Institute of Visualization and Interactive Systems
University of Stuttgart
Universitätsstraße 38
D - 70569 Stuttgart

Master's Thesis

An Analysis of Difficulties and Regularities in Optical Flow Benchmarks

Janik M. Hager

Course of Study:	Computer Science
Examiner:	Prof. Dr.-Ing. Andrés Bruhn
Supervisor:	Daniel Maurer, M.Sc.
Commenced:	03. October 2016
Completed:	04. April 2017
CR-Classification:	G.1.6, G.1.8, I.4.8

ABSTRACT

The extraction of information considering the movement of objects in an image sequence becomes more and more an important problem, amongst others in the area of computer vision. In most cases, a displacement vector field between two consecutive frames of an image sequence should be computed which is often called optical flow. Several methods and approaches have been introduced to compute this optical flow which is why some optical flow benchmarks have been created to evaluate them. These benchmarks contain synthetic and non-synthetic data like the Middlebury Benchmark, synthetic data from an animated short film like the MPI-Sintel Benchmark or real-world data collected by an autonomous driving car like the KITTI Vision Benchmark Suite. However, the difficulties of these benchmarks haven't been investigated so far which is why different metrics should be developed in this thesis to evaluate them. These address image statistics, optical flow statistics, illumination changes, the type of movement and egomotion in stereo scenes. First of all, they are applied on the training data with ground truth flow to estimate afterwards the difficulty of the testing data. The benchmarks which are used are the three mentioned before.

KURZFASSUNG

Die Extraktion von Informationen bezüglich der Bewegung von Objekten aus Bildsequenzen wird zu einem immer bedeutenderen Problem, unter anderem im Bereich des Maschinensehens. Meistens soll dabei zwischen zwei aufeinander folgenden Frames einer Bildsequenz ein Verschiebungsvektorfeld berechnet werden, welches häufig als optischer Fluss bezeichnet wird. Viele verschiedene Methoden und Ansätze wurden entwickelt, um den optischen Fluss zu berechnen, weshalb einige Benchmarks entworfen wurden, um sie zu bewerten. Diese enthalten beispielsweise synthetische und reale Daten, wie der Middlebury Benchmark, synthetische Daten aus einem animierten Kurzfilm, wie der MPI-Sintel Benchmark, oder reale Daten, die mithilfe eines selbstfahrenden Fahrzeugs gesammelt wurden, wie beim KITTI Vision Benchmark Suite. Allerdings wurden die Schwierigkeitsgrade dieser Benchmarks bisher kaum untersucht, weshalb in dieser Abschlussarbeit unterschiedliche Metriken entwickelt werden sollen, um diese zu evaluieren. Diese befassen sich mit Bildstatistiken, Statistiken zum optischen Fluss, Beleuchtungsänderungen, der Art der Bewegung und Eigenbewegung in Stereoszenen. Sie werden zunächst auf die Trainingsdaten mit dem Lösungsflussfeld angewendet, um anschließend eine Schätzung der Schwierigkeit für die Testdaten zu ermöglichen. Untersucht werden dabei die drei oben genannten Benchmarks.

CONTENTS

1. INTRODUCTION

1.1 MOTIVATION

Nowadays, it becomes more and more important that robots or computers are able to see images and extract valuable information from them to solve a specific task. The type of information depends on the task to be solved. An example for such a scenario could be a robot which tries to navigate through an unknown environment with obstacles. In this case, the robot needs to extract information about the surrounding obstacles from images to find a path. The complexity of this task increases if the obstacles would be moved or would move by themselves. This is comparable with an automotive driving car where driver assistance systems have to gather informations for example about other cars or street signs. Another possible scenario could be surveillance or tracking traffic.

In many cases, such as in the previous examples, one of these important informations to be extracted is the displacement field between two consecutive images of an image sequence. The displacement field is a vector field which basically shows for each point in the first image the direction and and how far it is moving such that it is found in the second image, that means that it contains a vector for each point from the previous to the new location. In literature, this vector field is also often called optical flow. The problem of finding such an optical flow between two images belongs to the correspondence problems which are one of the most important problems in computer vision.

This is the reason why many strategies and approaches have been invented to solve the optical flow problem. Furthermore, for a possibility to measure the quality of these methods and to compare their strengths and weaknesses with each other, in the last years some optical flow benchmarks have been introduced. One important advantage of the optical flow benchmarks is additionally that they led to significant progress in the development of new methods because of the challenges they proposed. These benchmarks cover various difficulties of optical flow problems and they contain for example simple non-synthetic and complex synthetic data like the Middlebury Benchmark [1], real-world data from the automotive environment like the KITTI Vision Benchmark Suite [2]–[4] and data from an animated 3D short film like the MPI-Sintel Benchmark [5].

These benchmarks offer a good comparison of recent state-of-the-art algorithms but they haven't been analysed themselves so far with respect to their difficulties and regularities. Especially considering some challenges of optical flow problems, the scenes

included in the benchmarks should be evaluated to give a hint about the difficulty of the scenes and the whole benchmark. Some of the more important challenges are the change of illumination, large displacements and the type of movement.

1.2 CONTRIBUTION

Therefore, the goal of this thesis is to determine metrics to provide some kind of measurement for the difficulties and regularities of optical flow benchmarks and the image sequences of them. As mentioned before, the most important and difficult challenges of optical flow problems concern illumination changes, large displacements and categorising the type of movement. The illumination changes could be for example multiplicative or additive and the type of movement could on the one hand be for example constant or affine and on the other hand self motion or independent moving objects. Furthermore, the illumination changes and the movement can be global for the whole image or only local for some objects. That is why the focus of the evaluation is on these challenging topics to allow a good measurement of the difficulties of the Middlebury, the KITTI 2012, the KITTI 2015 and the MPI-Sintel benchmarks.

The first step is to propose some metrics which can be helpful to investigate the benchmarks. For the second step, the given benchmarks are examined by considering the training data sets which contains the ground truth vector field. That means that the solution for the image sequences of this data set is given. In the third and last step, the optical flow is computed from the testing data sets such that the complexity and difficulty of these image sequences can be estimated.

1.3 RELATED WORK

Since the so far existing optical flow benchmarks haven't really been evaluated concerning some of the before mentioned challenges of optical flow, this section concentrates more on a brief overview of the history of these optical flow benchmarks. A more detailed description about the benchmarks which are evaluated in this thesis is to be found in Section 3.

Benchmarks in general are always very important for researching purposes because weaknesses of state-of-the-art algorithms can be shown this way. Furthermore, a comparison and quantitative evaluation of these methods is possible with their help and they allow an improvement of the so far existing work. This is the main advantage of benchmarks and the reason for introducing them. However, Butler et al. [5] mention in a summary of main aspects of benchmarks in general that some problems with benchmarks may arise, too. One of the biggest problems of benchmarks is the limited lifespan since at some point the existing algorithms are too advanced and therefore neither a good measurement nor a comparison between the methods nor a demonstra-

tion of current weaknesses of these approaches is possible anymore. Another problem is that benchmarks cover only some of the challenges of the task to be solved and hence there has to be made a selection of them. Nevertheless, it is indispensable to use benchmarks and to develop new ones from time to time which introduce new challenges and try to approximate realistic scenes more and more.

Therefore Barron et al. [6] presented in 1994 the first optical flow benchmarks which even was one of the first benchmarks in the field of computer vision. They used both real and synthetic data with ground truth flow for the evaluation of the algorithms. Although the data was very simple, especially for the current methods, there was a big difference in the quality of the evaluated algorithms. Thus this benchmark led to improvements because now the possibility of an evaluation of optical flow algorithms existed. In 1994, Otte and Nagel [7] introduced an optical flow benchmark, too, using a robot arm to extract the motion of the real scenery containing rather simple polyhedral objects and computing the ground truth flow this way. Both of these benchmarks were basically the first attempts of creating a good evaluation possibility for comparing optical flow methods with each other and they were used for several years until McCane et al. [8] presented a new benchmark in 2001 with more realistic polyhedral scenes and synthetic data with varying complexity.

These before mentioned benchmarks formed the basis for later appearing benchmarks, especially for the ones used in this thesis. The Middlebury Benchmark was introduced in 2007 by Baker et al. as a preliminary version and they reviewed and extended it in 2011 [1]. This benchmark contains both simple non-synthetic data and more complex and realistic synthetic data. The ground truth flow of the real data has been acquired by tracking a hidden fluorescent texture. In 2012, Geiger et al. [2], [4] presented the KITTI Vision Benchmark Suite with data from real sceneries from the automotive environment. The autonomous driving platform was navigating through a city and at the same time recording several images while using a laser scanner for the measurement of the ground truth flow. Butler et al. [5] introduced their MPI-Sintel Benchmark in 2012 as well. This benchmark contains synthetic data which has been obtained by rendering a 3D animated short film. This way, the ground truth flow could be computed.

1.4 ORGANISATION

In Section 2, some foundations are offered to ease the understanding of the metrics. Additionally, an overview of the definitions of correspondence problems, the optical flow problem in general and some of its difficulties and regularities is provided. The optical flow benchmarks which are going to be investigated in this thesis are presented in Section 3 where some of their characteristics are summarised. While the metrics are introduced in Section 4, the difficulties of the benchmarks are then evaluated in Section 5. This thesis concludes with a summary of the results in Section 6.

2. FOUNDATIONS

This section gives a brief overview of some foundations of computer vision regarding optical flow which are necessary to ease the understanding of the later introduced metrics and some definitions about the optical flow problem in general and its classification are explained. First of all, the basic characteristics of images and energy functions are summarised which is followed by the definitions of correspondence problems and the optical flow. After that, a parameterized model is proposed for the illumination changes between two images and the movement type of the optical flow.

2.1 IMAGES

Since the images which are used in the methods are stored in a discrete way, the following definition concentrate on this kind of images. In this thesis both grey value images and colour images are used. These images are defined to be a rectangular two-dimensional domain $\Omega = (1, N) \times (1, M)$. The cells of this grid are called pixels which have a value from the co-domain representing the grey value respectively the colour value of the corresponding pixel. The grey values are ranged from 0 to 255 in case of 8-bit images while colour values are represented by a vector containing three values. The colour images in this thesis are stored as RGB images where each vector component is ranged from 0 to 255 as well. For the grey value images, a small value means a dark grey value whereas a high value means a bright grey value. The three values of the colour vector determine the value of the red, green and blue channel, that means they tell how big the influence of the respective channel is to represent the colour. This way images can be described by their size and the grey values respectively colour values at each pixel. Therefore images are often defined as $f = \{f_{i,j} | i = 1, \ldots, N; j = 1, \ldots, M\}$ where N stands for the size of the image in x-direction and M in y-direction. Most algorithms are designed in such a way that they act on the assumption of continuous images. In the continuous case, the grey or colour values are then represented as a continuous function while for the discrete case, the grey or colour values are only given at the pixel positions.

2.2 ENERGY FUNCTIONS

So-called cost or energy functions E are often used to solve mathematical problems, especially when it comes down to find an optimum. These energy functions have to be minimised in order to find the best solution possible. The energy function always consists of a data term D where assumptions on the data are used. Sometimes a smooth-

ness term S is added to the data term where further assumptions or restrictions help to find for example an unique or a better solution which results in variational methods. One of the most used mathematical definitions for an energy function is $E = D + \alpha \cdot S$, where α is a regularisation parameter to tune the smoothness term. If the input is a continuous variable, the energy function is called a functional. In many cases, the energy functional is minimised by computing the corresponding Euler-Lagrange equations. Assuming that the following energy functional is given:

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

Then the Euler-Lagrange equation for this energy functional reads

$$F_u - \frac{\partial}{\partial x} F_{u_x} - \frac{\partial}{\partial y} F_{u_y} = 0 \ , \tag{2.2}$$

with the Neumann boundary conditions $n^T \nabla u = 0$.

2.3 CLASSIFICATION AND DEFINITION OF THE OPTICAL FLOW PROBLEM

The problem being addressed by the benchmarks to be investigated in this thesis belongs to the category of correspondence problems which is one of the key problems in computer vision. In the following section, the correspondence problems, the optical flow problem and some difficulties and regularities are defined.

2.3.1 *Correspondence Problems*

Since the investigated benchmarks are designed for the optical flow problem the following definition of correspondence problems focuses more on this topic. The simpliest definition of a correspondence problem is that basically there are two sets of entities given and the correspondences between entities of these two sets have to be identified. Using this definition in the context of computer vision would mean that the sets are images which contain pixels being their entities. Therefore the task of correspondence problems in computer vision is to find correspondences between two or more images, for example to recognise the same object in several images. Normally, these images show one or more objects but from different views or they are frames of an image sequence. Regarding problems like the optical flow problem or similar ones, one of these images is considered being the original image from which a disparity map or a displacement field towards the other image(s) should be computed. That means that the solution of a correspondence problem can be a vector field showing the shift from the points of one image to their corresponding positions in the other image(s). That means that in the ideal case the solution of a correspondence problem is a vector field that gives us a vector for each pixel, pointing to the location(s) of its

correspondence(s) in the other image(s). The most common matching scenarios are One-to-One (each pixel in the first image can be matched to exactly one pixel in the other image), Many-to-One (some pixels in the first image can be matched to exactly one pixel in the other image), One-to-Many (at least one pixel in the first image can be matched to more than one pixel in the other image) and Non-Dense (some pixels in the first image can't be matched to a pixel in the second image and vice versa). As written above, the first scenario is the most preferred one since it would mean that an ideal solution is found but the last scenario is the most common one since observed objects might move out of or into the image and therefore no match is possible.

The correspondence problems in computer vision are to be found in the context of for example optical flow estimation, stereo matching, scene flow estimation, medical image registration, particle image velocimetry and many more. As mentioned before, this thesis evaluates benchmarks which address the optical flow problem.

2.3.2 *The Optical Flow Problem*

In the case of the optical flow problem, the given images are from an image sequence where only one camera is used. This means that the images change over time and a displacement field between two consecutive frames is wanted. The optical flow problem is quite complex since many challenges appear making the matching task more difficult. These challenges are described in the next section.

Using mathematical notations, there are two images f and g given with their grey values $f_{i,j}$ and $g_{i,j}$ at pixel position (i, j). The vector field consists of displacements in x- and y-direction, denoted as $u_{i,j}$ and $v_{i,j}$. To solve the task of finding corresponding pixels in the two images, assumptions about the data are made. One type of assumptions that are used are the so-called constancy assumptions which regard a specific behaviour of the grey values respectively colour values of the images. One of the most used constraints is the grey value constancy assumption:

$$f_{i,j} = g_{i+u_{i,j},j+v_{i,j}} \quad . \tag{2.3}$$

The equation 2.3 basically tells us that the pixel of the first image $f_{i,j}$ and the corresponding pixel of the second image $g_{i+u_{i,j},j+v_{i,j}}$ have the same grey value.

These definitions assume discrete images but the pixel movements don't have to be necessarily discrete since the real world is continuous. Therefore most approaches use $f(x, y, t)$ as notation for images, using (x, y) for the position in the image and t denoting the time. That means that for the optical flow problem two frames $f(x, y, t)$ and $f(x, y, t + 1)$ of an image sequence are given. The vector field in this case consists of x-displacements $u(x, y, t)$ and y-displacements $v(x, y, t)$. This leads to a continuous version of the grey value constancy assumption:

Figure 2.1: Example of occluded pixels. The girl is not yet visible in the first image. She appears in the scenery after some frames where she gets occluded by a person which is passing by. **Upper left:** Frame 1. **Upper right:** Frame 5. **Lower left:** Frame 9. **Lower right:** Frame 15. [9]

$$f(x, y, t) = f(x + u, y + v, t + 1) \ . \tag{2.4}$$

Just like in the previous case, we assume that the corresponding pixels have the same grey value. But using this assumption wouldn't always lead to a solution for all pixels because not all grey values of the first image might also exist in the second image. This could happen for example due to illumination changes. Another possibility is that there might be too many or too few pixels with the same grey value. Therefore, deviations from the constancy assumption are allowed but penalised by minimising a cost or energy function. These energy functions depend on other assumptions which are chosen to be used and therefore their solutions differ from each other. In most cases, it is important to choose these other assumptions to guarantee solutions for both unknowns u and v. Furthermore, there are many different strategies that for example try to compensate illumination changes or to guess the movement of the observed objects.

2.3.2.1 *Visualisation of Optical Flow Fields*

An optical flow vector consists basically of a magnitude and a direction. There are three common possibilites to visualise these information for the whole image. The first one is a vector plot where the direction and the magnitude is represented by an arrow. However, this requires subsampling such that only some arrows are shown or else the arrows would overlie each other. The second possibility is a magnitude plot where the magnitude of each optical flow vector is represented as grey value. It is

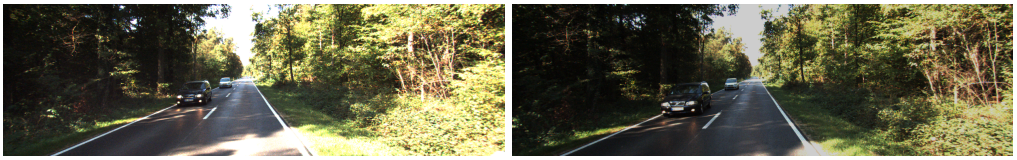

Figure 2.2: Example of illumination change where the second frame is darker than the first one. The second image has been manipulated slightly to demonstrate the effect better. **Left:** First frame. **Right:** Second frame. [3]

possible to show the magnitude of each point this way but the directional information is neglected. Therefore, the most common way to visualise a displacement vector field is a colour plot where the direction is represented by a colour and the magnitude by the brightness of the colour. Though, the colour can be difficult to interpret. This last visualisation is used in this thesis.

2.3.3 *Difficulties in Optical Flow Problems*

As mentioned before, the optical flow problem can be quite difficult due to some challenges. These possible difficulties are summarised in this section. Some of these challenges can be used to evaluate the difficulty of the considered image sequence. By means of this, the later discussed metrics can be used to estimate the difficulty of a benchmark itself. One of the biggest challenge in computing optical flow is the fact that objects in the scene move which means that they might disappear behind other objects or even leave the observed scene which is demonstrated in figure 2.1. These pixels are occluded and can't be matched to their other corresponding points. Something similar happens with objects that move closer to or farther away from the camera because these objects are represented either by a bigger or a smaller amount of pixels.

Another challenge is related to illumination, especially illumination changes between the images like in figure 2.2. This can be difficult to solve since the grey value constancy assumption is easily violated because of it. Furthermore, there are basically two types of illumination changes, namely global and local illumination changes. A global change is easier to handle since every point is changed the same way. But local illumination changes can be quite hard to resolve because they are independent from each other and can even change within an object. There exist possibilities to estimate illumination changes via brightness transfer functions which is discussed later. Illumination effects like specular reflections or shadows and shading which can be seen in figure 2.3 belong to this type of challenge as well which are quite hard to handle.

The last discussed challenge is connected to motion. This is a very important topic since the computation of optical flow is directly related to the motion. Like in the previous challenge, the motion might be either global or local. Local motion means

Figure 2.3: Example of occuring illumination effects. **Upper row:** The car on the left reflects the light such that lense flares appear in the image. **Lower row:** The white car passing from left to right leaves the shadow in the second frame which results in a brighter appearance. [3]

that the objects in the scene move independently from each other and especially from the scene. The scene itself could change due to motion, too, if e.g. the camera is moved. Furthermore, the motion can be of different orders which means that some objects could have a constant movement while others have an affine movement. Therefore, the arbitrary motion of objects in the scene influences the computation of the optical flow. Large displacements are difficult to be computed as well, especially for small objects because it is difficult to find a specific small object which is far away from its origin in the previous image. The estimation of egomotion in image pairs is related to the stereo matching problem but can be applied to optical flow as well.

2.3.4 *Regularities in Optical Flow Problems*

Regularities in general are properties of a problem which ease the solving process because additional assumptions can be taken into account. The regularities of the optical flow problem are related to the difficulties. One of them is the global illumination change since global effects are easier to handle than local ones. The same holds for global movement. Another regularity regarding movement is the type of it. Constant movement decreases the difficulty, especially if it holds true for a region like an object, since all neighbouring points move in the same way. Even affine movement can be taken into consideration.

2.4 MODEL FOR ILLUMINATION CHANGES

In the context of image sequences, it is possible that the illumination changes between two consecutive images. As mentioned before, the information of an image represents the brightness of the image which is why a parameterized brightness transfer function

Figure 2.4: Example of constant and affine movement. **Left:** The person in the foreground has a constant movement to the left since it has nearly everywhere the same colour. **Right:** The fruits on the ground are rotating which results in an affine movement. [9]

Φ is used to describe brightness changes between two images. These functions have been introduced by Grossberg and Nayar [10] in the context of approximation models for response functions and have been e.g. used by Demetz et al. [11] to model illumination changes to ease the computation of optical flow. Since this thesis examines optical flow benchmarks, the here used definition of parameterized brightness transfer functions follows the one of the latter. The parameterized brightness transfer function maps brightness values from the first image to brightness values of the second image. It uses a set of n basis functions $\phi_i : \mathbb{R} \to \mathbb{R}$ and is defined as follows:

$$\Phi(\mathbf{c}, f) = \bar{\phi}(f) + \sum_{i=1}^{n} c_i \cdot \phi_i(f) \quad . \tag{2.5}$$

The basis function $\bar{\phi}$ denotes the mean brightness transfer function and $\mathbf{c} = \{c_i\}$ contains the weights c_i of the corresponding basis functions ϕ_i. Depending on the task, there are many different basis functions which can be used and it is even possible to learn them. Once the basis functions are fixed, the weights c_i can be computed by solving the equations with the brightness values of the two images.

2.5 MODEL FOR MOVEMENT TYPES

The optical flow $(u(x,y,t), v(x,y,t))$, further defined in the next Section 2.3, can be expressed with a parameterization model like the illumination changes. Nir et al. [12] proposed a general over-parameterized space-time model for this purpose. Two sets of n basis functions $\phi_i : \mathbb{R} \to \mathbb{R}$ and $\eta_i : \mathbb{R} \to \mathbb{R}$ are used to describe each displacement direction $u(x,y,t)$ and $v(x,y,t)$ with their own basis functions. The optical flow can then be expressed as follows:

$$
\begin{aligned}
u(x,y,t) &= \sum_{i=1}^{n} A_i(x,y,t) \cdot \phi_i(x,y,t) \ , \\
v(x,y,t) &= \sum_{i=1}^{n} B_i(x,y,t) \cdot \eta_i(x,y,t) \ .
\end{aligned}
\tag{2.6}
$$

The weights A_i respectively B_i are space and time varying coefficients for the corresponding basis functions. Depending on the type of movement, different basis functions can be chosen to express it or they can be learned first. After having the basis functions fixed, the weights A_i and B_i can be computed like the weights for the illumination change basis functions by solving the equations.

