW91] J. Bigun, G. H. Granlund, J. Wiklund. Multidimensional orientation estimation with applications to texture analysis and optical flow. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(8):775–790, 1991. (Cited on page 12)

[BLLJ09] W. Bao, H. Li, N. Li, W. Jiang. A liveness detection method for face recognition based on optical flow field. In *Proc. of the International Conference on Image Analysis and Signal Processing*, pp. 233–236. 2009. (Cited on page 7)

[BSL$^+$11] S. Baker, D. Scharstein, J. Lewis, S. Roth, M. J. Black, R. Szeliski. A database and evaluation methodology for optical flow. *International Journal of Computer Vision*, 92(1):1–31, 2011. (Cited on pages 3, 4, 7, 8 and 15)

[BWSB12] D. J. Butler, J. Wulff, G. B. Stanley, M. J. Black. A naturalistic open source movie for optical flow evaluation. In *Proc. of the European Conference on Computer Vision*, pp. 611–625. 2012. (Cited on pages 3, 4, 8 and 15)

[CBFAB97] P. Charbonnier, L. Blanc-Féraud, G. Aubert, M. Barlaud. Deterministic edge-preserving regularization in computed imaging. *IEEE Transactions on Image Processing*, 6(2):298–311, 1997. (Cited on page 68)

[CLTX16] C. Chen, M.-Y. Liu, O. Tuzel, J. Xiao. R-CNN for small object detection. In *Proc. of the Asian Conference on Computer Vision*, pp. 214–230. 2016. (Cited on page 49)

[CM99] T. F. Chan, P. Mulet. On the convergence of the lagged diffusivity fixed point method in total variation image restoration. *SIAM Journal on Numerical Analysis*, 36(2):354–367, 1999. (Cited on page 30)

[DV97] D. C. Dobson, C. R. Vogel. Convergence of an iterative method for total variation denoising. *SIAM Journal on Numerical Analysis*, 34(5):1779–1791, 1997. (Cited on page 30)

[DZ13] P. Dollár, C. L. Zitnick. Structured forests for fast edge detection. In *Proc. of the IEEE International Conference on Computer Vision*, pp. 1841–1848. 2013. (Cited on page 22)

[GCT98] A. Giachetti, M. Campani, V. Torre. The use of optical flow for road navigation. *IEEE Transactions on Robotics and Automation*, 14(1):34–48, 1998. (Cited on page 7)

[GLU12] A. Geiger, P. Lenz, R. Urtasun. Are we ready for autonomous driving? The KITTI Vision Benchmark Suite. In *Proc.of the IEEE Conference on Computer Vision and Pattern Recognition*. 2012. (Cited on pages 3, 4, 8 and 15)

[GMN$^+$98] B. Galvin, B. McCane, K. Novins, D. Mason, S. Mills, et al. Recovering Motion Fields: An Evaluation of Eight Optical Flow Algorithms. In *Proc.of the British Machine Vision Conference*, volume 98, pp. 195–204. 1998. (Cited on page 7)

[GN04] M. D. Grossberg, S. K. Nayar. Modeling the space of camera response functions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(10):1272–1282, 2004. (Cited on page 25)

[GN06] M. Grossberg, S. Nayar. What can be known about the radiometric response from images? *Proc. of the European Conference on Computer Vision*, pp. 393–413, 2006. (Cited on page 18)

[Hag17] J. Hager. An Analysis of Difficulties and Regularities in Optical Flow Benchmarks. 2017. (Cited on pages 3, 4, 8, 9 and 66)

[Hee87] D. J. Heeger. Model for the extraction of image flow. *Journal of the Optical Society of America A*, 4(8):1455–1471, 1987. (Cited on page 12)

[HS81] B. K. Horn, B. G. Schunck. Determining optical flow. *Artificial Intelligence*, 17(1-3):185–203, 1981. (Cited on page 12)

[HSW15] D. Hafner, C. Schroers, J. Weickert. Introducing maximal anisotropy into second order coupling models. In *Proc. of the German Conference on Pattern Recognition*, pp. 79–90. 2015. (Cited on page 68)

[JHG99] B. Jähne, H. Haussecker, P. Geissler. *Handbook of computer vision and applications*, volume 2. 1999. (Cited on page 11)

[KS13] A. Kaplan, D. Stulik. The first scientific investigation of Niépce's images from UK and US collections: image substrate. *The Imaging Science Journal*, 61(8):629–646, 2013. (Cited on page 7)

[Lin94] T. Lindeberg. Scale-space theory: A basic tool for analyzing structures at different scales. *Journal of Applied Statistics*, 21(1-2):225–270, 1994. (Cited on page 19)

[LK$^+$81] B. D. Lucas, T. Kanade, et al. An iterative image registration technique with an application to stereo vision. 1981. (Cited on page 12)