3. BENCHMARKS FOR THE OPTICAL FLOW PROBLEM

This thesis investigates three of the most used benchmarks for analysing the performance of optical flow methods: the Middlebury Benchmark, the KITTI Vision Benchmark Suite and the MPI-Sintel Benchmark. These benchmarks were designed to evaluate the optical flow algorithms by using challenging difficulties. This way, it is possible to tell how good the methods really work, especially since the benchmarks try to simulate real life circumstances. Another reason for designing these benchmarks is to have a common basis to compare the algorithms. The importance of designing such benchmarks for various problems as well as their chances and the challenges that may arise have been explained in detail in Section 1.3. Each benchmark consists basically of two data sets. The first one is the training data set and includes images of the first frame, the second frame and the corresponding ground truth flow, that means the correct optical flow, either measured or computed. This way a comparison between the given solution and the solution computed by the algorithm to be tested is possible. Another use of the training data set is for training methods to use learned parameters later on the testing data set. For the real evaluation purpose, the second data set is for testing the performance of the optical flow methods and doesn't contain the ground truth flow. That means that only the first and the second frame are given from which a solution has to be computed. The evaluation of the computed optical flow is typically done by a separate program or on the corresponding website. In the following chapter, these benchmarks are introduced by summarizing their database design and evaluation methodologies.

3.1 THE MIDDLEBURY BENCHMARK

The Middlebury Benchmark is the first of of the three evaluated benchmarks. It consists basically of both non-synthetic and synthetic data. More specifically said it contains four types of data: non-synthetic sequences with hidden fluorescent texture, realistic synthetic sequences, interpolation frames from high frame-rate videos and modified stereo sequences [1]. The four data sets consist of colour images, although the grey value versions of these images exist there as well. Considering the number of image pairs, there are 12 training image sequences with corresponding ground truth and 12 testing image sequences without ground truth for the evaluation purpose. They tried to capture eight frames for each image sequence for which the ground truth flow is between the middle pair. The four data sets are explained more detailed below. For the evaluation of optical flow methods, both a relative and an absolute error metric have been chosen from which some statistics have been computed using special

Figure 3.1: An image sequence of the first category of the Middlebury Benchmark where the ground truth flow has been computed via a hidden fluorescent texture. **Left:** First frame. **Middle:** Second Frame. **Right:** Corresponding ground truth flow. [1]

region masks. The benchmark data as well as the evaluation of several optical flow algorithms can be found on the web at http://vision.middlebury.edu/flow/. For further information about this benchmark please read [1].

3.1.1 *Database Design*

The first data set contains real image sequences where the ground truth flow is computed via hidden fluorescent texture. A scene has been built on a stage which can be moved by a computer. Furthermore, this scene is covered everywhere by a pattern of fluorescent paint in different colours. This way a pair of pictures is taken both under ambient lighting and under UV lighting when the stage doesn't move. After that, the stage moves a bit and another pair of pictures is taken. By tracking the fluorescent paint droplets in the UV lighting images, a quite accurate and dense ground truth flow can be computed from this for the corresponding ambient lighting images. Occluded regions can be detected by using crosschecking. An example of such a scene together with the corresponding ground truth can be seen in figure 3.1. The maximal motion in the image sequences can be up to 12 pixels. The advantages of this data set are the non-rigid motion and, due to using a real camera and a real scene, realistic photometric effects. Nevertheless, some regions may be occluded and therefore there is no ground truth available for these regions. Additionally, the pictures aren't taken of real-world scenes but of artificial scenes in a lab and the stage is temporarily stopped during motion.

Because of the disadvantages of the first data set, especially considering the ground truth flow, the second data set consists of realistic synthetic image sequences. With the help of computer graphics, the complexity of the scenes can be adapted easily and individually, for example the choice of the camera motion or object textures. This way, two types of complex synthetic outdoor scenes have been generated, natural and urban scenes, which can be seen in figure 3.2. The natural scenes contain trees with a bit of wind motion and rocks. Appearing occlusions together with the texture of

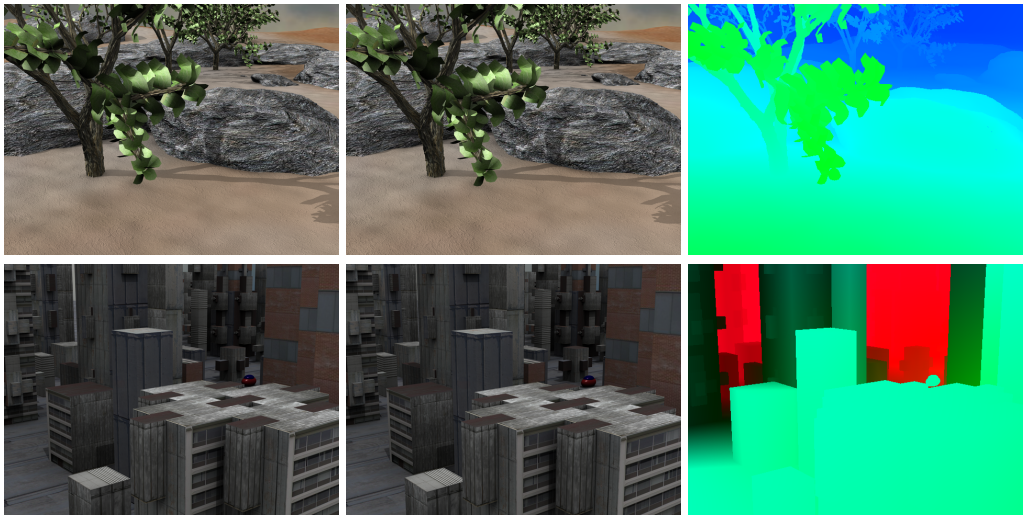

Figure 3.2: Image sequences of the second category of the Middlebury Benchmark which consist of realistic synthetic images. The upper row shows a natural scene while the images in the lower row are from an urban scene. **Left column:** First frame. **Middle column:** Second Frame. **Right column:** Corresponding ground truth flow. [1]

the objects increase the difficulty of these scenes. The urban scenes consists of some moving cars and buildings with different shapes and textures. During the rendering process of these scenes, the ground truth flow can be computed easily and nearly perfect. Therefore a larger motion range is used of up to 35 pixels. Various illumination settings increase the difficulty of all of these synthetic scenes further.

The third data set is only discussed briefly since it handles the problem of frame interpolation instead of the optical flow problem. For this kind of problem, it is more important to estimate a good intermediate frame rather than to compute an optical flow which matches the ground truth flow as good as possible. Therefore, cameras with a high frame rate have been chosen to capture real scenes such that each second image is used as ground truth intermediate frame. This way, the remaining images form the data set to evaluate frame interpolation algorithms. Nevertheless, these image sequences can be used for optical flow as well, although there doesn't exist a ground truth flow. Figure 3.3 shows an example for this type of data set.

The last data set of the Middlebury Benchmark contains modified stereo data which is shown in figure 3.4. That means that this data set uses images which were originally used for stereo matching and for which the ground truth is computed via structured lighting by Scharstein and Szeliski [13]. To adapt these images to the optical flow problem, only some subregions of them have been used and the disparity range had to be shifted. The advantage of these image sequences is that, like for the first data set

Figure 3.3: An image sequence of the third category of the Middlebury Benchmark which belongs to the frame interpolation problem and is captured from real scenes with high frame rate cameras. The ground truth flow isn't shown in this image since it doesn't exist for this category. **Left:** First frame. **Right:** Second Frame. [1]

with the hidden fluorescent pattern images, a real scene and a real camera have been used which increases the realism. Furthermore, it is possible to compare optical flow algorithms with stereo algorithms this way.

3.1.2 *Evaluation Methodologies*

The Middlebury Benchmark not only evaluates optical flow algorithms but also frame interpolation methods. Therefore, they suggest and report error metrics for each of these issues. Since this thesis is focused on the optical flow topic, only the relevant error measures are explained here. They mention first the performance measures and then how they use them to get error statistics from them. Furthermore, these error statistics are reported for three types of region masks.

Regarding the performance measures, two error metrics were used. The first one is the angular error which was formerly one of the most commonly used error metrics for evaluating optical flow algorithms. It computes the angle in 3D space between two flow vectors, that means in the case of error computation between a computed flow vector and the corresponding ground truth flow vector. Fleet and Jepson [14] introduced this measurement method while Barron et al. [6] made it popular in their paper. The angular error between the vectors (u_0, v_0) and (u_1, v_1) can be computed by first appending a 1 to each vector such that they are three-dimensional and then using the following formula:

$$AE = \cos^{-1}\left(\frac{1 + u_0 * u_1 + v_0 * v_1}{\sqrt{1 + u_0^2 + v_0^2} * \sqrt{1 + u_1^2 + v_1^2}}\right) . \tag{3.1}$$

Figure 3.4: An image sequence of the fourth category of the Middlebury Benchmark where the images of a stereo scene have been modified such that the computation of optical flow can be applied. **Left:** First frame. **Middle:** Second Frame. **Right:** Corresponding ground truth flow. [1]

Since the error penalisation is dependent on the length of the flow vector, the angular error yields a relative error measurement. Due to this reason, Baker et al. [1] additionally use an absolute measurement of performance, namely the error in flow endpoint which is a very common error measure nowadays. It has been introduced by Otte and Nagel [7] and the corresponding formula for vectors (u_0, v_0) and (u_1, v_1) is defined as follows:

$$EE = \sqrt{(u_0 - u_1)^2 + (v_0 - v_1)^2} \ . \tag{3.2}$$

By using and reporting both error measurements, Baker et al. [1] were able to compare the evaluated methods in two ways.

The averages and standard deviations, like Barron et al. [6] did, some robustness statistics similar to those of Scharstein and Szeliski [15] as well as robust accuracy measures introduced by Seitz et al. [16] have been computed from the before mentioned error measurements. They used for the robustness statistics the notation of RX for the percentage of pixels with an error value of X or worse. This way, they computed for the angular error statistics for R2.5, R5.0 and R10.0 (value X in degrees) and for the endpoint error statistics for R0.5, R1.0 and R2.0 (value X in pixels) and reported them. The robust accuracy measures were defined by the notation AX where the errors have been sorted and then the Xth percentile has been used to compute its accuracy of the error measure. For this case, they used A50, A75 and A95.

The difficulty to compute optical flow is amongst others dependent on the region in the image where the flow is computed. Therefore the before mentioned error measure statistics are computed for the whole image as well as for some specific difficult regions. To implement this, the three region masks proposed by Scharstein and Szeliski [15] have been chosen where the flow is computed either everywhere or only in tex-

tureless regions or only around motion discontinuitites. Only pixels with a ground truth flow have been considered for all of these regions.

3.2 THE KITTI VISION BENCHMARK SUITE

The KITTI Vision Benchmark Suite consists basically of two benchmarks which are different versions of the same raw data. The KITTI 2012 Benchmark was designed to evaluate methods regarding stereo matching, optical flow visual odometry and 3D object detection while the KITTI 2015 Benchmark was created for the purpose of comparing scene flow algorithms, although it can be used for stereo matching and optical flow as well. These two versions are handled as two separate benchmarks and evaluated each on its own. The raw data set contains images which were captured while an automotive driving vehicle is navigating through a city. Both benchmarks consist therefore only of real data of real-world scenes because of which the data is realistic. The KITTI 2012 Benchmark includes 194 training and 195 testing colour and grey value image sequences while the KITTI 2015 Benchmark's contains 200 training and 200 testing colour image pairs. There are two absolute error metrics used for the evaluation of optical flow methods from which some statistics have been computed for two types of regions. The benchmark data can be downloaded on the web from http://www.cvlibs.net/datasets/kitti/ while the results of several optical flow algorithms are listed there as well. Further information can be found in [2] about the KITTI 2012 Benchmark, in [3] about the KITTI 2015 Benchmark and in [4] about the raw data set.

3.2.1 *Database Design*

As mentioned before, the images for both KITTI Benchmarks are obtained by capturing them during the car ride of an automotive driving vehicle. This car is equipped with several cameras, sensors, a laser scanner and a localization system which allow the computation of an accurate ground truth flow. The ground truth flow of the KITTI 2015 Benchmark was even improved by computing it separately for the static background and the independently moving objects for which a fitting system of models was used before all the flow information were combined. However, a problem of the KITTI benchmarks is that their ground truth flows are only sparse which means that many points don't have a flow vector. The result of using the whole system is real-world image data with realistic illumination and effects due to the fact that the images have been taken in the real world. A big challenge was the calibration of the whole system beforehand to make everything possible. After obtaining the raw data, a representative selection of it has been chosen for the benchmarks. Several examples of such scenes are shown in figure **??**. There are some black regions and several black dots in the ground truth images which means that there is no ground truth flow defined

Figure 3.5: Image sequences of the KITTI Vision Benchmark Suite. The two upper rows are from the KITTI 2012 Benchmark while the two bottom rows belong to the KITTI 2015 Benchmark. **Left column:** First frame. **Middle column:** Second Frame. **Right column:** Corresponding ground truth flow. [2], [3]

at these points. The fitted car models for which the ground truth flow is computed separately can be seen very good in the two ground truth images of the KITTI 2015 Benchmark.

3.2.2 *Evaluation Methodologies*

In contrast to the Middlebury Benchmark [1], there are two absolute error measures suggested for the KITTI Vision Benchmark Suite [2] which are used in both benchmarks. The first one is the average number of mismatched pixels considering disparity while the second one is the endpoint error defined above. To report the performance measurement, some threshold values have been chosen for both of the two error metrics to compute statistical values from them, namely $\tau \in \{2, \ldots, 5\}$. Two regions were used for the computation of the statistics. All pixels with a ground truth flow belong to the first region whereas only non-occluded pixels with a ground truth flow have been considered for the second case.

3.3 THE MPI-SINTEL BENCHMARK

The last of the three benchmarks is the MPI-Sintel Benchmark. Although it only consists of synthetic data, it claims to be quite complex and realistic since the data is gathered by rendering the 3D animated short film *Sintel*. Furthermore, Butler et al. [5] prove this via a comparison with data from real films. All of the images of this data set are colour images, although there exist three versions of different complexity of

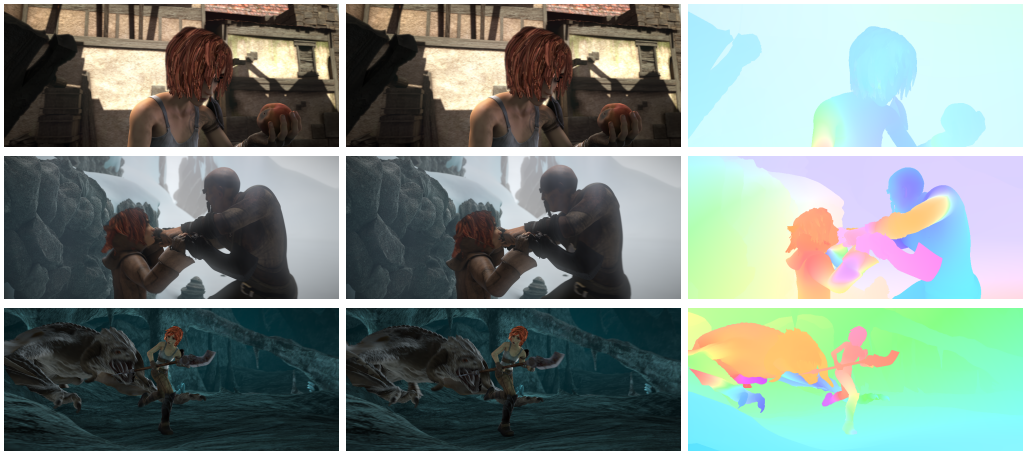

Figure 3.6: Image sequences of the MPI-Sintel Benchmark which are captured by rendering the 3D animated short film *Sintel*. **Left column:** First frame. **Middle column:** Second Frame. **Right column:** Corresponding ground truth flow. [5]

these images. The data set consists of 35 image sequences where each image sequence, except from some shorter ones, contains approximately 50 frames. The ground truth flow exists for every consecutive image pair, that means that there are about 49 image pairs per image sequence available. Therefore the number of frames for the training data set is 1064 from 23 image sequences with corresponding ground truth flow while the testing data set contains 564 frames from 12 image sequences without included ground truth flow. Hence, the MPI-Sintel Benchmark is the benchmark with the biggest amount of data of these three benchmarks. The evaluation of optical flow methods is done using only an absolute error metric but computing it for three region types. Furthermore, the error is gathered for some statistics from which error functions are computed. The benchmark data is together with the evaluation results of several optical flow algorithms available on the web at http://sintel.is.tue.mpg.de/. For further information about this benchmark please read [5] or their supplemental material [9].

3.3.1 *Database Design*

Sintel is an open source movie which means that both the film and all the graphic data from it are freely available for everyone. This allows that the movie can be rendered anew with different camera settings or effects by using its source files. Using this fact, Butler et al. [5] created the MPI-Sintel Benchmark from several scenes of this film which can be seen for example in figure 3.6. The chosen clips had to fulfill the requirement that the optical flow in these scenes was well defined. For this to happen, some settings were adjusted differently than in the movie. Furthermore, to prevent cheating since all the data is available, some scenes have been perturbed slightly with random

Figure 3.7: The three versions of the previous image sequences which were acquired by using different render passes. [5] **Left column:** Albedo pass. **Middle column:** Clean pass. **Right column:** Final pass.

offsets. This way it is possible to detect fraud. For locating ground truth motion boundaries and pixels that appear only in one image of the image pair, so called unmatched pixels, Butler et al. acted on the assumption that the affected pixels belong to a set of physical boundaries. Therefore, all relevant boundaries have been computed and their union is claimed to be an overestimated guess of the location of motion boundaries.

The benchmark offers three levels of difficulty which act like diverse data sets. They have been obtained by choosing render passes which come with different effects. Some example images after using one of the three render passes are shown in figure 3.7. The first rendering pass is called Albedo pass and is characterized by having nearly piecewise constant colours without any illumination effects from which the grey value constancy assumption should benefit. Therefore, this pass is the simplest pass. The Clean pass includes some illumination like shadows and reflections which increases the difficulty of this pass. The images which are the results of rendering the graphic data using the last pass resemble the images from the short film. The reason for this is that in the most complex pass, the Final pass, blurring and atmospheric effects have been added. A full overview of all effects that have been used for the passes can be found in [5].

3.3.2 *Evaluation Methodologies*

The first error metric used in the MPI-Sintel Benchmark [5] is, like for the previous two benchmarks, the average endpoint error. Furthermore, the error has been represented as functions. The endpoint error metric has been computed for three regions, namely

for all pixels, for all pixels that appear in both images and for all pixels that appear only in one of the images. All of the concidered pixels required to have a ground truth flow. In addition to computing the error for specific regions, the error has been reported for different speeds and for some distances towards the nearest occlusion boundary. The speed categories in pixels/frame (ppf) were ≤ 10 ppf, $10 - 40$ ppf and ≥ 40 ppf while the distances were classified into ≤ 10 pixels, $10 - 60$ pixels and ≥ 60 pixels. For the function representation, a function of speed on the one hand and a function of distance to occlusion boundaries on the other hand have been used.

3.4 SUMMARY

The four benchmarks can be summarised by the following table:

Characteristics	Data Type	Grey/Coloured	# Image Pairs	Ground Truth
Middlebury	non-synthetic and synthetic	coloured	$8 + 12$	quite dense
KITTI 2012	real-world	grey	$194 + 195$	sparse
KITTI 2015	real-world	coloured	$200 + 200$	sparse
MPI-Sintel	synthetic	coloured	$1064 + 564$	dense

4. METRICS

The main task of this thesis is the evaluation of four optical flow benchmarks and the introduction of an evaluation methodology for other benchmarks regarding the optical flow problem. Optical flow benchmarks, like other benchmarks as well, consist of difficulties and regularities which have to be analyzed for this task. Therefore, several metrics are going to be used for the evaluation which consider different observable categories. These metrics suggest the analysis of various topics which could be important for both the evaluation of existing benchmarks and the creation of new ones. This section introduces the metrics to be evaluated while the question about the usefulness and the significance of the proposed metrics is answered in the subsequent Section 5 where the results of the evaluation are presented. The suggested metrics belong to the categories of image statistics, optical flow statistics, illumination changes from the first frame to the next one, the movement between them and how similar the observed scene is to a stereo scene because of egomotion. Most of the metrics can only be applied to images with given ground truth flow of the training data sets which are used as an indicator for the difficulty. Furthermore, the computed flow is used for the testing data sets to estimate the difficulty of these image sequences. However, this means that the results depend partially on the quality of the computed flow used for the evaluation. Some of these metrics are introduced first in a continuous way from which a discrete version is derived to apply it on the images.

4.1 IMAGE STATISTICS

Image statistics consider only the information contained in one image. The difficulty of the images themselves should be measured without considering the optical flow by using them. The information contained in the images are the grey values respectively the colour values at each pixel. These statistics are independent of the ground truth flow and can be applied to every image. However, it is difficult to derive metrics from them to analyse the difficulty. Therefore, the image statistics are regarded as additional information about the whole sequence rather than as a measure of difficulty.

The first statistics which are collected are the maximal, the minimal and the average grey respectively colour value and the corresponding variance respectively standard deviation. For the colour values, each channel is regarded separately. To simplify notations, both grey values and colour values are abbreviated as pixel values. Using these statistics, some things can be said about the considered image. From the minimal and the maximal pixel value, the range can be determined which gives an idea of how many different pixel values might appear. The average brightness of the image can

be derived from the average pixel value while the variance respectively the standard deviation give an overview of the distribution of the pixel values.

After getting a rough overview about the image, a histogram gives more details about the real distribution of the pixel values where each pixel value is counted separately for the histogram. After having created the histogram, it is possible to see whether the pixel values are well distributed or whether some pixel values dominate. Furthermore, it is obvious whether the image is rather bright or dark. As mentioned before, these are just some additional information to evaluate the images but doesn't necessarily say something about the difficulty of the image sequence.

4.2 OPTICAL FLOW STATISTICS

The category of optical flow statistics contains metrics which evaluate the difficulty of the optical flow itself without including any further assumptions. Therefore, the optical flow is evaluated independently from the corresponding image. As defined in Section 2.3, the optical flow consists of two parts, namely the x-displacement $u(x,y,t)$ and the y-displacement $v(x,y,t)$ which are going to be used for these metrics.

4.2.1 *Magnitude Statistics*

The magnitude statistics consider the length of an optical flow vector at a point (x,y,t) which is defined as $\left|(u(x,y,t),v(x,y,t))^{\mathrm{T}}\right| = \sqrt{u(x,y,t)^2+v(x,y,t)^2}$. The first categories of the magnitude statistics that can be measured for each image pair are the maximal length, the minimal length, the average length and the variance respectively the standard deviation. These values can only give a very rough overview about the optical flow of the considered image sequence. Nevertheless, these values can be used to estimate the length of the occurring displacements. This information is important to get an overview of how long optical flow vectors might be since large displacements increase the difficulty significantly while having only small displacements can decrease the search range to find the corresponding points in the other image. The larger the range between the minimal length and the maximal length, the more different lengths might probably exist. A high average length means that there are more long displacements than short ones. And the standard deviation can give an idea of how well the lenghts are distributed around the average length. This means that image sequences with a high range between minimal and maximal length, a high average length and a wide distribution are more difficult than other sequences. The testing image sequences can be only roughly characterized by using computed flow to determine the difficulty.

The second possibility to use the lengths of the optical flow vectors is to group them into bins of a certain size. This way, a histogram can be created by counting the points of which the optical flow has a specific value. This gives a more detailed overview of

the distribution of the optical flow vector's lengths. In this case it is also possible to estimate the distribution for the testing image sequences by using a computed flow field. Furthermore, the lenghts with the highest occurences become visible by using a histogram.

4.2.2 *Angle Statistics*

The angle statistics for optical flow are similar to the ones for the magnitude, but instead of using the length of an optical flow vector the angle is computed. Since the angle has to be computed between two vectors, the reference vector $(1,0)^{\mathrm{T}}$ is used. The angle φ between two vectors a and b is defined in the following way:

$$\cos(\varphi) = \frac{\langle a,b\rangle}{|a|\cdot|b|} \Leftrightarrow \varphi = \arccos\left(\frac{\langle a,b\rangle}{|a|\cdot|b|}\right) \quad . \tag{4.1}$$

However, the resulting values range from 0 to π and are invariant from orientation. Therefore, it is necessary to include the orientation of the optical flow vector such that the values range from 0 to 2π. The resulting value can then be converted to degrees. The angle increases counterclockwise. An angle of 0° belongs to a vector pointing from the origin in positive x-direction, an angle of 90° is pointing in positive y-direction and so on. A big problem of the angle statistics is that values of $(360^\circ + x^\circ)$ with $x \geq 0^\circ$ are equal to $(0^\circ + x^\circ)$ which means that the angle is a circular measurement. That means that a vector with an angle of 2° is closer to a vector with an angle of 359° than to a vector with an angle of 15°. This is the reason why the standard versions of statistics like minimum, maximum, average and the variance respectively the standard deviation can't be used that easily for angles. Instead of that a circular mean can be computed which points in the main direction of all considered angles. More information about this can be found in [17].

The only other possibility to compute angle statistics is to create a histogram. Using bins with a size of 1° can give a precise overview of the optical flow angles. If larger bins are used, e.g. of size 10° or higher, the main directions can be observed from the histogram. This way it is possible to tell if and how many main directions exist or if the angles are well distributed.

4.3 ILLUMINATION

The metrics of the illumination category consider mainly illumination changes between the two images of the image sequence. Therefore, both the image information, that means the grey values respectively the colour values of each pixel of the two images, and the optical flow, that means the x- and the y-displacements of each pixel, are important for these metrics since the image information of both images has to be

used together. To do this, the corresponding pixels of both images have to be identified using the optical flow. The ground truth flow is used for the training data sets while the difficulty regarding illumination of the testing data sets is investigated with the help of computed flow.

4.3.1 *Pixel Value Comparison*

The pixel value comparison belongs to the category illumination because it compares the brightness of the two images. This metric is rather weak but still gives some information about the difficulty of the considered image sequence. The first possibility to compare the brightness of the images is by comparing the maximal, the minimal and the average pixel values of them. If large differences between these values are observed, it can be assumed that the illumination changes a lot from one image to the other. This can be proven more thoroughly by comparing the images' histograms. By doing this, the number of occurences of each pixel value can be compared this way. However, it is not possible to say which pixel value of the first image changed to which pixel value of the second image.

The other possibility to compare the two images is by warping the second image back and then subtracting both images from each other. It is assumed that the second image is basically equal to the first image but where optical flow has been applied. This is just the definition of optical flow which is mathematically defined as follows:

$$f(x,y,t) = f(x+u(x,y,t), y+v(x,y,t), t+1) \ . \tag{4.2}$$

By compensating the optical flow from the second image, it is possible to warp it back such that it should become the first image again. This means that the optical flow has to be subtracted from the second image. However, this is only possible for all the points where the optical flow is defined. After warping the second image back, the absolute value of the difference of the two images can now be computed at each point where the optical flow is defined. By doing this, a difference image is generated where the brightness change can be observed. This allows to determine the average and the standard deviation of the absolute brightness difference by computing the mean and the standard deviation of the difference image. The larger the average difference the higher the probability that illumination changes appear and the higher the illumination changes themselves.

4.3.2 *Illumination Changes*

A parameterized model to describe illumination changes has been proposed in Section 2.4 and is defined in equation 2.5. The basis functions that have been chosen for the

evaluation and are used for the remainder of this thesis are the ones that fit the affine model of Negahdaripour [18]:

$$\bar{\phi}(f) = 0, \quad \phi_1(f) = f, \quad \phi_2(f) = 1 \ . \tag{4.3}$$

Having defined the basis functions, the relation between the first frame and the second frame of an image sequence can be defined as follows

$$\bar{\phi}(f_1) + c_1 \cdot \phi_1(f_1) + c_2 \cdot \phi_2(f_1) = c_1 \cdot f_1 + c_2 = f_2 \ , \tag{4.4}$$

where f_1 and f_2 are the two frames of the image sequence respectively they represent a point in the corresponding frame. Since the observed scene isn't necessarily static, that means that objects might move, the optical flow has to be integrated in this equation. Furthermore, the pixel notation is used for the full equation to describe the illumination change between the two frames:

$$c_1 \cdot f(x, y, t) + c_2 = f(x + u(x, y, t), y + v(x, y, t), t + 1) \ . \tag{4.5}$$

For the evaluation of the benchmarks either the ground truth flow or a computed flow is given. That means that it is known which point of the first frame belongs to which point in the second frame. Using a similar notation like in equation 4.4, the discrete equation for each point (i, j) is simplified as follows

$$c_{1_{i,j}} \cdot f_{i,j}^t + c_{2_{i,j}} = f_{i,j}^{t+1} \ , \tag{4.6}$$

where $f_{i,j}^t$ denotes a point in the first frame and its corresponding point in the second frame is defined as $f_{i,j}^{t+1}$. Since the weights of the basis functions might be different for every point, the notation of them is adapted such that they correspond to a specific point: $c_{1_{i,j}}$ and $c_{2_{i,j}}$.

4.3.2.1 *Global Joint Illumination Changes*

The category of global joint illumination changes is defined in such a way that the illumination change from one frame to the other is the same for every point in the image. In terms of the definition above of illumination changes, the weights $c_{1_{i,j}}$ and $c_{2_{i,j}}$ are the same for all N points in the first image which can be matched to a point in the second image. Thus, the indices i and j can be dropped, making the illumination change global. A point in a colour value image has different channels, e.g. a red, a green and a blue channel. In this case, it is assumed that the illumination change is the same for all the channels such that it is a joint illumination change. Grey value images are not affected by this assumption. Defining global joint illumination changes as an equation for continuous images f leads towards

$$\begin{aligned} c_1 \cdot f(x,y,t) + c_2 = f(x + u(x,y,t), y + v(x,y,t), t+1) \Leftrightarrow \\ c_1 \cdot f(x,y,t) + c_2 - f(x + u(x,y,t), y + v(x,y,t), t+1) = 0 \ . \end{aligned} \tag{4.7}$$

The second part of this equation can be used to solve it for the weights c_1 and c_2. For this purpose, a continuous least squares energy function $\mathrm{E}(c_1, c_2)$ is defined which has to be minimized to get the best possible weights.

$$\mathrm{E}(c_1, c_2) = \int_\Omega \left(c_1 \cdot f(x,y,t) + c_2 - f(x + u(x,y,t), y + v(x,y,t), t+1)\right)^2 \mathrm{d}x\mathrm{d}y \ . \tag{4.8}$$

Since the optical flow probably isn't defined at each point of the image, the energy function is discretised by evaluating the images at pixels (i, j) at the corresponding time steps t and $t+1$.
The discrete version of the energy function is then written as

$$\mathrm{E}(c_1, c_2) = \sum_{i=1}^{N} \sum_{j=1}^{M} \left(c_1 \cdot f_{i,j}^t + c_2 - f_{i,j}^{t+1}\right)^2 \ . \tag{4.9}$$

This energy functional treats every point the same way, irrespective of the possibility of having outliers or noise which might have a bad influence on the result. To weight these outliers down, the following subquadratic penaliser is used

$$\psi(s^2) = 2 \cdot \sqrt{s^2 + \lambda^2} \ . \tag{4.10}$$

with a small regularisation parameter $\lambda > 0$. The properties of a penaliser are that it must be positive, increasing and strictly convex to allow an unique solution. By applying the penaliser to the energy function, it becomes robust with respect to outliers and noise. Thus, the energy function reads

$$\mathrm{E}(c_1, c_2) = \sum_{i=1}^{N} \sum_{j=1}^{M} \psi\left(\left(c_1 \cdot f_{i,j}^t + c_2 - f_{i,j}^{t+1}\right)^2\right) \ . \tag{4.11}$$

By using $\psi(s^2) = s^2$ as penalisation function, this energy function becomes the one from equation 4.9 again. To minimize this energy function, the derivatives of it with respect to the two weights c_1 and c_2 are computed:

$$\begin{aligned} \partial_{c_1} \mathrm{E}(c_1, c_2) &= 2 \cdot \sum_{i=1}^{N} \sum_{j=1}^{M} \psi'\left(\left(c_1 \cdot f_{i,j}^t + c_2 - f_{i,j}^{t+1}\right)^2\right) \cdot \left(c_1 \cdot f_{i,j}^t + c_2 - f_{i,j}^{t+1}\right) \cdot f_{i,j}^t = 0 \ , \\ \partial_{c_2} \mathrm{E}(c_1, c_2) &= 2 \cdot \sum_{i=1}^{N} \sum_{j=1}^{M} \psi'\left(\left(c_1 \cdot f_{i,j}^t + c_2 - f_{i,j}^{t+1}\right)^2\right) \cdot \left(c_1 \cdot f_{i,j}^t + c_2 - f_{i,j}^{t+1}\right) \cdot 1 = 0 \ . \end{aligned} \tag{4.12}$$

The derivative of the penalisation function is defined as follows:

$$\psi'(s^2) = \frac{1}{\sqrt{s^2 + \lambda^2}} \ . \tag{4.13}$$

To simplify these and the following equations the following abbreviation is used:

$$\psi'_{i,j} = \psi' \left(\left(c_1 \cdot f^t_{i,j} + c_2 - f^{t+1}_{i,j} \right)^2 \right) \ . \tag{4.14}$$

The weights c_1 and c_2 appear in the inner term of the penalisation function. This would make the computation of them more difficult. Therefore, the so-called lagged nonlinearity method is used which is explained below. For the following equations, the penalisation term is assumed to be fixed. Thus, this leads to the following equations:

$$\begin{aligned} \left(\sum_{i=1}^{N} \sum_{j=1}^{M} \psi'_{i,j} \cdot {f^t_{i,j}}^2 \right) \cdot c_1 + \left(\sum_{i=1}^{N} \sum_{j=1}^{M} \psi'_{i,j} \cdot f^t_{i,j} \right) \cdot c_2 &= \sum_{i=1}^{N} \sum_{j=1}^{M} \psi'_{i,j} \cdot f^t_{i,j} \cdot f^{t+1}_{i,j} \ , \\ \left(\sum_{i=1}^{N} \sum_{j=1}^{M} \psi'_{i,j} \cdot f^t_{i,j} \right) \cdot c_1 + \left(\sum_{i=1}^{N} \sum_{j=1}^{M} \psi'_{i,j} \right) \cdot c_2 &= \sum_{i=1}^{N} \sum_{j=1}^{M} \psi'_{i,j} \cdot f^{t+1}_{i,j} \ . \end{aligned} \tag{4.15}$$

Solving these equations can be done by solving the following 2×2 linear system of equations for the two variables c_1 and c_2:

$$\begin{pmatrix} \sum_{i=1}^{N} \sum_{j=1}^{M} \psi'_{i,j} \cdot {f^t_{i,j}}^2 & \sum_{i=1}^{N} \sum_{j=1}^{M} \psi'_{i,j} \cdot f^t_{i,j} \\ \sum_{i=1}^{N} \sum_{j=1}^{M} \psi'_{i,j} \cdot f^t_{i,j} & \sum_{i=1}^{N} \sum_{j=1}^{M} \psi'_{i,j} \end{pmatrix} \cdot \begin{pmatrix} c_1 \\ c_2 \end{pmatrix} = \begin{pmatrix} \sum_{i=1}^{N} \sum_{j=1}^{M} \psi'_{i,j} \cdot f^t_{i,j} \cdot f^{t+1}_{i,j} \\ \sum_{i=1}^{N} \sum_{j=1}^{M} \psi'_{i,j} \cdot f^{t+1}_{i,j} \end{pmatrix} \ . \tag{4.16}$$

Cramer's rule is applied to this linear system of equations such that the solution for the variables c_1 and c_2 can be derived by the following equations:

$$\begin{aligned} c_1 = &\frac{\left(\sum_{i=1}^{N} \sum_{j=1}^{M} \psi'_{i,j} \right) \cdot \left(\sum_{i=1}^{N} \sum_{j=1}^{M} \psi'_{i,j} \cdot f^t_{i,j} \cdot f^{t+1}_{i,j} \right)}{\left(\sum_{i=1}^{N} \sum_{j=1}^{M} \psi'_{i,j} \right) \cdot \left(\sum_{i=1}^{N} \sum_{j=1}^{M} \psi'_{i,j} \cdot {f^t_{i,j}}^2 \right) - \left(\sum_{i=1}^{N} \sum_{j=1}^{M} \psi'_{i,j} \cdot f^t_{i,j} \right)^2} \\ &- \frac{\left(\sum_{i=1}^{N} \sum_{j=1}^{M} \psi'_{i,j} \cdot f^t_{i,j} \right) \cdot \left(\sum_{i=1}^{N} \sum_{j=1}^{M} \psi'_{i,j} \cdot f^{t+1}_{i,j} \right)}{\left(\sum_{i=1}^{N} \sum_{j=1}^{M} \psi'_{i,j} \right) \cdot \left(\sum_{i=1}^{N} \sum_{j=1}^{M} \psi'_{i,j} \cdot {f^t_{i,j}}^2 \right) - \left(\sum_{i=1}^{N} \sum_{j=1}^{M} \psi'_{i,j} \cdot f^t_{i,j} \right)^2} \ , \end{aligned} \tag{4.17}$$

$$c_2 = \frac{\left(\sum_{i=1}^{N}\sum_{j=1}^{M}\psi'_{i,j}\cdot {f^t_{i,j}}^2\right)\cdot\left(\sum_{i=1}^{N}\sum_{j=1}^{M}\psi'_{i,j}\cdot f^{t+1}_{i,j}\right)}{\left(\sum_{i=1}^{N}\sum_{j=1}^{M}\psi'_{i,j}\right)\cdot\left(\sum_{i=1}^{N}\sum_{j=1}^{M}\psi'_{i,j}\cdot {f^t_{i,j}}^2\right)-\left(\sum_{i=1}^{N}\sum_{j=1}^{M}\psi'_{i,j}\cdot f^t_{i,j}\right)^2} - \frac{\left(\sum_{i=1}^{N}\sum_{j=1}^{M}\psi'_{i,j}\cdot f^t_{i,j}\right)\cdot\left(\sum_{i=1}^{N}\sum_{j=1}^{M}\psi'_{i,j}\cdot f^t_{i,j}\cdot f^{t+1}_{i,j}\right)}{\left(\sum_{i=1}^{N}\sum_{j=1}^{M}\psi'_{i,j}\right)\cdot\left(\sum_{i=1}^{N}\sum_{j=1}^{M}\psi'_{i,j}\cdot {f^t_{i,j}}^2\right)-\left(\sum_{i=1}^{N}\sum_{j=1}^{M}\psi'_{i,j}\cdot f^t_{i,j}\right)^2}\,. \tag{4.18}$$

If the penaliser $\psi(s^2) = s^2$ has been chosen, its derivative would be $\psi'(s^2) = 1$ which is why the equations above would give the final solution for the weights. In the other case, if the above defined subquadratic penaliser has been used, the equations system would be nonlinear instead of linear. To solve this problem, the lagged nonlinearity method is applied which fixes all the $\psi'_{i,j}$ like assumed before. This way, it is possible to solve a linear system of equations. A good initialisation for the weights of illumination changes to fix the penaliser term is to set $c_1 = 1$ and $c_2 = 0$ which is equal to no illumination change. The next step would be to recompute the $\psi'_{i,j}$ with the solution of the linear system of equations. By doing several iterations of alternating between solving the system of equations and recomputing the values of the penaliser, the solution can be approximated quite good. Global illumination changes where the multiplicative component c_1 is equal to 1 and the additive component c_2 is equal to 0 imply that there is no change in illumination at all and the brightness is in general preserved. Higher differences from these two values mean that an illumination change is more likely which can be used to estimate the overall brightness change of the images.

4.3.2.2 *Global Channelwise Illumination Changes*

The computation of the weights for the global channelwise illumination changes works nearly the same as the one for the global joint illumination changes. The only difference is that each of the channels of a coloured image has independent weights. In case of a RGB image, the discrete energy function would read as follows:

$$\begin{aligned}\mathrm{E}(c_{R_1}, c_{R_2}, c_{G_1}, c_{G_2}, c_{B_1}, c_{B_2}) = \sum_{i=1}^{N}\sum_{j=1}^{M} &\psi\left(\left(c_{R_1}\cdot f^t_{R_{i,j}} + c_{R_2} - f^{t+1}_{R_{i,j}}\right)^2\right)\\ &+\psi\left(\left(c_{G_1}\cdot f^t_{G_{i,j}} + c_{G_2} - f^{t+1}_{G_{i,j}}\right)^2\right)\\ &+\psi\left(\left(c_{B_1}\cdot f^t_{B_{i,j}} + c_{B_2} - f^{t+1}_{B_{i,j}}\right)^2\right)\,,\end{aligned} \tag{4.19}$$

where $c_{R_1}, c_{R_2}, c_{G_1}, \ldots$ denote the multiplicative and the additive components of the corresponding colour channels R, G and B. It is now possible with this energy function to compute the weights for each channel on its own by modifying the equations for the global joint illumination changes.

4.3.2.3 *Local Illumination Changes*

In contrast to the global illumination changes, the local illumination changes assume that the weights for the basis functions may be different for each point in the image. This means that each point has its own weights $c_{1_{i,j}}$ and $c_{2_{i,j}}$ which additionally includes separate weights for the colour channels. Since the case of grey scale images can be easily extended to the case of colour values, the derivations to compute the weights are limited to the grey scale images. The notation $\mathbf{c}_1$ and $\mathbf{c}_2$ is used for the function parameters and means that $\mathbf{c}_1$ and $\mathbf{c}_2$ include all the weights $c_{1_{i,j}}$ and $c_{2_{i,j}}$. It was easier to compute the weights for the global illumination changes than for the local illumination changes because it was possible to set up two equations for two unknown variables such that in the end a linear system could be solved. But since each point has its own weights, the energy functional has to be extended. Otherwise it wouldn't be possible to compute an unique solution for the weights. In contrast to the previous illumination change energy function, a functional is used here because the weights $\mathbf{c}_1$ and $\mathbf{c}_2$ are defined as continuous functions now which can be evaluated at each point. The energy functional for the local case consists of two terms, a data term and a smoothness term. A penalisation function can be applied to both the data term and the smoothness term. This case is described further down below. The data term is quite similar to the term used for the energy functional of the global case:

$$\mathrm{D}(\mathbf{c}_1, \mathbf{c}_2) = \left(c_1(x,y,t) \cdot f(x,y,t) + c_2(x,y,t) - f(x + u(x,y,t), y + v(x,y,t), t+1)\right)^2 \quad . \tag{4.20}$$

The smoothness term assumes that the values of the weights change only smoothly within a small neighbourhood such that neighbouring points of the image have similar weights. This can be done by penalising the gradient of the weights. This assumption leads to the following smoothness term:

$$\mathrm{S}(\mathbf{c}_1, \mathbf{c}_2) = \left(|\nabla \mathbf{c}_1(x,y,t)|^2 + |\nabla \mathbf{c}_2(x,y,t)|^2\right) \quad . \tag{4.21}$$

By using the abbreviations $\mathbf{c}_1$ for $c_1(x,y,t)$, $\mathbf{c}_2$ for $c_2(x,y,t)$, f^t for $f(x,y,t)$ and f^{t+1} for $f(x + u(x,y,t), y + v(x,y,t), t+1)$, the energy functional can then be defined with the regularisation parameter $\alpha > 0$ to control how big the influence of the smoothness term is:

$$\begin{aligned} \mathrm{E}(\mathbf{c}_1, \mathbf{c}_2) &= \int_\Omega \quad \mathrm{D}(\mathbf{c}_1, \mathbf{c}_2) \quad + \alpha \quad \mathrm{S}(\mathbf{c}_1, \mathbf{c}_2) \quad \mathrm{d}x\mathrm{d}y \\ &= \int_\Omega \gamma \left(\left(\mathbf{c}_1 \cdot f^t + \mathbf{c}_2 - f^{t+1} \right)^2 \right) + \alpha \left(|\nabla \mathbf{c}_1|^2 + |\nabla \mathbf{c}_2|^2 \right) \mathrm{d}x\mathrm{d}y \ , \end{aligned} \tag{4.22}$$

where Ω denotes the domain and γ is an indicator function which returns the argument if the optical flow is defined at the considered position and 0 otherwise. This

prevents the solver which is later proposed from using the data term where no flow is defined and fills in the information from the neighbouring points. The indicator function is dropped for the following explanations to ease the understanding of the methods. To minimize this type of energy functional, the Euler-Lagrange equations are derived from it in the following way:

$$\begin{aligned} 2\left(\mathbf{c}_1 f^t + \mathbf{c}_2 - f^{t+1}\right) f^t - 2\alpha(\mathbf{c}_1)_{xx} - 2\alpha(\mathbf{c}_1)_{yy} = 0 \ , \\ 2\left(\mathbf{c}_1 f^t + \mathbf{c}_2 - f^{t+1}\right) \quad - 2\alpha(\mathbf{c}_2)_{xx} - 2\alpha(\mathbf{c}_2)_{yy} = 0 \ , \end{aligned} \tag{4.23}$$

with the boundary conditions $\mathrm{n}^{\mathrm{T}}\nabla\mathbf{c}_1 = 0$ and $\mathrm{n}^{\mathrm{T}}\nabla\mathbf{c}_2 = 0$ where n is defined to be the normal vector. By using the definition of the Laplacian $\Delta a = a_{xx} + a_{yy}$, these equations can be simplified as follows:

$$\begin{aligned} \left(\mathbf{c}_1 f^t + \mathbf{c}_2 - f^{t+1}\right) f^t - \alpha\Delta\mathbf{c}_1 = 0 \ , \\ \left(\mathbf{c}_1 f^t + \mathbf{c}_2 - f^{t+1}\right) \quad - \alpha\Delta\mathbf{c}_2 = 0 \ . \end{aligned} \tag{4.24}$$