[Low99] D. G. Lowe. Object recognition from local scale-invariant features. In *Proc. of the Seventh IEEE International Conference on Computer Vision*, volume 2, pp. 1150–1157. 1999. (Cited on pages 19 and 49)

[LY12] W. Liu, F. Yamazaki. Object-based shadow extraction and correction of high-resolution optical satellite images. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 5(4):1296–1302, 2012. (Cited on page 79)

[LZ05] S. Lin, L. Zhang. Determining the radiometric response function from a single grayscale image. In *Proc.of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pp. 66–73. 2005. (Cited on page 18)

[MG15] M. Menze, A. Geiger. Object scene flow for autonomous vehicles. In *Proc.of the IEEE Conference on Computer Vision and Pattern Recognition*. 2015. (Cited on pages 3, 4, 8 and 15)

[MLBV10] V. Mahadevan, W. Li, V. Bhalodia, N. Vasconcelos. Anomaly detection in crowded scenes. In *Proc.of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1975–1981. 2010. (Cited on page 7)

[MN99] T. Mitsunaga, S. K. Nayar. Radiometric self calibration. In *Proc.of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pp. 374–380. 1999. (Cited on page 18)

[MSB17] D. Maurer, M. Stoll, A. Bruhn. Order-Adaptive Regularisation for Variational Optical Flow: Global, Local and in Between. In *Proc. of the International Conference on Scale Space and Variational Methods in Computer Vision*, pp. 550–562. 2017. (Cited on pages 3, 4, 9, 65, 67 and 71)

[NBK08] T. Nir, A. M. Bruckstein, R. Kimmel. Over-parameterized variational optical flow. *International Journal of Computer Vision*, 76(2):205–216, 2008. (Cited on page 16)

[NY93] S. Negahdaripour, C.-H. Yu. A generalized brightness change model for computing optical flow. In *Proc. of the Fourth IEEE International Conference on Computer Vision*, pp. 2–11. 1993. (Cited on page 26)

[PM90] P. Perona, J. Malik. Scale-space and edge detection using anisotropic diffusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(7):629–639, 1990. (Cited on page 68)

[RWHS15] J. Revaud, P. Weinzaepfel, Z. Harchaoui, C. Schmid. Epicflow: Edge-preserving interpolation of correspondences for optical flow. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1164–1172. 2015. (Cited on pages 21, 22 and 23)

[S$^+$94] J. Shi, et al. Good features to track. In *Proc.of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 593–600. 1994. (Cited on page 16)

[S$^+$97] S. W. Smith, et al. The scientist and engineer's guide to digital signal processing. 1997. (Cited on page 19)

[VRKM08] T. Vaudrey, C. Rabe, R. Klette, J. Milburn. Differences between stereo and motion behaviour on synthetic and real-world stereo sequences. In *Proc. of the 23rd International Conference on Image and Vision Computing New Zealand*, pp. 1–6. 2008. (Cited on page 7)

[Was13] L. Wasserman. *All of statistics: a concise course in statistical inference*. 2013. (Cited on page 22)

[WBSB12] J. Wulff, D. Butler, G. Stanley, M. Black. Lessons and insights from creating a synthetic optical flow benchmark. In *Proc. of the European Conference on Computer Vision*, pp. 168–177. 2012. (Cited on page 7)

[WT05] T.-P. Wu, C.-K. Tang. A bayesian approach for shadow extraction from a single image. In *Proc. of the Tenth IEEE International Conference on Computer Vision*, volume 1, pp. 480–487. 2005. (Cited on page 79)

[YLL$^+$08] D. Yang, H. Li, D. A. Low, J. O. Deasy, I. El Naqa. A fast inverse consistent deformable image registration method based on symmetric optical flow computation. *Physics in Medicine and Biology*, 53(21):6143, 2008. (Cited on page 7)

[ZBW$^+$09] H. Zimmer, A. Bruhn, J. Weickert, L. Valgaerts, A. Salgado, B. Rosenhahn, H.-P. Seidel. Complementary optic flow. In *Proc. of the International Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition*, pp. 207–220. 2009. (Cited on page 40)

[ZBW11] H. Zimmer, A. Bruhn, J. Weickert. Optic flow in harmony. *International Journal of Computer Vision*, 93(3):368–388, 2011. (Cited on pages 14 and 68)

All links were last followed on November 5, 2017.

Declaration

I hereby declare that the work presented in this thesis is entirely my own and that I did not use any other sources and references than the listed ones. I have marked all direct or indirect statements from other sources contained therein as quotations. Neither this work nor significant parts of it were part of another examination procedure. I have not published this work in whole or in part before. The electronic copy is consistent with all submitted copies.

place, date, signature