The next step in computing the weights is to discretise the equations above. The weights and the images are evaluated pointwise while the Laplacian is discretised in the following way:

$$\Delta c_1 = \sum_{l\in\{x,y\}} \sum_{\tilde{i},\tilde{j}\in\mathcal{N}_l(i,j)} \frac{c_{1_{\tilde{i},\tilde{j}}} - c_{1_{i,j}}}{h_l^2} \ , \quad \Delta c_2 = \sum_{l\in\{x,y\}} \sum_{\tilde{i},\tilde{j}\in\mathcal{N}_l(i,j)} \frac{c_{2_{\tilde{i},\tilde{j}}} - c_{2_{i,j}}}{h_l^2} \ , \tag{4.25}$$

where $\mathcal{N}_l(i,j)$ denotes the neighbourhood of pixel (i,j) in x- respectively y-direction and h_l the pixel size in x- respectively y-direction. For the following computations, it is assumed that $h = h_x = h_y$ to simplify the equations. Now it is possible to express the discretised version of the Euler-Lagrange equations:

$$\begin{aligned} \left(c_{1_{i,j}} f_{i,j}^t + c_{2_{i,j}} - f_{i,j}^{t+1}\right) f_{i,j}^t - \alpha \sum_{\tilde{i},\tilde{j}\in\mathcal{N}(i,j)} \frac{c_{1_{\tilde{i},\tilde{j}}} - c_{1_{i,j}}}{h^2} = 0 \ , \\ \left(c_{1_{i,j}} f_{i,j}^t + c_{2_{i,j}} - f_{i,j}^{t+1}\right) \quad - \alpha \sum_{\tilde{i},\tilde{j}\in\mathcal{N}(i,j)} \frac{c_{2_{\tilde{i},\tilde{j}}} - c_{2_{i,j}}}{h^2} = 0 \ . \end{aligned} \tag{4.26}$$

This finally yields a linear system of equations for all $N \cdot M$ matched points which is why this system of equations is of size $NM \times NM$. Since it would mean a huge computational effort to solve it, an iterative solver is used to prevent this effort. To compute the weights of the new time step, the values of the previous time step are used which means that so-called fixed point iterations are done. In this case, the Jacobi

method is used where the central pixel (i,j) of the weights is from the new time step and the neighbours are from the old time step by introducing the superscript k for the time steps. The images f^t and f^{t+1} are not affected by the iterations because they don't change during the computations. This leads to these equations where the weights of the new time step are already separated from the ones of the old time step:

$$\begin{aligned} c_{1_{i,j}}^{k+1}\left({f_{i,j}^t}^2+\alpha\frac{|\mathcal{N}(i,j)|}{h^2}\right) &= f_{i,j}^t f_{i,j}^{t+1}-c_{2_i}^k f_{i,j}^t+\alpha\sum_{\tilde{i},\tilde{j}\in\mathcal{N}(i,j)}\frac{c_{1_{\tilde{i},\tilde{j}}}^k}{h^2}\ , \\ c_{2_{i,j}}^{k+1}\left(1\quad+\alpha\frac{|\mathcal{N}(i,j)|}{h^2}\right) &= \quad f_{i,j}^{t+1}-c_{1_i}^k f_{i,j}^t+\alpha\sum_{\tilde{i},\tilde{j}\in\mathcal{N}(i,j)}\frac{c_{2_{\tilde{i},\tilde{j}}}^k}{h^2}\ , \end{aligned} \tag{4.27}$$

where $|\mathcal{N}(i,j)|$ denotes the number of neighbours of pixel (i,j). Now the final equations to compute the values of the new time step can be written as

$$\begin{aligned} c_{1_{i,j}}^{k+1} &= \frac{f_{i,j}^t f_{i,j}^{t+1}-c_{2_i}^k f_{i,j}^t+\alpha\sum_{\tilde{i},\tilde{j}\in\mathcal{N}(i,j)}\frac{c_{1_{\tilde{i},\tilde{j}}}^k}{h^2}}{{f_{i,j}^t}^2+\alpha\frac{|\mathcal{N}(i,j)|}{h^2}}\ , \\ c_{2_{i,j}}^{k+1} &= \quad\frac{f_{i,j}^{t+1}-c_{1_i}^k f_{i,j}^t+\alpha\sum_{\tilde{i},\tilde{j}\in\mathcal{N}(i,j)}\frac{c_{2_{\tilde{i},\tilde{j}}}^k}{h^2}}{1+\alpha\frac{|\mathcal{N}(i,j)|}{h^2}}\ . \end{aligned} \tag{4.28}$$

As mentioned before, the here used energy functional is not robust with respect to outliers and noise. To solve this problem, a penaliser ψ_D can be applied to the data term. The resulting energy functional then reads

$$\mathrm{E}(c_1,c_2)=\int_\Omega \psi_D\left(\left(c_1\cdot f^t+c_2-f^{t+1}\right)^2\right)+\alpha\left(|\nabla c_1|^2+|\nabla c_2|^2\right)\,\mathrm{d}x\mathrm{d}y\ . \tag{4.29}$$

By using the abbreviation $\psi'_{D_{i,j}}$ for $\psi'_D\left(\left(c_{1_{i,j}}\cdot f_{i,j}^t+c_{2_{i,j}}-f_{i,j}^{t+1}\right)^2\right)$, the final equation for the fixed point iterations for the weights can be written as

$$\begin{aligned} c_{1_{i,j}}^{k+1} &= \frac{\psi'_{D_{i,j}} f_{i,j}^t f_{i,j}^{t+1}-\psi'_{D_{i,j}} c_{2_i}^k f_{i,j}^t+\alpha\sum_{\tilde{i},\tilde{j}\in\mathcal{N}(i,j)}\frac{c_{1_{\tilde{i},\tilde{j}}}^k}{h^2}}{\psi'_{D_{i,j}}{f_{i,j}^t}^2+\alpha\frac{|\mathcal{N}(i,j)|}{h^2}}\ , \\ c_{2_{i,j}}^{k+1} &= \quad\frac{\psi'_{D_{i,j}} f_{i,j}^{t+1}-\psi'_{D_{i,j}} c_{1_i}^k f_{i,j}^t+\alpha\sum_{\tilde{i},\tilde{j}\in\mathcal{N}(i,j)}\frac{c_{2_{\tilde{i},\tilde{j}}}^k}{h^2}}{\psi'_{D_{i,j}}+\alpha\frac{|\mathcal{N}(i,j)|}{h^2}}\ , \end{aligned} \tag{4.30}$$

where the lagged nonlinearity method has to be applied by alternatively computing the weights and updating the penalisation term. There exists another possibility to make the energy functional more robust, namely by allowing discontinuities in the smoothness of the weights. This could be important for example if one object in the image has a different illumination change than the neighbouring objects. Therefore, the edges of the illumination change have to be equal to the ones of the object which is why they shouldn't be smoothed. This can be achieved by applying a penaliser ψ_S on the smoothness term, resulting in the following energy functional:

$$\mathrm{E}(c_1, c_2) = \int_\Omega \left(c_1 \cdot f^t + c_2 - f^{t+1}\right)^2 + \alpha\, \psi_S \left(|\nabla c_1|^2 + |\nabla c_2|^2\right) \mathrm{d}x\mathrm{d}y \ . \tag{4.31}$$

However, this makes the computation of the weights c_1 and c_2 more difficult. The reason for this are the Euler-Lagrange equations which now read

$$\begin{aligned} \left(c_1 f^t + c_2 - f^{t+1}\right) f^t - \alpha \operatorname{div}\left(\psi_S' \left(|\nabla c_1|^2 + |\nabla c_2|^2\right) c_1\right) = 0 \ , \\ \left(c_1 f^t + c_2 - f^{t+1}\right) \quad - \alpha \operatorname{div}\left(\psi_S' \left(|\nabla c_1|^2 + |\nabla c_2|^2\right) c_2\right) = 0 \ . \end{aligned} \tag{4.32}$$

To discretise these equations, the divergence is computed as follows:

$$\begin{aligned} \operatorname{div}\left(\psi_S' \left(|\nabla c_1|^2 + |\nabla c_2|^2\right) c_1\right) &= \sum_{l\in\{x,y\}} \sum_{\tilde{i},\tilde{j}\in\mathcal{N}_l(i)} \left(\frac{\psi_{S_j}' + \psi_{S_{i,j}}'}{2}\right) \left(\frac{c_{1_{\tilde{i},\tilde{j}}} - c_{1_{i,j}}}{h_l^2}\right) \ , \\ \operatorname{div}\left(\psi_S' \left(|\nabla c_1|^2 + |\nabla c_2|^2\right) c_2\right) &= \sum_{l\in\{x,y\}} \sum_{\tilde{i},\tilde{j}\in\mathcal{N}_l(i)} \left(\frac{\psi_{S_j}' + \psi_{S_{i,j}}'}{2}\right) \left(\frac{c_{2_{\tilde{i},\tilde{j}}} - c_{2_{i,j}}}{h_l^2}\right) \ . \end{aligned} \tag{4.33}$$

The derivative of the penalisation term is approximated in the following way:

$$\psi_{S_{i,j}}' = \psi_S' \left((c_{1_{i,j}})_x^2 + (c_{1_{i,j}})_y^2 + (c_{2_{i,j}})_x^2 + (c_{2_{i,j}})_y^2\right) \ , \tag{4.34}$$

where the derivatives of the weights are computed by using finite differences, e.g. central differences. The final equations for the computation of the weights changes to the following equations:

$$\begin{aligned} c_{1_{i,j}}^{k+1} &= \frac{f_{i,j}^t f_{i,j}^{t+1} - c_{2_i}^k f_{i,j}^t + \alpha \sum_{\tilde{i},\tilde{j}\in\mathcal{N}(i,j)} \left(\frac{\psi_{S_j}' + \psi_{S_{i,j}}'}{2}\right) \frac{c_{1_{\tilde{i},\tilde{j}}}^k}{h^2}}{{f_{i,j}^t}^2 + \alpha \sum_{\tilde{i},\tilde{j}\in\mathcal{N}(i,j)} \frac{\psi_{S_j}' + \psi_{S_{i,j}}'}{2h^2}} \ , \\ c_{2_{i,j}}^{k+1} &= \frac{f_{i,j}^{t+1} - c_{1_i}^k f_{i,j}^t + \alpha \sum_{\tilde{i},\tilde{j}\in\mathcal{N}(i,j)} \left(\frac{\psi_{S_j}' + \psi_{S_{i,j}}'}{2}\right) \frac{c_{2_{\tilde{i},\tilde{j}}}^k}{h^2}}{1 + \alpha \sum_{\tilde{i},\tilde{j}\in\mathcal{N}(i,j)} \frac{\psi_{S_j}' + \psi_{S_{i,j}}'}{2h^2}} \ . \end{aligned} \tag{4.35}$$

For an even more robust energy functional, a penaliser can be applied both on the data term and the smoothness term which can be computed by merging the equations for each method. This way it is possible to use different approaches to compute the weights c_1 and c_2 for the basis functions of illumination changes.

4.4 MOVEMENT

The movement category's metrics evaluate the type of the movement between the two images and try to detect pixels with a large displacement. Since movement doesn't consider image movement and is only dependent from the optical flow, only the ground truth flow and the computed flow are necessary for these metrics. Like in the previous case of the illumination metrics, the metrics of the movement category can be considered on their own by using only the ground truth flow or the results of the ground truth flow are compared with the results of the computed flow.

4.4.1 *Type of Movement*

A possibility to describe the movement type is the parameterized model which has been introduced in Section 2.5 and defined in equation 2.6. Two sets of basis functions have been chosen since mainly two types of movement are investigated. The first one is constant movement with the basis functions

$$\phi_1(x,y,t) = 1, \quad \eta_1(x,y,t) = 1 \ . \tag{4.36}$$

Having defined the basis functions for constant movement, the optical flow can be expressed in the following way:

$$\begin{aligned} A_1 \cdot \phi_1(x,y,t) = A_1 = u(x,y,t) \ , \\ B_1 \cdot \eta_1(x,y,t) = B_1 = v(x,y,t) \ . \end{aligned} \tag{4.37}$$

The basis functions for the second movement type, the affine model, are defined as follows:

$$\begin{aligned} \phi_1(x,y,t) = \hat{x}, \quad \phi_2(x,y,t) = \hat{y}, \quad \phi_3(x,y,t) = 1, \\ \eta_1(x,y,t) = \hat{x}, \quad \eta_2(x,y,t) = \hat{y}, \quad \eta_3(x,y,t) = 1 \ , \end{aligned} \tag{4.38}$$

where $\hat{x} = \rho(x - x_0)$ and $\hat{y} = \rho(y - y_0)$ with $\rho = 1$ and x_0 and y_0 being half of the image width respectively half of the image height. This way the optical flow can be defined as affine movement by using the basis functions in the following way:

$$\begin{aligned} A_1 \cdot \phi_1(x,y,t) + A_2 \cdot \phi_2(x,y,t) + A_3 \cdot \phi_3(x,y,t) = A_1 \cdot \hat{x} + A_2 \cdot \hat{y} + A_3 = u(x,y,t) \ , \\ B_1 \cdot \eta_1(x,y,t) + \ B_2 \cdot \eta_2(x,y,t) + \ B_3 \cdot \eta_3(x,y,t) = B_1 \cdot \hat{x} \ + B_2 \cdot \hat{y} \ + B_3 = v(x,y,t) \ . \end{aligned} \tag{4.39}$$

For the evaluation of the benchmarks either the ground truth flow or a computed flow is given. That means that it is known which point of the first frame belongs to which point in the second frame. Using this information, the equations can be used to compute the weights for the basis functions. Since the observed scene isn't necessarily static, that means that objects might move, the optical flow could be different for all the points. Therefore, the pixel notation is integrated such that the weights to describe the movement between the two frames are denoted as A_{1_i}, A_{2_i}, A_{3_i}, B_{1_i}, B_{2_i} and B_{3_i}.

4.4.1.1 *Global Constant Movement*

The category of global constant movement is defined in such a way that the optical flow between two frames is the same constant movement for every point in the image. In terms of the definition above of constant movement, the weights A_{1_i} and B_{1_i} are the same for all N points in the first image which can be matched to a point in the second image. Thus, the indices i and j can be dropped, making the constant movement global. Defining global constant movement as an equation to create an energy function from it leads to

$$\begin{aligned} A_1 = u(x,y,t) \Leftrightarrow A_1 - u(x,y,t) = 0 \ , \\ B_1 = v(x,y,t) \Leftrightarrow \ B_1 - v(x,y,t) = 0 \ . \end{aligned} \tag{4.40}$$

The flow components are abbreviated by using $u_{i,j}$ for $u(x,y,t)$ and $v_{i,j}$ for $v(x,y,t)$. The energy function can then be derived from these equations as follows:

$$\mathrm{E}(A_1, B_1) = \sum_{i=1}^{N} \sum_{j=1}^{M} \left(A_1 - u_{i,j}\right)^2 + \left(B_1 - v_{i,j}\right)^2 \ . \tag{4.41}$$

Like in the case of illumination changes, the same penalisation function can be applied to the energy function such that it reads

$$\mathrm{E}(A_1, B_1) = \sum_{i=1}^{N} \sum_{j=1}^{M} \psi\left(\left(A_1 - u_{i,j}\right)^2\right) + \psi\left(\left(B_1 - v_{i,j}\right)^2\right) \ . \tag{4.42}$$

To minimize this energy function, the derivatives with respect to the weights have to be computed:

$$\begin{aligned}\partial_{A_1} \mathrm{E}(A_1, B_1) &= 2 \cdot \sum_{i=1}^{N} \sum_{j=1}^{M} \psi' \left(\left(A_1 - u_{i,j} \right)^2 \right) \cdot \left(A_1 - u_{i,j} \right) = 0 \ , \\ \partial_{B_1} \mathrm{E}(A_1, B_1) &= 2 \cdot \sum_{i=1}^{N} \sum_{j=1}^{M} \psi' \left(\left(B_1 - v_{i,j} \right)^2 \right) \cdot \left(B_1 - v_{i,j} \right) = 0 \ .\end{aligned} \tag{4.43}$$

The derivative of the penalisation function is abbreviated again as follows:

$$\psi'_{A_{i,j}} = \psi' \left(\left(A_1 - u_{i,j} \right)^2 \right), \quad \psi'_{B_{i,j}} = \psi' \left(\left(B_1 - v_{i,j} \right)^2 \right) \ . \tag{4.44}$$

Since the weights A_1 and B_1 appear in the terms of the penaliser, they are kept fixed for the following equations such that in the end the lagged nonlinearity method can be applied. The equations can be simplified then in the following way:

$$\begin{aligned}\left(\sum_{i=1}^{N} \sum_{j=1}^{M} \psi'_{A_{i,j}} \right) \cdot A_1 &= \sum_{i=1}^{N} \sum_{j=1}^{M} \psi'_{A_{i,j}} \cdot u_{i,j} \ , \\ \left(\sum_{i=1}^{N} \sum_{j=1}^{M} \psi'_{B_{i,j}} \right) \cdot B_1 &= \sum_{i=1}^{N} \sum_{j=1}^{M} \psi'_{B_{i,j}} \cdot v_{i,j} \ .\end{aligned} \tag{4.45}$$

These equations can easily be solved for A_1 and B_1:

$$\begin{aligned}A_1 &= \frac{\sum_{i=1}^{N} \sum_{j=1}^{M} \psi'_{A_{i,j}} \cdot u_{i,j}}{\sum_{i=1}^{N} \sum_{j=1}^{M} \psi'_{A_{i,j}}} \ , \\ B_1 &= \frac{\sum_{i=1}^{N} \sum_{j=1}^{M} \psi'_{B_{i,j}} \cdot v_{i,j}}{\sum_{i=1}^{N} \sum_{j=1}^{M} \psi'_{B_{i,j}}} \ .\end{aligned} \tag{4.46}$$

After being able to solve the equations, in case of having chosen the subquadratic penaliser, the lagged nonlinearity method can be applied by iteratively computing the values for the weights and updating the penaliser terms.

4.4.1.2 *Global Affine Movement*

The global affine movement assumes, like the global constant movement, to use the same global weights for every point in the image. The indices i and j are therefore dropped such that the global weights A_1, A_2, A_3, B_1, B_2 and B_3 are used for all N matched points. However, the affine model is applied to express the optical flow between the two images of the given sequence which makes the model and the computations of the weights more complex. The equation 4.39 can then be reformulated:

$$\begin{aligned} A_1 \cdot \hat{x} + A_2 \cdot \hat{y} + A_3 = u(x,y,t) \Leftrightarrow A_1 \cdot \hat{x} + A_2 \cdot \hat{y} + A_3 - u(x,y,t) = 0 \ , \\ B_1 \cdot \hat{x} + B_2 \cdot \hat{y} + B_3 = v(x,y,t) \Leftrightarrow B_1 \cdot \hat{x} + B_2 \cdot \hat{y} + B_3 - v(x,y,t) = 0 \ . \end{aligned} \tag{4.47}$$

Using these equations and the abbreviations $u_{i,j}$ for $u(x,y,t)$ and $v_{i,j}$ for $v(x,y,t)$ for the optical flow, the energy function for the global affine movement can be expressed in the following way:

$$\begin{aligned} \mathrm{E}(A_1, A_2, A_3, B_1, B_2, B_3) = \sum_{i=1}^{N} \sum_{j=1}^{M} & \left(A_1 \cdot \hat{x} + A_2 \cdot \hat{y} + A_3 - u_{i,j}\right)^2 \\ & + \left(B_1 \cdot \hat{x} + B_2 \cdot \hat{y} + B_3 - v_{i,j}\right)^2 \ . \end{aligned} \tag{4.48}$$

To make this energy function more robust with respect to outliers and noise, the above defined penaliser is used such that the full energy function reads

$$\begin{aligned} \mathrm{E}(A_1, A_2, A_3, B_1, B_2, B_3) = \sum_{i=1}^{N} \sum_{j=1}^{M} & \psi\left(\left(A_1 \cdot \hat{x} + A_2 \cdot \hat{y} + A_3 - u_{i,j}\right)^2\right) \\ & + \psi\left(\left(B_1 \cdot \hat{x} + B_2 \cdot \hat{y} + B_3 - v_{i,j}\right)^2\right) \ . \end{aligned} \tag{4.49}$$

The derivatives of the energy function are used to minimize it since the best possible weights have to be found. To simplify the equations, the penalisation term is abbreviated as follows:

$$\psi'_{A_{i,j}} = \psi'\left(\left(A_1 \cdot \hat{x} + A_2 \cdot \hat{y} + A_3 - u_{i,j}\right)^2\right), \quad \psi'_{B_{i,j}} = \psi'\left(\left(B_1 \cdot \hat{x} + B_2 \cdot \hat{y} + B_3 - v_{i,j}\right)^2\right) \ . \tag{4.50}$$

This results in the following equations for each of the weights:

$$\partial_{A_1}\mathrm{E}(A_1, A_2, A_3, B_1, B_2, B_3) = 2 \cdot \sum_{i=1}^{N} \sum_{j=1}^{M} \psi'_{A_{i,j}} \cdot \left(A_1 \cdot \hat{x} + A_2 \cdot \hat{y} + A_3 - u_{i,j}\right) \cdot \hat{x} = 0 \ , \tag{4.51}$$

$$\partial_{A_2}\mathrm{E}(A_1, A_2, A_3, B_1, B_2, B_3) = 2 \cdot \sum_{i=1}^{N} \sum_{j=1}^{M} \psi'_{A_{i,j}} \cdot \left(A_1 \cdot \hat{x} + A_2 \cdot \hat{y} + A_3 - u_{i,j}\right) \cdot \hat{y} = 0 \ , \tag{4.52}$$

$$\partial_{A_3}\mathrm{E}(A_1, A_2, A_3, B_1, B_2, B_3) = 2 \cdot \sum_{i=1}^{N} \sum_{j=1}^{M} \psi'_{A_{i,j}} \cdot \left(A_1 \cdot \hat{x} + A_2 \cdot \hat{y} + A_3 - u_{i,j}\right) \cdot 1 = 0 \ , \tag{4.53}$$

$$\partial_{B_1} \mathrm{E}(A_1, A_2, A_3, B_1, B_2, B_3) = 2 \cdot \sum_{i=1}^{N} \sum_{j=1}^{M} \psi'_{B_{i,j}} \cdot \left(B_1 \cdot \hat{x} + B_2 \cdot \hat{y} + B_3 - v_{i,j} \right) \cdot \hat{x} = 0 \ , \tag{4.54}$$

$$\partial_{B_2} \mathrm{E}(A_1, A_2, A_3, B_1, B_2, B_3) = 2 \cdot \sum_{i=1}^{N} \sum_{j=1}^{M} \psi'_{B_{i,j}} \cdot \left(B_1 \cdot \hat{x} + B_2 \cdot \hat{y} + B_3 - v_{i,j} \right) \cdot \hat{y} = 0 \ , \tag{4.55}$$

$$\partial_{B_3} \mathrm{E}(A_1, A_2, A_3, B_1, B_2, B_3) = 2 \cdot \sum_{i=1}^{N} \sum_{j=1}^{M} \psi'_{B_{i,j}} \cdot \left(B_1 \cdot \hat{x} + B_2 \cdot \hat{y} + B_3 - v_{i,j} \right) \cdot 1 = 0 \ . \tag{4.56}$$

Using the abbreviations of the penalisation terms as fixed values, it is possible to reformulated these equations as two linear systems of equations since the weights A_i are independent from the weights B_i. This way the lagged nonlinearity method can be applied in the end to solve these nonlinear equations. The linear systems can be derived in the following way:

$$\begin{aligned}
&\begin{pmatrix} \sum_{i=1}^{N} \sum_{j=1}^{M} \psi'_{A_{i,j}} \cdot \hat{x}^2 & \sum_{i=1}^{N} \sum_{j=1}^{M} \psi'_{A_{i,j}} \cdot \hat{x} \cdot \hat{y} & \sum_{i=1}^{N} \sum_{j=1}^{M} \psi'_{A_{i,j}} \cdot \hat{x} \\ \sum_{i=1}^{N} \sum_{j=1}^{M} \psi'_{A_{i,j}} \cdot \hat{x} \cdot \hat{y} & \sum_{i=1}^{N} \sum_{j=1}^{M} \psi'_{A_{i,j}} \cdot \hat{y}^2 & \sum_{i=1}^{N} \sum_{j=1}^{M} \psi'_{A_{i,j}} \cdot \hat{y} \\ \sum_{i=1}^{N} \sum_{j=1}^{M} \psi'_{A_{i,j}} \cdot \hat{x} & \sum_{i=1}^{N} \sum_{j=1}^{M} \psi'_{A_{i,j}} \cdot \hat{y} & \sum_{i=1}^{N} \sum_{j=1}^{M} \psi'_{A_{i,j}} \end{pmatrix} \cdot \begin{pmatrix} A_1 \\ A_2 \\ A_3 \end{pmatrix} \\
&\qquad = \begin{pmatrix} \sum_{i=1}^{N} \sum_{j=1}^{M} \psi'_{A_{i,j}} \cdot \hat{x} \cdot u_{i,j} \\ \sum_{i=1}^{N} \sum_{j=1}^{M} \psi'_{A_{i,j}} \cdot \hat{y} \cdot u_{i,j} \\ \sum_{i=1}^{N} \sum_{j=1}^{M} \psi'_{A_{i,j}} \cdot u_{i,j} \end{pmatrix} \ , \\
&\begin{pmatrix} \sum_{i=1}^{N} \sum_{j=1}^{M} \psi'_{B_{i,j}} \cdot \hat{x}^2 & \sum_{i=1}^{N} \sum_{j=1}^{M} \psi'_{B_{i,j}} \cdot \hat{x} \cdot \hat{y} & \sum_{i=1}^{N} \sum_{j=1}^{M} \psi'_{B_{i,j}} \cdot \hat{x} \\ \sum_{i=1}^{N} \sum_{j=1}^{M} \psi'_{B_{i,j}} \cdot \hat{x} \cdot \hat{y} & \sum_{i=1}^{N} \sum_{j=1}^{M} \psi'_{B_{i,j}} \cdot \hat{y}^2 & \sum_{i=1}^{N} \sum_{j=1}^{M} \psi'_{B_{i,j}} \cdot \hat{y} \\ \sum_{i=1}^{N} \sum_{j=1}^{M} \psi'_{B_{i,j}} \cdot \hat{x} & \sum_{i=1}^{N} \sum_{j=1}^{M} \psi'_{B_{i,j}} \cdot \hat{y} & \sum_{i=1}^{N} \sum_{j=1}^{M} \psi'_{B_{i,j}} \end{pmatrix} \cdot \begin{pmatrix} B_1 \\ B_2 \\ B_3 \end{pmatrix} \\
&\qquad = \begin{pmatrix} \sum_{i=1}^{N} \sum_{j=1}^{M} \psi'_{B_{i,j}} \cdot \hat{x} \cdot u_{i,j} \\ \sum_{i=1}^{N} \sum_{j=1}^{M} \psi'_{B_{i,j}} \cdot \hat{y} \cdot u_{i,j} \\ \sum_{i=1}^{N} \sum_{j=1}^{M} \psi'_{B_{i,j}} \cdot u_{i,j} \end{pmatrix} \ .
\end{aligned} \tag{4.57}$$

These linear systems of equations can be solved for the weights by applying Cramer's rule to them. Since the resulting equations are quite complex, some abbreviations are used to simplify them. This results in the following equations for each of the weights:

$$\begin{aligned}
A_1 &= \frac{\det(M_{A_1})}{\det(M_A)} \ , \quad A_2 = \frac{\det(M_{A_2})}{\det(M_A)} \ , \quad A_3 = \frac{\det(M_{A_3})}{\det(M_A)} \ , \\
B_1 &= \frac{\det(M_{B_1})}{\det(M_B)} \ , \quad B_2 = \frac{\det(M_{B_2})}{\det(M_B)} \ , \quad B_3 = \frac{\det(M_{B_3})}{\det(M_B)} \ .
\end{aligned} \tag{4.58}$$

The abbreviations are defined as follows for the weights A_i and B_i:

$$\begin{aligned}
\det(M_{A_1}) = &\left(\sum_{i=1}^{N}\sum_{j=1}^{M}\psi'_{A_{i,j}}\hat{x}u_{i,j}\right)\left(\sum_{i=1}^{N}\sum_{j=1}^{M}\psi'_{A_{i,j}}\hat{y}^2\right)\left(\sum_{i=1}^{N}\sum_{j=1}^{M}\psi'_{A_{i,j}}\right)\\
&+\left(\sum_{i=1}^{N}\sum_{j=1}^{M}\psi'_{A_{i,j}}u_{i,j}\right)\left(\sum_{i=1}^{N}\sum_{j=1}^{M}\psi'_{A_{i,j}}\hat{y}\right)\left(\sum_{i=1}^{N}\sum_{j=1}^{M}\psi'_{A_{i,j}}\hat{x}\hat{y}\right)\\
&+\left(\sum_{i=1}^{N}\sum_{j=1}^{M}\psi'_{A_{i,j}}\hat{x}\right)\left(\sum_{i=1}^{N}\sum_{j=1}^{M}\psi'_{A_{i,j}}\hat{y}\right)\left(\sum_{i=1}^{N}\sum_{j=1}^{M}\psi'_{A_{i,j}}\hat{y}u_{i,j}\right)\\
&-\left(\sum_{i=1}^{N}\sum_{j=1}^{M}\psi'_{A_{i,j}}u_{i,j}\right)\left(\sum_{i=1}^{N}\sum_{j=1}^{M}\psi'_{A_{i,j}}\hat{x}\right)\left(\sum_{i=1}^{N}\sum_{j=1}^{M}\psi'_{A_{i,j}}\hat{y}^2\right)\\
&-\left(\sum_{i=1}^{N}\sum_{j=1}^{M}\psi'_{A_{i,j}}\hat{y}\right)^2\left(\sum_{i=1}^{N}\sum_{j=1}^{M}\psi'_{A_{i,j}}\hat{x}u_{i,j}\right)\\
&-\left(\sum_{i=1}^{N}\sum_{j=1}^{M}\psi'_{A_{i,j}}\hat{x}\hat{y}\right)\left(\sum_{i=1}^{N}\sum_{j=1}^{M}\psi'_{A_{i,j}}\hat{y}u_{i,j}\right)\left(\sum_{i=1}^{N}\sum_{j=1}^{M}\psi'_{A_{i,j}}\right)\quad,
\end{aligned}\tag{4.59}$$

$$\begin{aligned}
\det(M_{A_2}) = &\left(\sum_{i=1}^{N}\sum_{j=1}^{M}\psi'_{A_{i,j}}\hat{x}^2\right)\left(\sum_{i=1}^{N}\sum_{j=1}^{M}\psi'_{A_{i,j}}\hat{y}u_{i,j}\right)\left(\sum_{i=1}^{N}\sum_{j=1}^{M}\psi'_{A_{i,j}}\right)\\
&+\left(\sum_{i=1}^{N}\sum_{j=1}^{M}\psi'_{A_{i,j}}\hat{x}\right)\left(\sum_{i=1}^{N}\sum_{j=1}^{M}\psi'_{A_{i,j}}\hat{y}\right)\left(\sum_{i=1}^{N}\sum_{j=1}^{M}\psi'_{A_{i,j}}\hat{x}u_{i,j}\right)\\
&+\left(\sum_{i=1}^{N}\sum_{j=1}^{M}\psi'_{A_{i,j}}\hat{x}\right)\left(\sum_{i=1}^{N}\sum_{j=1}^{M}\psi'_{A_{i,j}}\hat{x}\hat{y}\right)\left(\sum_{i=1}^{N}\sum_{j=1}^{M}\psi'_{A_{i,j}}u_{i,j}\right)\\
&-\left(\sum_{i=1}^{N}\sum_{j=1}^{M}\psi'_{A_{i,j}}\hat{x}\right)^2\left(\sum_{i=1}^{N}\sum_{j=1}^{M}\psi'_{A_{i,j}}\hat{y}u_{i,j}\right)\\
&-\left(\sum_{i=1}^{N}\sum_{j=1}^{M}\psi'_{A_{i,j}}\hat{x}^2\right)\left(\sum_{i=1}^{N}\sum_{j=1}^{M}\psi'_{A_{i,j}}\hat{y}\right)\left(\sum_{i=1}^{N}\sum_{j=1}^{M}\psi'_{A_{i,j}}u_{i,j}\right)\\
&-\left(\sum_{i=1}^{N}\sum_{j=1}^{M}\psi'_{A_{i,j}}\hat{x}\hat{y}\right)\left(\sum_{i=1}^{N}\sum_{j=1}^{M}\psi'_{A_{i,j}}\hat{x}u_{i,j}\right)\left(\sum_{i=1}^{N}\sum_{j=1}^{M}\psi'_{A_{i,j}}\right)\quad,
\end{aligned}\tag{4.60}$$

$$
\begin{aligned}
\det(M_{A_3}) = &\left(\sum_{i=1}^{N}\sum_{j=1}^{M}\psi'_{A_{i,j}}\hat{x}^2\right)\left(\sum_{i=1}^{N}\sum_{j=1}^{M}\psi'_{A_{i,j}}\hat{y}^2\right)\left(\sum_{i=1}^{N}\sum_{j=1}^{M}\psi'_{A_{i,j}}u_{i,j}\right)\\
&+\left(\sum_{i=1}^{N}\sum_{j=1}^{M}\psi'_{A_{i,j}}\hat{x}\right)\left(\sum_{i=1}^{N}\sum_{j=1}^{M}\psi'_{A_{i,j}}\hat{x}\hat{y}\right)\left(\sum_{i=1}^{N}\sum_{j=1}^{M}\psi'_{A_{i,j}}\hat{y}u_{i,j}\right)\\
&+\left(\sum_{i=1}^{N}\sum_{j=1}^{M}\psi'_{A_{i,j}}\hat{y}\right)\left(\sum_{i=1}^{N}\sum_{j=1}^{M}\psi'_{A_{i,j}}\hat{x}\hat{y}\right)\left(\sum_{i=1}^{N}\sum_{j=1}^{M}\psi'_{A_{i,j}}\hat{x}u_{i,j}\right)\\
&-\left(\sum_{i=1}^{N}\sum_{j=1}^{M}\psi'_{A_{i,j}}\hat{x}\right)\left(\sum_{i=1}^{N}\sum_{j=1}^{M}\psi'_{A_{i,j}}\hat{y}^2\right)\left(\sum_{i=1}^{N}\sum_{j=1}^{M}\psi'_{A_{i,j}}\hat{x}u_{i,j}\right)\\
&-\left(\sum_{i=1}^{N}\sum_{j=1}^{M}\psi'_{A_{i,j}}\hat{y}\right)\left(\sum_{i=1}^{N}\sum_{j=1}^{M}\psi'_{A_{i,j}}\hat{x}^2\right)\left(\sum_{i=1}^{N}\sum_{j=1}^{M}\psi'_{A_{i,j}}\hat{y}u_{i,j}\right)\\
&-\left(\sum_{i=1}^{N}\sum_{j=1}^{M}\psi'_{A_{i,j}}\hat{x}\hat{y}\right)^2\left(\sum_{i=1}^{N}\sum_{j=1}^{M}\psi'_{A_{i,j}}u_{i,j}\right) \quad ,
\end{aligned}
\tag{4.61}
$$

$$
\begin{aligned}
\det(M_A) = &\left(\sum_{i=1}^{N}\sum_{j=1}^{M}\psi'_{A_{i,j}}\hat{x}^2\right)\left(\sum_{i=1}^{N}\sum_{j=1}^{M}\psi'_{A_{i,j}}\hat{y}^2\right)\left(\sum_{i=1}^{N}\sum_{j=1}^{M}\psi'_{A_{i,j}}\right)\\
&+2\left(\sum_{i=1}^{N}\sum_{j=1}^{M}\psi'_{A_{i,j}}\hat{x}\right)\left(\sum_{i=1}^{N}\sum_{j=1}^{M}\psi'_{A_{i,j}}\hat{y}\right)\left(\sum_{i=1}^{N}\sum_{j=1}^{M}\psi'_{A_{i,j}}\hat{x}\hat{y}\right)\\
&-\left(\sum_{i=1}^{N}\sum_{j=1}^{M}\psi'_{A_{i,j}}\hat{x}\right)^2\left(\sum_{i=1}^{N}\sum_{j=1}^{M}\psi'_{A_{i,j}}\hat{y}^2\right)\\
&-\left(\sum_{i=1}^{N}\sum_{j=1}^{M}\psi'_{A_{i,j}}\hat{y}\right)^2\left(\sum_{i=1}^{N}\sum_{j=1}^{M}\psi'_{A_{i,j}}\hat{x}^2\right)\\
&-\left(\sum_{i=1}^{N}\sum_{j=1}^{M}\psi'_{A_{i,j}}\hat{x}\hat{y}\right)^2\left(\sum_{i=1}^{N}\sum_{j=1}^{M}\psi'_{A_{i,j}}\right) \quad ,
\end{aligned}
\tag{4.62}
$$

$$
\begin{aligned}
\det(M_{B_1}) = &\left(\sum_{i=1}^{N}\sum_{j=1}^{M}\psi'_{B_{i,j}}\hat{x}v_{i,j}\right)\left(\sum_{i=1}^{N}\sum_{j=1}^{M}\psi'_{B_{i,j}}\hat{y}^2\right)\left(\sum_{i=1}^{N}\sum_{j=1}^{M}\psi'_{B_{i,j}}\right)\\
&+\left(\sum_{i=1}^{N}\sum_{j=1}^{M}\psi'_{B_{i,j}}v_{i,j}\right)\left(\sum_{i=1}^{N}\sum_{j=1}^{M}\psi'_{B_{i,j}}\hat{y}\right)\left(\sum_{i=1}^{N}\sum_{j=1}^{M}\psi'_{B_{i,j}}\hat{x}\hat{y}\right)\\
&+\left(\sum_{i=1}^{N}\sum_{j=1}^{M}\psi'_{B_{i,j}}\hat{x}\right)\left(\sum_{i=1}^{N}\sum_{j=1}^{M}\psi'_{B_{i,j}}\hat{y}\right)\left(\sum_{i=1}^{N}\sum_{j=1}^{M}\psi'_{B_{i,j}}\hat{y}v_{i,j}\right)\\
&-\left(\sum_{i=1}^{N}\sum_{j=1}^{M}\psi'_{B_{i,j}}v_{i,j}\right)\left(\sum_{i=1}^{N}\sum_{j=1}^{M}\psi'_{B_{i,j}}\hat{x}\right)\left(\sum_{i=1}^{N}\sum_{j=1}^{M}\psi'_{B_{i,j}}\hat{y}^2\right)\\
&-\left(\sum_{i=1}^{N}\sum_{j=1}^{M}\psi'_{B_{i,j}}\hat{y}\right)^2\left(\sum_{i=1}^{N}\sum_{j=1}^{M}\psi'_{B_{i,j}}\hat{x}v_{i,j}\right)\\
&-\left(\sum_{i=1}^{N}\sum_{j=1}^{M}\psi'_{B_{i,j}}\hat{x}\hat{y}\right)\left(\sum_{i=1}^{N}\sum_{j=1}^{M}\psi'_{B_{i,j}}\hat{y}v_{i,j}\right)\left(\sum_{i=1}^{N}\sum_{j=1}^{M}\psi'_{B_{i,j}}\right)\quad,
\end{aligned}
\tag{4.63}
$$

$$
\begin{aligned}
\det(M_{B_2}) = &\left(\sum_{i=1}^{N}\sum_{j=1}^{M}\psi'_{B_{i,j}}\hat{x}^2\right)\left(\sum_{i=1}^{N}\sum_{j=1}^{M}\psi'_{B_{i,j}}\hat{y}v_{i,j}\right)\left(\sum_{i=1}^{N}\sum_{j=1}^{M}\psi'_{B_{i,j}}\right)\\
&+\left(\sum_{i=1}^{N}\sum_{j=1}^{M}\psi'_{B_{i,j}}\hat{x}\right)\left(\sum_{i=1}^{N}\sum_{j=1}^{M}\psi'_{B_{i,j}}\hat{y}\right)\left(\sum_{i=1}^{N}\sum_{j=1}^{M}\psi'_{B_{i,j}}\hat{x}v_{i,j}\right)\\
&+\left(\sum_{i=1}^{N}\sum_{j=1}^{M}\psi'_{B_{i,j}}\hat{x}\right)\left(\sum_{i=1}^{N}\sum_{j=1}^{M}\psi'_{B_{i,j}}\hat{x}\hat{y}\right)\left(\sum_{i=1}^{N}\sum_{j=1}^{M}\psi'_{B_{i,j}}v_{i,j}\right)\\
&-\left(\sum_{i=1}^{N}\sum_{j=1}^{M}\psi'_{B_{i,j}}\hat{x}\right)^2\left(\sum_{i=1}^{N}\sum_{j=1}^{M}\psi'_{B_{i,j}}\hat{y}v_{i,j}\right)\\
&-\left(\sum_{i=1}^{N}\sum_{j=1}^{M}\psi'_{B_{i,j}}\hat{x}^2\right)\left(\sum_{i=1}^{N}\sum_{j=1}^{M}\psi'_{B_{i,j}}\hat{y}\right)\left(\sum_{i=1}^{N}\sum_{j=1}^{M}\psi'_{B_{i,j}}v_{i,j}\right)\\
&-\left(\sum_{i=1}^{N}\sum_{j=1}^{M}\psi'_{B_{i,j}}\hat{x}\hat{y}\right)\left(\sum_{i=1}^{N}\sum_{j=1}^{M}\psi'_{B_{i,j}}\hat{x}v_{i,j}\right)\left(\sum_{i=1}^{N}\sum_{j=1}^{M}\psi'_{B_{i,j}}\right)\quad,
\end{aligned}
\tag{4.64}
$$

$$
\begin{aligned}
\det(M_{B_3}) = &\left(\sum_{i=1}^{N}\sum_{j=1}^{M}\psi'_{B_{i,j}}\hat{x}^2\right)\left(\sum_{i=1}^{N}\sum_{j=1}^{M}\psi'_{B_{i,j}}\hat{y}^2\right)\left(\sum_{i=1}^{N}\sum_{j=1}^{M}\psi'_{B_{i,j}}v_{i,j}\right)\\
&+\left(\sum_{i=1}^{N}\sum_{j=1}^{M}\psi'_{B_{i,j}}\hat{x}\right)\left(\sum_{i=1}^{N}\sum_{j=1}^{M}\psi'_{B_{i,j}}\hat{x}\hat{y}\right)\left(\sum_{i=1}^{N}\sum_{j=1}^{M}\psi'_{B_{i,j}}\hat{y}v_{i,j}\right)\\
&+\left(\sum_{i=1}^{N}\sum_{j=1}^{M}\psi'_{B_{i,j}}\hat{y}\right)\left(\sum_{i=1}^{N}\sum_{j=1}^{M}\psi'_{B_{i,j}}\hat{x}\hat{y}\right)\left(\sum_{i=1}^{N}\sum_{j=1}^{M}\psi'_{B_{i,j}}\hat{x}v_{i,j}\right)\\
&-\left(\sum_{i=1}^{N}\sum_{j=1}^{M}\psi'_{B_{i,j}}\hat{x}\right)\left(\sum_{i=1}^{N}\sum_{j=1}^{M}\psi'_{B_{i,j}}\hat{y}^2\right)\left(\sum_{i=1}^{N}\sum_{j=1}^{M}\psi'_{B_{i,j}}\hat{x}v_{i,j}\right)\\
&-\left(\sum_{i=1}^{N}\sum_{j=1}^{M}\psi'_{B_{i,j}}\hat{y}\right)\left(\sum_{i=1}^{N}\sum_{j=1}^{M}\psi'_{B_{i,j}}\hat{x}^2\right)\left(\sum_{i=1}^{N}\sum_{j=1}^{M}\psi'_{B_{i,j}}\hat{y}v_{i,j}\right)\\
&-\left(\sum_{i=1}^{N}\sum_{j=1}^{M}\psi'_{B_{i,j}}\hat{x}\hat{y}\right)^2\left(\sum_{i=1}^{N}\sum_{j=1}^{M}\psi'_{B_{i,j}}v_{i,j}\right) \quad ,
\end{aligned}
\tag{4.65}
$$

$$
\begin{aligned}
\det(M_B) = &\left(\sum_{i=1}^{N}\sum_{j=1}^{M}\psi'_{B_{i,j}}\hat{x}^2\right)\left(\sum_{i=1}^{N}\sum_{j=1}^{M}\psi'_{B_{i,j}}\hat{y}^2\right)\left(\sum_{i=1}^{N}\sum_{j=1}^{M}\psi'_{B_{i,j}}\right)\\
&+2\left(\sum_{i=1}^{N}\sum_{j=1}^{M}\psi'_{B_{i,j}}\hat{x}\right)\left(\sum_{i=1}^{N}\sum_{j=1}^{M}\psi'_{B_{i,j}}\hat{y}\right)\left(\sum_{i=1}^{N}\sum_{j=1}^{M}\psi'_{B_{i,j}}\hat{x}\hat{y}\right)\\
&-\left(\sum_{i=1}^{N}\sum_{j=1}^{M}\psi'_{B_{i,j}}\hat{x}\right)^2\left(\sum_{i=1}^{N}\sum_{j=1}^{M}\psi'_{B_{i,j}}\hat{y}^2\right)\\
&-\left(\sum_{i=1}^{N}\sum_{j=1}^{M}\psi'_{B_{i,j}}\hat{y}\right)^2\left(\sum_{i=1}^{N}\sum_{j=1}^{M}\psi'_{B_{i,j}}\hat{x}^2\right)\\
&-\left(\sum_{i=1}^{N}\sum_{j=1}^{M}\psi'_{B_{i,j}}\hat{x}\hat{y}\right)^2\left(\sum_{i=1}^{N}\sum_{j=1}^{M}\psi'_{B_{i,j}}\right) \quad .
\end{aligned}
\tag{4.66}
$$

After having defined the way to compute solutions for the weights, it depends on whether the quadratic or the subquadratic penaliser has been chosen. If the subquadratic penalisation function has been used, the lagged nonlinearity method can be applied now like in the previous cases with the global illumination changes and the global constant movement.

4.4.1.3 *Local Constant Movement*

In contrast to the global constant movement, the local constant movement assumes that the weights are continuous and might be different for every point. This means

that the assumption is similar to the one for local illumination changes. But instead of computing the weights to express the optical flow, the optical flow should be reconstructed by including a specific smoothness assumption. The resulting new optical flow $(\tilde{u}, \tilde{v})$ can then be evaluated to characterise the type of movement. Since most of the following computations are similar to those for the local illumination changes, they are presented more briefly. Penalisation functions for the data term or the smoothness term are discussed later. First of all, the energy functional has to be defined by choosing the following data term $\mathrm{D}(\tilde{u}, \tilde{v})$:

$$\mathrm{D}(\tilde{u}, \tilde{v}) = (\tilde{u}(x,y,t) - u(x,y,t))^2 + (\tilde{v}(x,y,t) - v(x,y,t))^2 \quad . \tag{4.67}$$

Since in this part the constant movement is considered, the smoothness term penalises deviations from the gradient of the new optical flow such that it can be written as follows

$$\mathrm{S}(\tilde{u}, \tilde{v}) = \left(|\nabla \tilde{u}(x,y,t)|^2 + |\nabla \tilde{v}(x,y,t)|^2 \right) \quad . \tag{4.68}$$

The abbreviations $u = u(x,y,t)$, $v = v(x,y,t)$, $\tilde{u} = \tilde{u}(x,y,t)$ and $\tilde{v} = \tilde{v}(x,y,t)$ are used. From these two terms, the continuous energy functional can be derived:

$$\begin{aligned} \mathrm{E}(\tilde{u}, \tilde{v}) &= \int_\Omega \quad \mathrm{D}(\tilde{u}, \tilde{v}) \quad + \alpha \quad \mathrm{S}(\tilde{u}, \tilde{v}) \quad \mathrm{d}x\mathrm{d}y \\ &= \int_\Omega \gamma \left((\tilde{u} - u)^2 + (\tilde{v} - v)^2 \right) + \alpha \left(|\nabla \tilde{u}|^2 + |\nabla \tilde{v}|^2 \right) \mathrm{d}x\mathrm{d}y \quad , \end{aligned} \tag{4.69}$$

where γ is an indicator function again to handle points where no optical flow is defined. To ease the following equations, the indicator function is dropped. Having determined the energy functional, it is possible to derive the Euler-Lagrange equations from it which are already simplified:

$$\begin{aligned} (\tilde{u} - u) - \alpha \Delta \tilde{u} &= 0 \quad , \\ (\tilde{v} - v) - \alpha \Delta \tilde{v} &= 0 \quad . \end{aligned} \tag{4.70}$$

The discretisation is done similarly like for the illumination changes with the same approximation for the Laplacian and by using $u_{i,j}$, $v_{i,j}$, $\tilde{u}_{i,j}$ and $\tilde{v}_{i,j}$ for the discrete optical flows and $\mathcal{N}(i,j)$ for the whole neighbourhood of a pixel (i,j):

$$\begin{aligned} \left(\tilde{u}_{i,j} - u_{i,j} \right) - \alpha \sum_{\tilde{i},\tilde{j} \in \mathcal{N}(i,j)} \frac{\tilde{u}_{\tilde{i},\tilde{j}} - \tilde{u}_{i,j}}{h^2} &= 0 \quad , \\ \left(\tilde{v}_{i,j} - v_{i,j} \right) - \alpha \sum_{\tilde{i},\tilde{j} \in \mathcal{N}(i,j)} \frac{\tilde{v}_{\tilde{i},\tilde{j}} - \tilde{v}_{i,j}}{h^2} &= 0 \quad . \end{aligned} \tag{4.71}$$

To compute the values of the new optical flow, fixed point iterations are used instead of solving the whole linear system of equations. Therefore, the considered central pixel is separated from the rest of the equations and gets the time step superscript $k+1$ while the rest of the equation is from the older time step k. Since the given optical flow components u and v don't change during the computations, they don't need the superscript k. This yields

$$\begin{aligned} \tilde{u}_{i,j}^{k+1}\left(1+\alpha\frac{|\mathcal{N}(i,j)|}{h^2}\right) &= u_{i,j}+\alpha\sum_{\tilde{i},\tilde{j}\in\mathcal{N}(i,j)}\frac{\tilde{u}_{\tilde{i},\tilde{j}}^{k}}{h^2}\ , \\ \tilde{v}_{i,j}^{k+1}\left(1+\alpha\frac{|\mathcal{N}(i,j)|}{h^2}\right) &= v_{i,j}+\alpha\sum_{\tilde{i},\tilde{j}\in\mathcal{N}(i,j)}\frac{\tilde{v}_{\tilde{i},\tilde{j}}^{k}}{h^2}\ , \end{aligned} \tag{4.72}$$

where $|\mathcal{N}(i,j)|$ denotes the number of neighbours of the pixel (i,j). Then the final equation reads

$$\begin{aligned} \tilde{u}_{i,j}^{k+1} &= \frac{u_{i,j}+\alpha\sum_{\tilde{i},\tilde{j}\in\mathcal{N}(i,j)}\frac{\tilde{u}_{\tilde{i},\tilde{j}}^{k}}{h^2}}{1+\alpha\frac{|\mathcal{N}(i,j)|}{h^2}}\ , \\ \tilde{v}_{i,j}^{k+1} &= \frac{v_{i,j}+\alpha\sum_{\tilde{i},\tilde{j}\in\mathcal{N}(i,j)}\frac{\tilde{v}_{\tilde{i},\tilde{j}}^{k}}{h^2}}{1+\alpha\frac{|\mathcal{N}(i,j)|}{h^2}}\ . \end{aligned} \tag{4.73}$$

Since these two equations have no variables in common, they can be computed independently from each other. However, these equations have some limitations since first of all an optical flow is needed to compute a new optical flow from it. Secondly, the optical flow might not be defined for all points (i,j) which can be resolved e.g. by using a flag variable which is equal to 1 if the flow is defined and 0 if not. This flag variable can be then multiplied to $u_{i,j}$ respectively $v_{i,j}$. In these cases where the optical flow is not defined, the smoothness term takes over and fills in information by using the neighbours of the considered pixel. As mentioned before, penalisers can be applied to the data term and the smoothness term. Both possibilites change the final equations for the fixed point iterations like they did for the local illumination changes. Nevertheless, they are summarised in the following. The penaliser for the data term ψ_D is applied to each optical flow component separately:

$$\mathrm{E}(\tilde{u},\tilde{v}) = \int_\Omega \psi_D\left((\tilde{u}-u)^2\right)+\psi_D\left((\tilde{v}-v)^2\right)+\alpha\left(|\nabla\tilde{u}|^2+|\nabla\tilde{v}|^2\right)\mathrm{d}x\mathrm{d}y\ . \tag{4.74}$$

Using the abbreviations $\psi'^{u}_{D_{i,j}}$ for $\psi'_D\left(\left(\tilde{u}_{i,j}-u_{i,j}\right)^2\right)$ and $\psi'^{v}_{D_{i,j}}$ for $\psi'_D\left(\left(\tilde{v}_{i,j}-v_{i,j}\right)^2\right)$, the final equations of the fixed point iterations look like this:

$$\begin{aligned}\tilde{u}_{i,j}^{k+1} &= \frac{\psi'^{u}_{D_{i,j}} u_{i,j} + \alpha \sum_{\tilde{i},\tilde{j}\in\mathcal{N}(i,j)} \frac{\tilde{u}^{k}_{\tilde{i},\tilde{j}}}{h^2}}{\psi'^{u}_{D_{i,j}} + \alpha \frac{|\mathcal{N}(i,j)|}{h^2}} \ , \\ \tilde{v}_{i,j}^{k+1} &= \frac{\psi'^{v}_{D_{i,j}} v_{i,j} + \alpha \sum_{\tilde{i},\tilde{j}\in\mathcal{N}(i,j)} \frac{\tilde{v}^{k}_{\tilde{i},\tilde{j}}}{h^2}}{\psi'^{v}_{D_{i,j}} + \alpha \frac{|\mathcal{N}(i,j)|}{h^2}} \ .\end{aligned} \tag{4.75}$$

The penaliser of the smoothness term ψ_S is applied to each optical flow component separately as well, resulting in the following energy functional:

$$\mathrm{E}(\tilde{u},\tilde{v}) = \int_\Omega (\tilde{u}-u)^2 + (\tilde{v}-v)^2 + \alpha \left(\psi_s\left(|\nabla\tilde{u}|^2\right) + \psi_S\left(|\nabla\tilde{v}|^2\right)\right) \mathrm{d}x\mathrm{d}y \ . \tag{4.76}$$

The Euler-Lagrange equations which are computed from this energy functional are similar to the ones from the energy functional for local illumination changes with robust smoothness term. That means that the divergence has to be approximated in the same way. A more detailed derivation of the final equations to compute the weights can be found above. Using the abbreviations $\psi'^{u}_{S_{i,j}}$ and $\psi'^{v}_{S_{i,j}}$ for the discretised versions of the penalisation term, the equations for the fixed point iterations then read

$$\begin{aligned}\tilde{u}_{i,j}^{k+1} &= \frac{u_{i,j} + \alpha \sum_{\tilde{i},\tilde{j}\in\mathcal{N}(i,j)} \left(\frac{\psi'^{u}_{S_{\tilde{i},\tilde{j}}} + \psi'^{u}_{S_{i,j}}}{2}\right) \frac{\tilde{u}^{k}_{\tilde{i},\tilde{j}}}{h^2}}{1 + \alpha \sum_{\tilde{i},\tilde{j}\in\mathcal{N}(i,j)} \frac{\psi'^{u}_{S_{\tilde{i},\tilde{j}}} + \psi'^{u}_{S_{i,j}}}{2h^2}} \ , \\ \tilde{v}_{i,j}^{k+1} &= \frac{v_{i,j} + \alpha \sum_{\tilde{i},\tilde{j}\in\mathcal{N}(i,j)} \left(\frac{\psi'^{v}_{S_{\tilde{i},\tilde{j}}} + \psi'^{v}_{S_{i,j}}}{2}\right) \frac{\tilde{v}^{k}_{\tilde{i},\tilde{j}}}{h^2}}{1 + \alpha \sum_{\tilde{i},\tilde{j}\in\mathcal{N}(i,j)} \frac{\psi'^{v}_{S_{\tilde{i},\tilde{j}}} + \psi'^{v}_{S_{i,j}}}{2h^2}} \ .\end{aligned} \tag{4.77}$$

After having computed the new optical flow components $\tilde{u}$ and $\tilde{v}$, they can be analysed now. This can be done by applying the smoothness term on the image and evaluating the result. This basically means that the absolute values of the gradients $|\nabla\tilde{u}|^2$ and $|\nabla\tilde{v}|^2$ have to be computed. Four possible cases can be derived from this analysis:

- $|\nabla\tilde{u}|^2 \leq T_u$ and $|\nabla\tilde{v}|^2 \leq T_v$ which means that both gradients are small and therefore, the movement is constant both in x- and y-direction.
- $|\nabla\tilde{u}|^2 \leq T_u$ but $|\nabla\tilde{v}|^2 > T_v$ which means that only $\nabla\tilde{u}$ is small and therefore, the movement is constant only in x-direction.
- $|\nabla\tilde{v}|^2 \leq T_v$ but $|\nabla\tilde{u}|^2 > T_u$ which means that only $\nabla\tilde{v}$ is small and therefore, the movement is constant only in y-direction.

- $|\nabla\tilde{u}|^2 > T_u$ and $|\nabla\tilde{v}|^2 > T_v$ which means that neither of them is small and the movement isn't constant in any direction.

This analysis of the type of movement should be done especially for the points were the optical flow (u, v) was defined.

4.4.1.4 *Local Affine Movement*

Just like for the local constant movement, it is assumed that the weights for the local affine movement are continuous and might be different for every point. Furthermore, a new optical flow $(\tilde{u}, \tilde{v})$ is computed which is then evaluated to characterise the type of movement. Since some of the following computations are similar to those for the local constant movement, they are presented more briefly. Penalisation functions for the data term or the smoothness term are discussed later. Due to the fact that the new optical flow is computed from the given one, the same data term from equation 4.67 is used. However, this time affine movement should be detected as well. This means that a different smoothness term has to be used, namely by using the Hessian instead of the gradient and penalising deviations from its Frobenius norm. This yields the following smoothness term:

$$\mathrm{S}(\tilde{u}, \tilde{v}) = \left(\|\mathrm{H}(\tilde{u}(x, y, t))\|_F^2 + \|\mathrm{H}(\tilde{v}(x, y, t))\|_F^2 \right) \quad . \tag{4.78}$$

The abbreviations $u = u(x, y, t)$, $v = v(x, y, t)$, $\tilde{u} = \tilde{u}(x, y, t)$ and $\tilde{v} = \tilde{v}(x, y, t)$ are used in the following computations. The Frobenius norm of the Hessian is defined as follows:

$$\|\mathrm{H}(\tilde{u})\|_F^2 = \tilde{u}_{xx}^2 + \tilde{u}_{xy}^2 + \tilde{u}_{yx}^2 + \tilde{u}_{yy}^2 \quad , \tag{4.79}$$

where it is assumed that $\tilde{u}_{xy} = \tilde{u}_{yx}$. The same holds for the other flow component $\tilde{v}$. From the data term and the smoothness term, the continuous energy functional can be derived:

$$\begin{aligned} \mathrm{E}(\tilde{u}, \tilde{v}) &= \int_\Omega \quad \mathrm{D}(\tilde{u}, \tilde{v}) \quad + \alpha \quad \mathrm{S}(\tilde{u}, \tilde{v}) \quad \mathrm{d}x\mathrm{d}y \\ &= \int_\Omega \gamma \left((\tilde{u} - u)^2 + (\tilde{v} - v)^2 \right) + \alpha \left(\|\mathrm{H}(\tilde{u})\|_F^2 + \|\mathrm{H}(\tilde{v})\|_F^2 \right) \mathrm{d}x\mathrm{d}y \quad , \end{aligned} \tag{4.80}$$

with the indicator function γ for points without flow which is not used in the following equations though. Instead of computing the Euler-Lagrange equations and discretising it afterwards, the energy functional is first discretised to simplify the derivations:

$$\mathrm{E}(\tilde{u}, \tilde{v}) = \sum_{i=1}^{N} \sum_{j=1}^{M} \left(\tilde{u}_{i,j} - u_{i,j} \right)^2 + \left(\tilde{v}_{i,j} - v_{i,j} \right)^2 + \alpha \left(\|\mathrm{H}(\tilde{u}_{i,j})\|_F^2 + \|\mathrm{H}(\tilde{v}_{i,j})\|_F^2 \right) \quad , \tag{4.81}$$

where $u_{i,j}$, $v_{i,j}$, $\tilde{u}_{i,j}$ and $\tilde{v}_{i,j}$ denote the discrete optical flow components evaluated at the point (i,j). Since the two flow components $\tilde{u}$ and $\tilde{v}$ are independent from each other, they can be computed separately. Therefore, to further simplify the following equations, v and $\tilde{v}$ are dropped and only $\tilde{v}$ is computed since the equations can be easily modified to compute $\tilde{v}$ by exchanging the corresponding variables. Therefore, the energy functional is changed in the following way:

$$\mathrm{E}(\tilde{u}) = \sum_{i=1}^{N}\sum_{j=1}^{M}\left(\tilde{u}_{i,j} - u_{i,j}\right)^2 + \alpha\|\mathrm{H}(\tilde{u}_{i,j})\|_F^2 = \sum_{i=1}^{N}\sum_{j=1}^{M}\left(\tilde{u}_{i,j} - u_{i,j}\right)^2 + \alpha\left(\tilde{u}_{xx_{i,j}}^2 + 2\tilde{u}_{xy_{i,j}}^2 + \tilde{u}_{yy_{i,j}}^2\right) \quad . \tag{4.82}$$

The derivatives of the optical flow component $\tilde{u}$ are approximated by using finite differences as follows:

$$\begin{aligned} \tilde{u}_{xx_{i,j}} &= \frac{\tilde{u}_{i-1,j} - 2\tilde{u}_{i,j} + \tilde{u}_{i+1,j}}{h_x^2} \\ \tilde{u}_{yy_{i,j}} &= \frac{\tilde{u}_{i,j-1} - 2\tilde{u}_{i,j} + \tilde{u}_{i,j+1}}{h_y^2} \\ \tilde{u}_{xy_{i,j}} &= \frac{\tilde{u}_{i-1,j-1} - \tilde{u}_{i-1,j+1} - \tilde{u}_{i+1,j-1} + \tilde{u}_{i+1,j+1}}{4h_x h_y} \quad . \end{aligned} \tag{4.83}$$

The equation from the energy functional 4.82 needs to be minimised by differentiating it now with respect to $\tilde{u}_{i,j}$. Because of the complexity of smoothness term, each part is differentiated separately since the sums can be split. The first part is the data term of which the derivative is easy to determine:

$$\partial_{\tilde{u}_{i,j}} \sum_{i=1}^{N}\sum_{j=1}^{M}\left(\tilde{u}_{i,j} - u_{i,j}\right)^2 = 2(\tilde{u}_{i,j} - u_{i,j}) \quad . \tag{4.84}$$

A pixel (i,j) might appear more than once in the approximation of the second derivatives which is why there are different equations depending on whether the considered pixel (i,j) is located at a boundary or not. Mirrored image boundaries are assumed for these following approximations. The derivative of the first part of the smoothness term $\partial_{\tilde{u}_{i,j}}\tilde{u}_{xx}^2$ reads

$$
\begin{aligned}
2\left(\frac{\tilde{u}_{i,j}-2\tilde{u}_{i+1,j}+\tilde{u}_{i+2,j}}{h_x^4}\right) &\quad \text{for } i=1,\\
-4\left(\frac{\tilde{u}_{i-1,j}-2\tilde{u}_{i,j}+\tilde{u}_{i+1,j}}{h_x^4}\right)+2\left(\frac{\tilde{u}_{i,j}-2\tilde{u}_{i+1,j}+\tilde{u}_{i+2,j}}{h_x^4}\right) &\quad \text{for } i=2,\\
2\left(\frac{\tilde{u}_{i-2,j}-2\tilde{u}_{i-1,j}+\tilde{u}_{i,j}}{h_x^4}\right)-4\left(\frac{\tilde{u}_{i-1,j}-2\tilde{u}_{i,j}+\tilde{u}_{i+1,j}}{h_x^4}\right)+2\left(\frac{\tilde{u}_{i,j}-2\tilde{u}_{i+1,j}+\tilde{u}_{i+2,j}}{h_x^4}\right) &\quad \text{for } i=3\ldots N-2,\\
2\left(\frac{\tilde{u}_{i-2,j}-2\tilde{u}_{i-1,j}+\tilde{u}_{i,j}}{h_x^4}\right)-4\left(\frac{\tilde{u}_{i-1,j}-2\tilde{u}_{i,j}+\tilde{u}_{i+1,j}}{h_x^4}\right) &\quad \text{for } i=N-1,\\
2\left(\frac{\tilde{u}_{i-2,j}-2\tilde{u}_{i-1,j}+\tilde{u}_{i,j}}{h_x^4}\right) &\quad \text{for } i=N.
\end{aligned}
\tag{4.85}
$$

The same holds for the derivative of the last part of the smoothness term $\partial_{\tilde{u}_{i,j}}\tilde{u}_{yy}^2$:

$$
\begin{aligned}
2\left(\frac{\tilde{u}_{i,j}-2\tilde{u}_{i,j+1}+\tilde{u}_{i,j+2}}{h_y^4}\right) &\quad \text{for } j=1,\\
-4\left(\frac{\tilde{u}_{i,j-1}-2\tilde{u}_{i,j}+\tilde{u}_{i,j+1}}{h_y^4}\right)+2\left(\frac{\tilde{u}_{i,j}-2\tilde{u}_{i,j+1}+\tilde{u}_{i,j+2}}{h_y^4}\right) &\quad \text{for } j=2,\\
2\left(\frac{\tilde{u}_{i,j-2}-2\tilde{u}_{i,j-1}+\tilde{u}_{i,j}}{h_y^4}\right)-4\left(\frac{\tilde{u}_{i,j-1}-2\tilde{u}_{i,j}+\tilde{u}_{i,j+1}}{h_y^4}\right)+2\left(\frac{\tilde{u}_{i,j}-2\tilde{u}_{i,j+1}+\tilde{u}_{i,j+2}}{h_y^4}\right) &\quad \text{for } j=3\ldots M-2,\\
2\left(\frac{\tilde{u}_{i,j-2}-2\tilde{u}_{i,j-1}+\tilde{u}_{i,j}}{h_y^4}\right)-4\left(\frac{\tilde{u}_{i,j-1}-2\tilde{u}_{i,j}+\tilde{u}_{i,j+1}}{h_y^4}\right) &\quad \text{for } j=M-1,\\
2\left(\frac{\tilde{u}_{i,j-2}-2\tilde{u}_{i,j-1}+\tilde{u}_{i,j}}{h_y^4}\right) &\quad \text{for } j=M.
\end{aligned}
\tag{4.86}
$$

The derivative of the middle part of the smoothness term $\partial_{\tilde{u}_{i,j}} \tilde{u}_{xy}^2$ is more complex since the central pixel appears in the approximation of four pixels. Furthermore, the amount of cases increases a lot due to the fact that neighbours both in x- and in y-direction appear in the equations. Therefore, only the equation for the pixels which are not close to the image boundaries is given here from which the other cases can be derived easily. Additionally, this equation is simplified in such a way that the pixels are factored out and the weights for each pixel are added up:

$$\begin{aligned} &\frac{1}{4h_x^2h_y^2}\tilde{u}_{i-2,j-2} - \frac{1}{2h_x^2h_y^2}\tilde{u}_{i,j-2} + \frac{1}{4h_x^2h_y^2}\tilde{u}_{i+2,j-2} \\ -&\frac{1}{2h_x^2h_y^2}\tilde{u}_{i-2,j} \quad + \frac{1}{h_x^2h_y^2}\tilde{u}_{i,j} \quad - \frac{1}{2h_x^2h_y^2}\tilde{u}_{i+2,j} \\ &\frac{1}{4h_x^2h_y^2}\tilde{u}_{i-2,j+2} - \frac{1}{2h_x^2h_y^2}\tilde{u}_{i,j+2} + \frac{1}{4h_x^2h_y^2}\tilde{u}_{i+2,j+2} \quad . \end{aligned} \tag{4.87}$$

After having derived all equations, it is possible to introduce a weighting function w which holds the weights of the corresponding neighbours. These weights are summed up from all the terms above which are applied to the corresponding pixel. This means that all equations above can be summarised to the following derivative of the discrete energy functional with respect to $\tilde{u}_{i,j}$ after simplifying as follows:

$$\tilde{u}_{i,j} - u_{i,j} + \alpha \sum_{\tilde{i},\tilde{j} \in \mathcal{N}(i,j)} w(\tilde{i}-i, \tilde{j}-j)\tilde{u}_{\tilde{i},\tilde{j}} = 0 \quad , \tag{4.88}$$

where $\mathcal{N}(i,j)$ is a neighbourhood of size 5×5 with the pixel (i,j) in the middle of it. By separating the central pixel from the rest, the fixed point iteration scheme can be set up:

$$\tilde{u}_{i,j}^{k+1}\left(1 + \alpha \cdot w(0,0)\right) = u_{i,j} - \alpha \sum_{\tilde{i},\tilde{j} \in \overline{\mathcal{N}}(i,j)} w(\tilde{i}-i, \tilde{j}-j)\tilde{u}_{\tilde{i},\tilde{j}}^{k} \quad , \tag{4.89}$$

where $\overline{\mathcal{N}}(i,j)$ is the same neighbourhood like before but without the central pixel (i,j). The final equation to compute the new optical flow values $\tilde{u}_{i,j}$ can then be written as

$$\tilde{u}_{i,j}^{k+1} = \frac{u_{i,j} - \alpha \sum_{\tilde{i},\tilde{j} \in \overline{\mathcal{N}}(i,j)} w(\tilde{i}-i, \tilde{j}-j)\tilde{u}_{\tilde{i},\tilde{j}}^{k}}{1 + \alpha \cdot w(0,0)} \quad . \tag{4.90}$$

As mentioned before, the same holds for the values of the second optical flow component $\tilde{v}_{i,j}$ by exchanging the corresponding variables. The application of a penaliser ψ_D on the data term changes the energy functional in the following way:

$$\mathrm{E}(\tilde{u},\tilde{v}) = \int_\Omega \psi_D\left((\tilde{u}-u)^2\right) + \psi_D\left((\tilde{v}-v)^2\right) + \alpha\left(\|\mathrm{H}(\tilde{u})\|_F^2 + \|\mathrm{H}(\tilde{v})\|_F^2\right) \mathrm{d}x\mathrm{d}y \quad . \tag{4.91}$$

The final equation for the computation of $\tilde{u}_{i,j}$ then reads

$$\tilde{u}_{i,j}^{k+1} = \frac{\psi'_{D_{i,j}} u_{i,j} - \alpha \sum_{\tilde{i},\tilde{j} \in \overline{\mathcal{N}}(i,j)} w(\tilde{i}-i, \tilde{j}-j) \tilde{u}_{\tilde{i},\tilde{j}}^{k}}{\psi'_{D_{i,j}} + \alpha \cdot w(0,0)} \ . \tag{4.92}$$

The energy functional with a penaliser ψ_S on the smoothness term is defined as follows:

$$\mathrm{E}(\tilde{u}, \tilde{v}) = \int_\Omega (\tilde{u}-u)^2 + (\tilde{v}-v)^2 + \alpha \left(\psi_S \left(\|\mathrm{H}(\tilde{u})\|_F^2 \right) + \psi_S \left(\|\mathrm{H}(\tilde{v})\|_F^2 \right) \right) \mathrm{d}x\mathrm{d}y \ . \tag{4.93}$$

The derivation of the new equations with a penalisation function for the smoothness term can be set up easier because of the weighting function w which yields

$$\tilde{u}_{i,j}^{k+1} = \frac{u_{i,j} - \alpha \sum_{\tilde{i},\tilde{j} \in \overline{\mathcal{N}}(i,j)} \psi'_{S_{\tilde{i},\tilde{j}}} w(\tilde{i}-i, \tilde{j}-j) \tilde{u}_{\tilde{i},\tilde{j}}^{k}}{\psi'_{D_{i,j}} + \alpha \psi'_{S_{i,j}} \cdot w(0,0)} \ . \tag{4.94}$$

The final step is to evaluate the resulting optical flow components. This time, both the magnitude of the gradients $|\nabla \tilde{u}|^2$ and $|\nabla \tilde{v}|^2$ and the Frobenius norm of the Hessians $\|\mathrm{H}(\tilde{u})\|_F^2$ and $\|\mathrm{H}(\tilde{v})\|_F^2$ are analysed such that it is possible to tell whether the movement is constant, affine or of a higher order. A small gradient value implies a small Hessian value. Hence, the possible cases are the following with $a \in \{\tilde{u}, \tilde{v}\}$:

- $|\nabla a|^2 \leq T_{\nabla a}$ which means that ∇a is small and therefore, the movement is constant in the corresponding direction.
- $|\nabla a|^2 > T_{\nabla a}$ and $\|\mathrm{H}(a)\|_F^2 \leq T_{\mathrm{H}(a)}$ which means that $\|\mathrm{H}(a)\|_F^2$ is small and therefore, the movement is affine in the corresponding direction.
- $|\nabla a|^2 > T_{\nabla a}$ and $\|\mathrm{H}(a)\|_F^2 > T_{\mathrm{H}(a)}$ which means that neither of them is small and the movement is of higher order in the corresponding direction.

4.4.2 *Large Displacements*

The last topic with respect to movement is about large displacements. As mentioned before, the magnitude of an optical flow vector denotes the length of the displacement at this point. However, a large displacement might be different for each image sequence, depending e.g. on the number and size of objects in the scene. To get an idea of how many pixels might have a large displacement, the following three categories "$\geq 0.5 \cdot \max\left(\left|(u,v)^{\mathrm{T}}\right|\right)$", "$\geq 0.8 \cdot \max\left(\left|(u,v)^{\mathrm{T}}\right|\right)$" and "$\geq 0.9 \cdot \max\left(\left|(u,v)^{\mathrm{T}}\right|\right)$" are used, where $\max\left(\left|(u,v)^{\mathrm{T}}\right|\right)$ denotes the biggest length of all optical flow vectors in the considered image sequence. This means that $\max\left(\left|(u,v)^{\mathrm{T}}\right|\right)$ can be different for each image sequence and therefore the number of pixels with a large displacement

is relative to this value. However, this way it is possible to tell how many pixels out of all pixels where the optical flow is defined are in each category. Furthermore, the percentage of all pixels with a rather large displacement can be estimated.

4.5 FUNDAMENTAL MATRIX ESTIMATION

This topic regards the similarity of an image sequence to a stereo image sequence which means that the potential of the scene to be a stereo scene should be measured. It is possible to compute an optical flow field between two stereo images which is for example proposed by the Middlebury Benchmark. The fourth data set of this benchmark uses modified stereo data. The interesting aspect about the similarity between the observed scene and a stereo scene considering optical flow is to find out whether there exists egomotion of the camera. To measure this, the Fundamental Matrix of the image sequence is estimated by using the optical flow. Without going to deep into stereo geometry, some basics about the Fundamental Matrix and its estimation are discussed in the following. More details can be found, e.g. in [19], [20].

To estimate the Fundamental Matrix F of size 3×3, the so-called epipolar constraint $p_2^{\mathrm{T}} F^{\mathrm{T}} p_1 = 0$ is fulfilled for the two corresponding points $p_1 = (x_1, y_1, 1)^{\mathrm{T}}$ and $p_2 = (x_2, y_2, 1)^{\mathrm{T}}$. This equation can be reformulated as $p^{\mathrm{T}} f = 0$, where

$$\begin{aligned} p &= (x_1 x_2, y_1 x_2, x_2, x_1 y_2, y_1 y_2, y_2, x_1, y_1, 1)^{\mathrm{T}} \ , \\ f &= (f_{11}, f_{12}, f_{13}, f_{21}, f_{22}, f_{23}, f_{31}, f_{32}, f_{33})^{\mathrm{T}} \ . \end{aligned} \tag{4.95}$$

By using the optical flow, corresponding points in both images can be found. For each of these correspondences i, a correspondence constraint $p_i^{\mathrm{T}} f = 0$ can be set up. The additional constraint which has to be fulfilled is a rank constraint because F has to have rank 2. To compute a Fundamental Matrix, only 8 corresponding point pairs are needed. But since probably there exist more than these 8 correspondences, the following total least squares fit has to be minimised for $N \geq 8$ correspondences:

$$\mathrm{E}(f) = \sum_{i=1}^{N} \left(p_i^{\mathrm{T}} f \right)^2 \ . \tag{4.96}$$

Because of the trivial solution $f = 0$, the constraint $|f| = 1$ is added as Lagrangian multiplier $(1 - f^{\mathrm{T}} f)$. This leads to the modified equation

$$\mathrm{E}^*(f) = \sum_{i=1}^{N} \left(p_i^{\mathrm{T}} f \right)^2 + \lambda (1 - f^{\mathrm{T}} f) = f^{\mathrm{T}} P^{\mathrm{T}} P f + \lambda (1 - f^{\mathrm{T}} f) \ , \tag{4.97}$$

where P holds all the point correspondences as follows:

$$P = \begin{pmatrix} p_1^{\mathrm{T}} \\ p_2^{\mathrm{T}} \\ \vdots \\ p_N^{\mathrm{T}} \end{pmatrix} \quad . \tag{4.98}$$

The total least squares fit $\mathrm{E}(f)$ can be minimised now by differentiating $\mathrm{E}^*(f)$ with respect to f and λ which comes down to solving the eigenvalue problem

$$(P^{\mathrm{T}}P - \lambda \mathrm{I})f = 0, \quad |f| = 1 \ . \tag{4.99}$$

This means that the solution for f is the eigenvector to the smallest eigenvalue of the symmetric 9×9 matrix $P^{\mathrm{T}}P$. However, the problem that arises with this total least squares fit is that outliers like flawed correspondences might have a bad influence on the result. This problem can be solved by using M-estimators where a subquadratic penaliser ψ like the one proposed above is applied on the correspondence constraint, resulting in the following equation:

$$\mathrm{E}^*(f) = \sum_{i=1}^{N} \psi\left(\left(p_i^{\mathrm{T}} f\right)^2\right) + \lambda(1 - f^{\mathrm{T}} f) \ . \tag{4.100}$$

This leads to a nonlinear system of equations by differentiating the equation with respect to f and λ:

$$(P^{\mathrm{T}}WP - \lambda \mathrm{I})f = 0, \quad |f| = 1 \ , \tag{4.101}$$

where W holds the penalisation weights $w_{ii} = \psi'\left(\left(p_i^{\mathrm{T}} f\right)^2\right)$ on its diagonal. This nonlinear system can be solved as a series of linear systems by applying the lagged nonlinearity method like for the global joint illumination changes with subquadratic penaliser where several iterations of alternating between the calculations of f and the recomputations of the weights w_{ii} are done until the solution is satisfying.
A problem of the expression $p_2^{\mathrm{T}} F^{\mathrm{T}} p_1$ is that it has no geometrical meaning and is only used to enforce the correspondence constraints. Therefore, the idea of geometric approaches is to minimise the distance of each point p_{2_i} to its epipolar line l_{2_i}. This assumption leads to the following normalised nonlinear distance measure:

$$d_{\mathrm{NL}}(p_{2_i}, Fp_{1_i}) = \frac{1}{\sqrt{(Fp_{1_i})_1^2 + (Fp_{1_i})_2^2}} p_{2_i}^{\mathrm{T}} F^{\mathrm{T}} p_{1_i} \ , \tag{4.102}$$

where $(Fp_{1_i})_1$ and $(Fp_{1_i})_2$ are the two first entries of the resulting vector. This kind of approach is used to evaluate the estimated Fundamental Matrix and to measure the average distance of points to their epipolar lines. Using this distance measure, an energy function can be defined from it as follows:

$$\mathrm{E}(F) = \sum_{i=1}^{N} d_{\mathrm{NL}}^2(p_{2_i}, Fp_{1_i}) = \sum_{i=1}^{N} \frac{1}{(Fp_{1_i})_1^2 + (Fp_{1_i})_2^2 + \lambda^2} \left(p_{2_i}^{\mathrm{T}} F^{\mathrm{T}} p_{1_i}\right)^2 \quad , \tag{4.103}$$

where a small λ^2 is added to prevent division by 0 or values very close to 0. To compute the average distance of a point to its epipolar line in pixels, the energy function is modified as follows

$$\mathrm{E}(F) = \frac{1}{N} \sum_{i=1}^{N} \sqrt{\frac{1}{(Fp_{1_i})_1^2 + (Fp_{1_i})_2^2 + \lambda^2} \left(p_{2_i}^{\mathrm{T}} F^{\mathrm{T}} p_{1_i}\right)^2} \quad . \tag{4.104}$$

The resulting value can then give a hint on whether the scene is a stereo one or not because stereo scenes contain mainly egomotion. Thus, it is easily possible to estimate a Fundamental Matrix such that the average distance of points to their epipolar lines is less than 1 pixel.

5. EVALUATION

In this section, the previous introduced metrics are computed for the four benchmarks Middlebury, KITTI 2012, KITTI 2015 and MPI-Sintel and their results are afterwards evaluated and compared such that the difficulty of the benchmarks can be classified. Only the images of the Clean pass were used to evaluate the MPI-Sintel Benchmark. Since the metrics regard different topics and difficulties for optical flow computations, the benchmark's strengths and weaknesses concerning these metrics can be investigated. In the first step, these metrics are calculated for the training data sets of the benchmarks by using the given ground truth flow. Therefore, this should give quite accurate results to represent the benchmark's difficulties. In the second step, the difficulties of the testing data sets are investigated by using computed optical flow which is why these results have to be seen as estimations. The metrics are calculated for each image pair separately and mean values of them represent then the overall benchmark. Due to the huge amount of data, all the metrics' results are only presented for the whole benchmarks.

5.1 TRAINING DATA SETS WITH GROUND TRUTH FLOW

5.1.1 *Image Statistics*

The first metrics are about image statistics and help to get an impression on the images themselves. The first statistics characterise the images in general in terms of minimal, maximal and average pixel values and the standard deviation. The tables 5.1, 5.2, 5.3

Table 5.1: Image statistics for the training data set of the Middlebury Benchmark.

Pixel Values	Minimum	Maximum	Average	Stand. Dev.
Middlebury Frame 0 Channel 0	10.25	241.25	117.137	56.2332
Middlebury Frame 1 Channel 0	10.375	242.5	117.543	56.271
Middlebury Frame 0 Channel 1	1.875	237.625	87.2295	79.3969
Middlebury Frame 1 Channel 1	2	237.625	87.6217	79.5372
Middlebury Frame 0 Channel 2	1	223.25	71.4081	89.1656
Middlebury Frame 1 Channel 3	1	221.375	71.5979	89.5376
Middlebury Overall Mean	4.4167	233.9375	92.0895	75.0236

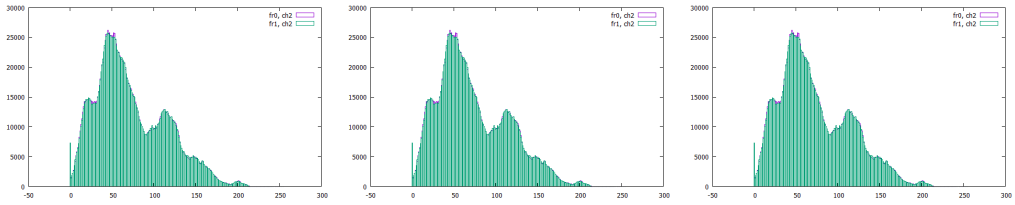

Figure 5.1: The accumulated channelwise histogram of both frames of the training data set of the Middlebury Benchmark. **Left:** First channel. **Middle:** Second channel. **Right:** Third channel.

and 5.4 show the corresponding values together with the overall mean. As can be seen from the tables, the pixel values of all four benchmarks use almost the full range of values from 0 to 255. It is possible to tell from the average pixel values that all the images are rather a bit darker, especially the images of the MPI-Sintel Benchmark. The pixel values are best distributed for the KITTI 2012 and the KITTI 2015 benchmarks. Furthermore, the values of both frames don't differ much from each other which is also true for the channels.

Table 5.2: Image statistics for the training data set of the KITTI 2012 Benchmark.

Pixel Values	Minimum	Maximum	Average	Stand. Dev.
KITTI 2012 Frame 0	8.0309	255	93.4166	81.0183
KITTI 2012 Frame 1	8.1031	255	93.2129	80.8894
KITTI 2012 Overall Mean	8.067	255	93.3148	80.9539

Table 5.3: Image statistics for the training data set of the KITTI 2015 Benchmark.

Pixel Values	Minimum	Maximum	Average	Stand. Dev.
KITTI 2015 Frame 0 Channel 0	0.015	255	96.6732	88.7176
KITTI 2015 Frame 1 Channel 0	0.02	255	96.5105	88.5708
KITTI 2015 Frame 0 Channel 1	2.625	255	101.607	88.1858
KITTI 2015 Frame 1 Channel 1	2.695	255	101.48	88.0605
KITTI 2015 Frame 0 Channel 2	0.08	255	97.8312	91.8683
KITTI 2015 Frame 1 Channel 3	0.075	255	97.7844	91.7143
KITTI 2015 Overall Mean	0.918	255	98.6477	89.5196

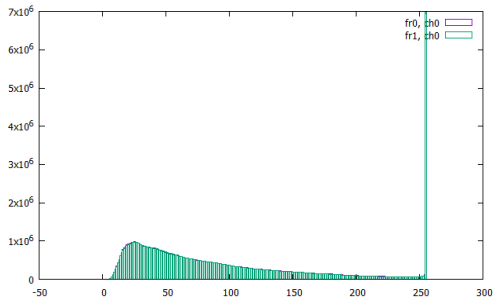

Figure 5.2: The accumulated histogram of both frames of the training data set of the KITTI 2012 Benchmark.

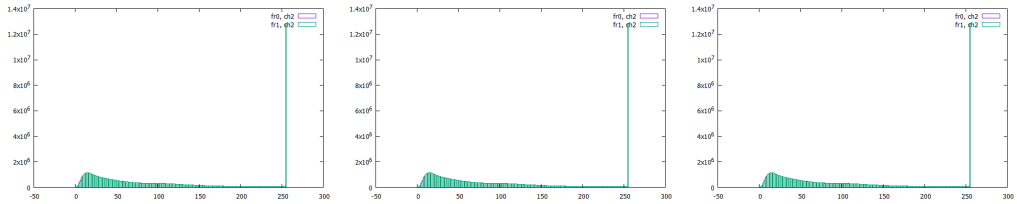

Figure 5.3: The accumulated channelwise histogram of both frames of the training data set of the KITTI 2015 Benchmark. **Left:** First channel. **Middle:** Second channel. **Right:** Third channel.

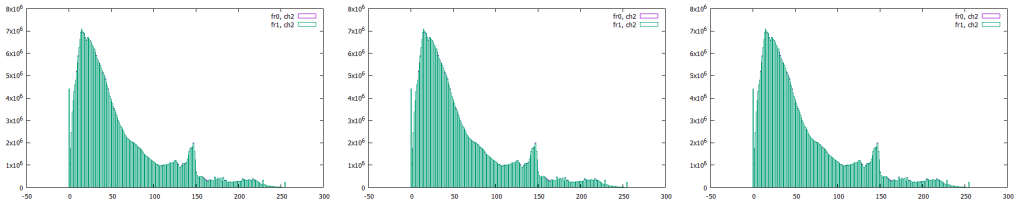

Figure 5.4: The accumulated channelwise histogram of both frames of the training data set of the MPI-Sintel Benchmark. **Left:** First channel. **Middle:** Second channel. **Right:** Third channel.

The histograms, shown in the figures 5.1, 5.2, 5.3 and 5.4, represent a more accurate distribution of the pixel values. As can be seen from the images, the assumption that all images are rather dark can be confirmed, especially for the MPI-Sintel Benchmark. Furthermore, it is observable that the highest value 255 appears extremely often in the images of the KITTI 2012 and of the KITTI 2015 benchmarks which probably have another reason than the appearance in the image. Since these image statistics are too weak to be used as difficulty metric they are good to give an impression of the ap-

Table 5.4: Image statistics for the training data set of the MPI-Sintel Benchmark.

Pixel Values	Minimum	Maximum	Average	Stand. Dev.
MPI-Sintel Frame 0 Channel 0	0.1037	242.646	76.3991	58.3217
MPI-Sintel Frame 1 Channel 0	0.1316	242.631	76.4472	58.336
MPI-Sintel Frame 0 Channel 1	0.146	242.937	69.424	57.814
MPI-Sintel Frame 1 Channel 1	0.1844	242.927	69.4451	57.833
MPI-Sintel Frame 0 Channel 2	0.1652	235.934	58.0139	63.5252
MPI-Sintel Frame 1 Channel 3	0.2085	235.882	58.0308	63.5418
MPI-Sintel Overall Mean	0.1566	240.4928	67.96	59.8953

pearing pixel values. Nevertheless, in addition to the distribution, the histograms can reveal irregularities like the ones from the KITTI benchmarks.

5.1.2 *Optical Flow Statistics*

The optical flow statistics should give an impression of the difficulty of the optical flow in general. Two statistical values, the magnitude and the angle, are interesting for this metric. The magnitude of an optical flow vector denotes the speed of the corresponding point while the angle is important for the direction. The magnitude statistics are composed of the minimal, maximal and average magnitude and its standard deviation together with a histogram where buckets of size 2 are used. The general overview can be seen in table 5.5. All benchmarks have optical flow vectors with a small magnitude but only the two KITTI benchmarks offer very large displacements. Especially the KITTI 2015 Benchmarks has an average maximal disparity value of about 200 pixels. The MPI-Sintel benchmark has a high mean maximal magnitude value as well. Due to the designing of the Middlebury Benchmark, the maximal length is rather short. Con-

Table 5.5: Magnitude statistics for the training data sets.

Magnitude Values	Minimum	Maximum	Average	Stand. Dev.
Middlebury	0.5339	11.6532	4.19377	2.4224
KITTI 2012	2.4621	185.831	33.8806	28.0062
KITTI 2015	0.672	203.285	36.7127	32.7505
MPI-Sintel	1.2885	62.8199	13.4957	9.8179

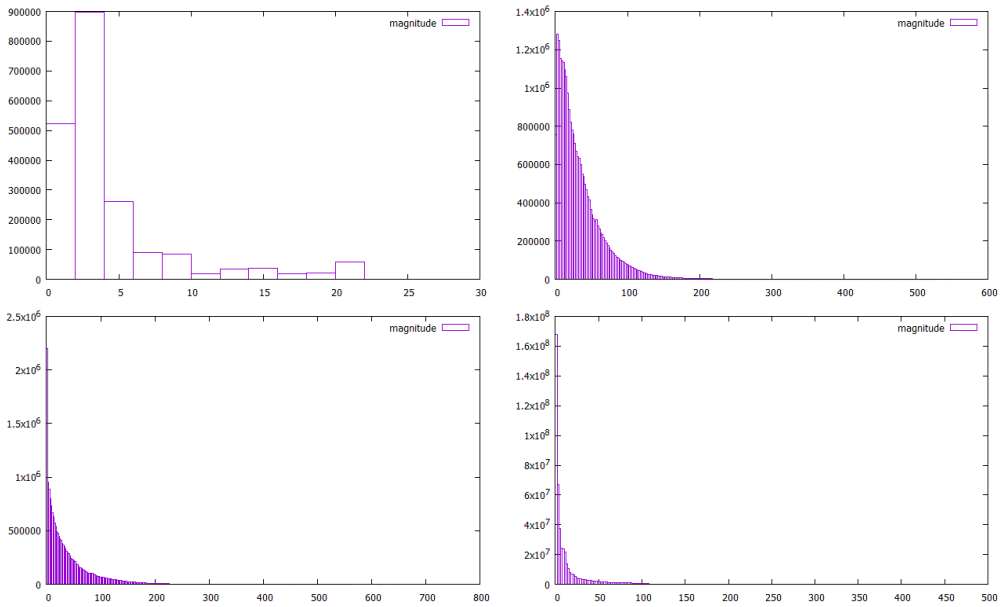

Figure 5.5: The accumulated magnitude histogram of the optical flow of the training data sets. **Upper left:** Middlebury. **Upper right:** KITTI 2012. **Lower left:** KITTI 2015. **Lower right:** MPI-Sintel.

sidering the magnitude, the Middlebury Benchmark is therefore easier than the others and both KITTI benchmarks offer a rather wide range of magnitude values which can be seen from the standard deviation as well. The magnitude histogram, depicted in figure 5.5, shows the real distribution of the magnitude values. The most values are of course small since small displacements are most common in optical flow. Considering the histogram, the KITTI 2012 Benchmark has in comparison with the other benchmarks the highest amount of large optical flow vectors which makes it more difficult than the others.

To find out which directions are the most prominent ones in the benchmarks and if the angles are well distributed, an accumulated histogram can be used where all the angles of the optical flow vectors are added up for each benchmark. These histograms are shown in figure 5.6. It is obvious to see that some angles are appearing quite often while others are very rare. Furthermore, the main optical flow directions which are along the main axes seem to be rather similar for most of the benchmarks. Most of the angles from the Middlebury Benchmark are around $0°$ while the second most common vector direction is around $180°$ which means that the main directions are along the x-axis. This is easy to explain with the design of this benchmark since most camera and object movements were along this axis which is why the optical flow vectors point in the same direction. These directions seem to be the main directions for the KITTI 2012

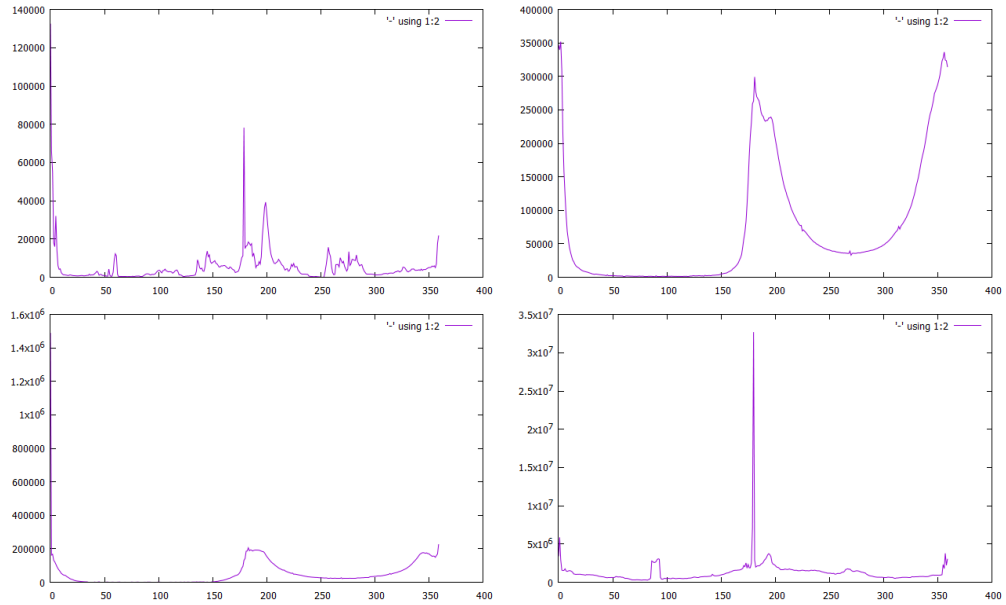

Figure 5.6: The accumulated angle histogram of the optical flow of the training data sets. **Upper left:** Middlebury. **Upper right:** KITTI 2012. **Lower left:** KITTI 2015. **Lower right:** MPI-Sintel.

and the KITTI 2015 benchmarks as well. The reason for this is also the design of these benchmarks. While moving along a path, static objects which are on the left and right side of the view become bigger and move more and more to the side when passing them. However, the positive x-direction is much more prominent for the KITTI 2015 benchmark than the negative direction. In contrast to this, the main direction of optical flow vectors of the MPI-Sintel Benchmark is the negative x-direction. Considering the distribution of angles, the Middlebury Benchmark has many small fluctuations which means that many other directions appear. The distribution for the two KITTI benchmarks is quite smooth, even though angles between 50° and 150° don't seem to appear. It can be seen in the histogram of the MPI-Sintel Benchmark that some smaller fluctuations exist as well which is why more different optical flow directions appear in the image sequences. These fluctuations increase the difficulty in contrast to the smooth distribution because a smooth distribution can be interpreted in such a way that neighbouring points have a similar direction and only change their direction slowly which holds for the whole image.

5.1.3 *Illumination*

To get a general impression of possible illumination changes, the difference of the two images can be computed where the second image has been compensated by the ground truth flow. The results of subtracting the two frames are listed in table 5.6. The Middlebury, the KITTI 2012 and MPI-Sintel benchmarks have similar average difference values while the KITTI 2015 Benchmark's difference value is nearly twice as much. Nevertheless, the values are rather small which means that the brightness differences between the two frames are small as well.

This assumption can be varified by having a look on the estimated components of the global illumination change, listed in table 5.7. It is obvious that the multiplicative component is always very close to 1 for both energy functions. This means that the multiplicative component doesn't change the brightness very much, making it slightly darker since the values are a bit smaller than 1. The multiplicative component's influence increases with the pixel value which means that high pixel values are more affected by it than smaller ones. Due to the fact that the images are a bit darker, the impact on the pixel values is not that big. Considering the additive components which result from the non-penalised energy function, a small change in brightness is observ-

Table 5.6: Difference statistics for the training data sets.

Difference Values	Average	Stand. Dev.
Middlebury	6.1319	10.0621
KITTI 2012	6.8609	11.8007
KITTI 2015	11.6291	19.1685
MPI-Sintel	6.2048	13.1499

Table 5.7: Global illumination change components for the training data sets. The components that have been computed using a robust energy function on which a subquadratic penaliser has been applied are listed in the last two columns.

Comp. Values	Mult. comp.	Add. comp.	Mult. comp. (rob.)	Add. comp. (rob.)
Middlebury	0.9602	3.3564	0.9966	0.3444
KITTI 2012	0.9727	1.7961	0.9953	−0.0517
KITTI 2015	0.9234	5.7892	0.9838	0.1767
MPI-Sintel	0.9348	3.724	0.9943	0.2641

able. However, the additive components computed by the penalised energy function are around 0 which would mean that the brightness is preserved. It can be concluded that the higher values of the additive component are caused by outliers. This is probably the reason for the average difference values as well. Overall it can be said that, considering the global illumination change, it is brightness preserving for all four benchmarks.

Table 5.8: Statistics for the multiplicative component of local illumination changes for the training data sets. The second row of each benchmark contains the values for the robust energy function.

Comp. Values	Minimum	Maximum	Average	Stand. Dev.
Middlebury	−8.1384	23.5703	1.0095	0.5573
Middlebury (rob.)	−0.3077	34.3344	1.0179	0.5817
KITTI 2012	0.0035	8.4984	0.916	0.335
KITTI 2012 (rob.)	0.002	8.5055	0.8926	0.3857
KITTI 2015	−0.5695	32.9675	0.9305	0.6299
KITTI 2015 (rob.)	−0.0106	38.1133	0.9213	0.6648
MPI-Sintel	−11.0553	30.5066	0.9824	0.6695
MPI-Sintel (rob.)	−0.8968	47.7127	1.0201	0.7181

Table 5.9: Statistics for the additive component of local illumination changes for the training data sets. The second row of each benchmark contains the values for the robust energy function.

Comp. Values	Minimum	Maximum	Average	Stand. Dev.
Middlebury	−0.5154	65.9378	0.1857	1.2138
Middlebury (rob.)	−0.5344	62.9758	0.0355	0.4528
KITTI 2012	−0.0226	0.5601	0.0109	0.0136
KITTI 2012 (rob.)	−0.0016	0.4764	0.0193	0.0178
KITTI 2015	−0.1863	67.389	0.0819	0.7484
KITTI 2015 (rob.)	−0.0913	35.8122	0.0293	0.2147
MPI-Sintel	−0.6797	174.396	0.936	4.7492
MPI-Sintel (rob.)	−0.7139	171.024	0.1009	1.5966

To investigate this topic further, local illumination changes are assumed. Statistical values like minimum, maximum, average and standard deviation have been computed from the resulting components to get a better overview which are listed in the tables 5.8 and 5.9. The first impression of the range of values is interesting since it is larger than one might expect from the global components' values. There exist even quite large negative values for the multiplicative component of the Middlebury and the MPI-Sintel benchmarks which have to be the results of outliers since the values of the robust energy function are much smaller. Considering the additive component, the maximal value for the MPI-Sintel Benchmark is quite high as well as for the Middlebury and the KITTI 2015 benchmarks, even after penalising outliers. Nevertheless, the average values of the multiplicative component are around 1 and the ones of the additive component are around 0 again which means that the extrema are very rare and that the brightness in general is preserved. However, it seems like there are some local illumination changes for some objects. These are hard to take into account when computing optical flow, especially if they are as high as the minimal and maximal component values in the table. Therefore, the MPI-Sintel Benchmark is more difficult than the others because some of the local illumination changes are quite high while the two KITTI benchmarks offer only small illumination changes.

5.1.4 *Movement*

The metrics in this section try to estimate the type of movement. Two models were used to describe the movement, a constant and an affine one. The x-displacement u and the y-displacement v of the optical flow are evaluated separately. The assumption of global movement should give a first impression about the movement type. These values of the movement component are listed both for the constant movement model and the affine movement model in the tables 5.10 and 5.11. Considering the constant movement model first, the values of the constant parameter of the x-displacement are rather small, especially for the Middlebury and the MPI-Sintel benchmarks. However, the components for v are a bit larger for the KITTI 2012 and the KITTI 2015 bench-

Table 5.10: Average value of the movement components for the training data sets, assuming global constant movement.

Comp. Values	u_{const}	v_{const}	u_{const} (rob.)	v_{const} (rob.)
Middlebury	0.4479	1.1676	0.6185	0.8865
KITTI 2012	4.7167	8.8884	2.2001	6.0501
KITTI 2015	2.08124	10.8353	−0.0579	7.1091
MPI-Sintel	−0.6118	1.4969	−1.4662	1.5037

marks. This can be explained by having a look on the image acquisition method that has been used. Since the car is moving forward, most of the movement happens in y-direction which is why the values for the v-component are higher. When assuming global affine movement, the multiplicative components for the x- and y-values of u and v are all very close to 0 from which the conclusion can be drawn that the constant model suffices to describe the global movement. Global movement in general should ease the computation of optical flow, especially if it is constant and not affine because this would mean that all points have more or less the same kind of movement with constant speed. Therefore, the computation of optical flow for the two KITTI benchmarks should be easier because of the higher values for the global constant movement.

Instead of estimating the components of the assumed model for the local movement, the optical flow has been reconstructed from the given flow while filling in the missing information by using a smoothness term, depending on the assumed movement type. Therefore, the gradient magnitude and the Frobenius norm of the Hessian are computed to characterise the type of movement. Points with a small gradient are more probable to have a constant movement while points with a large gradient and a small Hessian are more likely to have an affine movement. To do this, the average number of pixels with small gradient or small Hessian have been computed. The chosen threshold T for small gradients and Hessians is defined to be half of the median of all gradient respectively Hessian values. Assuming local constant movement first, around $33 - 38\%$ of all points have a small u-gradient while the average amount of points with a small v-gradient ranges from around $32 - 45\%$. The amount of points with a small Hessian is only slightly higher than the one for small gradients except for the MPI-Sintel Benchmark. This means that the assumption of constant movement fits to around $30 - 40\%$ of the points of most benchmarks while the MPI-Sintel Benchmark

Table 5.11: Average value of the movement components for the training data sets, assuming global affine movement.

Comp. Values	$u_{x-\text{dir}}$	$u_{y-\text{dir}}$	u_{const}	$v_{x-\text{dir}}$	$v_{y-\text{dir}}$	v_{const}
Middlebury	-0.0004	-0.0096	-0.6595	0.0006	0.007	1.171
Middlebury (rob.)	-0.0011	-0.0075	0.2397	0	0.0073	1.2499
KITTI 2012	0.1107	0.0002	4.9484	0.0016	0.1366	-0.1843
KITTI 2012 (rob.)	0.0968	-0.0067	4.4286	0.0013	0.1228	0.0182
KITTI 2015	0.1031	0.0143	2.3811	0.0021	0.1747	-1.6154
KITTI 2015 (rob.)	0.0931	0.0246	0.4916	0.0013	0.1629	-1.8882
MPI-Sintel	0.0022	-0.0052	-0.8402	0.0001	0.0037	1.5281
MPI-Sintel (rob.)	0.0009	-0.0036	-1.0579	0	0.0021	1.6426

has several points with movement of higher order. Constant movement in general is easier to approximate which means that the optical flow for the Middlebury Benchmark should be easier to compute than the optical flow for the MPI-Sintel Benchmark. By using the Hessian in the smoothness term, the amount of pixels with affine move-

Table 5.12: Average number of all points with a small gradient or Hessian for the training data sets given in percent, assuming local constant movement. A small gradient or Hessian depends on the threshold T which is half the median of all gradient respectively Hessian values.

Average percentage	$\lvert\nabla u\rvert^2 \leq T_{\nabla u}$	$\lvert\nabla v\rvert^2 \leq T_{\nabla v}$	$\lVert \mathrm{H}(u)\rVert_F^2 \leq T_{\mathrm{H}(u)}$	$\lVert \mathrm{H}(v)\rVert_F^2 \leq T_{\mathrm{H}(v)}$
Middlebury	35.8534	44.1178	38.2497	45.3708
Middlebury (rob.)	35.906	45.0746	39.2222	46.5125
KITTI 2012	32.6168	34.3531	37.4571	32.866
KITTI 2012 (rob.)	32.6978	34.4343	35.746	31.4723
KITTI 2015	37.2228	38.3935	41.844	41.4227
KITTI 2015 (rob.)	38.4307	39.0153	42.4168	41.844
MPI-Sintel	32.7102	32.0489	43.2863	40.5167
MPI-Sintel (rob.)	35.0792	33.6748	45.0095	41.874

Table 5.13: Average number of all points with a small gradient or Hessian for the training data sets given in percent, assuming local affine movement. A small gradient or Hessian depends on the threshold T which is half the median of all gradient respectively Hessian values.

Average percentage	$\lvert\nabla u\rvert^2 \leq T_{\nabla u}$	$\lvert\nabla v\rvert^2 \leq T_{\nabla v}$	$\lVert \mathrm{H}(u)\rVert_F^2 \leq T_{\mathrm{H}(u)}$	$\lVert \mathrm{H}(v)\rVert_F^2 \leq T_{\mathrm{H}(v)}$
Middlebury	35.7911	43.6936	38.6964	45.3185
Middlebury (rob.)	35.9084	43.6194	39.6397	46.6244
KITTI 2012	40.9517	41.8109	39.5141	41.7196
KITTI 2012 (rob.)	44.693	45.3139	40.961	42.4184
KITTI 2015	42.9452	43.6258	41.8914	43.5321
KITTI 2015 (rob.)	46.9464	46.5966	42.5017	43.9057
MPI-Sintel	33.2392	32.2664	43.9334	42.1703
MPI-Sintel (rob.)	33.8105	32.3902	44.2518	44.3322

ment should be detected. However, using this assumption leads to a higher amount of points with a small gradient for the two KITTI benchmarks whereas the Middlebury and the MPI-Sintel benchmarks are not affected by the change of the smoothness term. Especially considering the amount of points with a small Hessian is similar to the number of points with a small Hessian when using the gradient in the smoothness term. Therefore, these values seem to be reliable which means that the MPI-Sintel Benchmark has a higher amount of points with affine movement which makes it more difficult while the points of the other benchmarks move in a rather constant way.

Considering the amount of points which have a large displacement relative to the maximal displacement, only about $3-5\%$ of all points of the KITTI 2012, the KITTI 2015 and the MPI-Sintel benchmarks have a displacement with a magnitude higher than half of the maximal displacement. Nearly 10% of all points of the Middlebury Benchmark have a larger displacement. For the factor 0.9, the amount of points decreases to $0-1\%$ for all benchmarks. These results are shown in table 5.14. By including the collected information about the magnitude statistics of the benchmarks, it is obvious that a large displacement of the Middlebury Benchmark is quite short in comparison with the other benchmarks. This means for the other benchmarks that $3-5\%$ of all points have a really large displacement which increases the difficulty.

5.1.5 *Fundamental Matrix Estimation*

It might be possible that egomotion of the camera is mainly responsible for the optical flow in a considered scene which is why these scenes are of the type stereo. In these cases, the relation of the points in the two images can be described by the Fundamental Matrix. Therefore, the Fundamental Matrix has been computed for all the image pairs of the benchmarks and evaluated by computing the average distance of points to their corresponding epipolar lines. The averaged values for the whole benchmark are listed in table 5.15. As can be seen, the Middlebury Benchmark has very low results for both

Table 5.14: Average number of all points with a large displacement for the training data sets given in percent. A large displacement is considered if the magnitude of the flow vector is equal or larger than the factor times the maximal displacement in the considered image sequence.

Average percentage	Factor 0.5	Factor 0.8	Factor 0.9
Middlebury	9.9071	2.1827	1.1092
KITTI 2012	2.938	0.3618	0.0929
KITTI 2015	3.3033	0.9913	0.5261
MPI-Sintel	5.0103	0.9552	0.374

Table 5.15: Average distance of a point to its epipolar line for the training data sets given in pixels.

Average distance	Total Least Squares Fit	M-Estimators
Middlebury	0.18472	0.181
KITTI 2012	1.23901	1.27085
KITTI 2015	1.33623	1.35292
MPI-Sintel	0.70561	1.06058

estimation methods. The reason for this is that most scenes contain mainly egomotion and therefore, their correspondences can be easily described by a Fundamental Matrix. The MPI-Sintel has a slightly higher value because the scenes often contain objects which move in different directions than the camera.

In comparison to the other benchmarks, the two KITTI benchmarks have slightly higher distance values which are higher than 1 pixel. Since the images were acquired by a camera which was mounted on a car, the motion in the images is, except for some independently moving cars, egomotion of the camera which is why these scenes should be considered to be stereo scenes. The distance values are slightly higher because the ground truth flow is sparse and outliers close to the epipoles have a higher impact on the resulting distance. Nevertheless, it can be concluded that the MPI-Sintel Benchmark and the two KITTI benchmarks have some stereo elements but their scenes contain enough independently moving objects as well.

5.2 TESTING DATA SETS WITH COMPUTED OPTICAL FLOW

A computed optical flow has to be used to determine the difficulty of the testing data sets since no ground truth flow is available for it. The optical flow which has been used in this thesis has been computed by using a first order approach by Zimmer et al. [21]. Therefore, the results for the testing data sets have to be seen as estimation of the difficulty since the computed flow is not as exact as the ground truth flow and hence the results depend on the quality of the computed flow.

5.2.1 *Image Statistics*

By having a look on the tables 5.16, 5.17, 5.18 and 5.19, it is easy to see that the testing data sets of the benchmarks have similar values for the image statistics which means that nearly the whole pixel value range is used and it can be assumed from the average pixel value that all images are slightly darker, especially the images of the MPI-Sintel

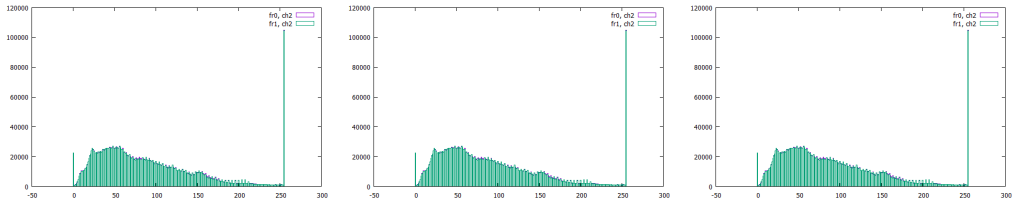

Figure 5.7: The accumulated channelwise histogram of both frames of the testing data set of the Middlebury Benchmark. **Left:** First channel. **Middle:** Second channel. **Right:** Third channel.

Benchmark again. Furthermore, the values of the two frames differ only slightly as well as the ones of the channels.

The assumption of slightly darker images can be supported by the histograms, shown in the figures 5.7, 5.8, 5.9 and 5.10. Especially the MPI-Sintel Benchmark has many very small pixel values. High occurrences of pixel values which appear to be irregularities are present in all benchmarks this time.

Table 5.16: Image statistics for the testing data set of the Middlebury Benchmark.

Pixel Values	Minimum	Maximum	Average	Stand. Dev.
Middlebury Frame 0 Channel 0	5.9167	247.333	114.493	77.2011
Middlebury Frame 1 Channel 0	5.8333	247.5	114.668	77.1629
Middlebury Frame 0 Channel 1	6.5833	241.167	110.612	79.7298
Middlebury Frame 1 Channel 1	6.9167	241.583	110.746	79.7321
Middlebury Frame 0 Channel 2	4.5833	235.667	90.266	95.9652
Middlebury Frame 1 Channel 3	4.5	236.417	90.2505	96.1475
Middlebury Overall Mean	5.7222	241.6112	105.1726	84.3231

Table 5.17: Image statistics for the testing data set of the KITTI 2012 Benchmark.

Pixel Values	Minimum	Maximum	Average	Stand. Dev.
KITTI 2012 Frame 0	7.6	255	93.0539	82.7634
KITTI 2012 Frame 1	7.4513	255	93.0087	82.7304
KITTI 2012 Overall Mean	7.5257	255	93.0313	82.7469

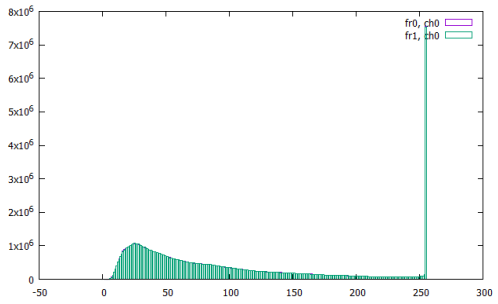

Figure 5.8: The accumulated histogram of both frames of the testing data set of the KITTI 2012 Benchmark.

Table 5.18: Image statistics for the testing data set of the KITTI 2015 Benchmark.

Pixel Values	Minimum	Maximum	Average	Stand. Dev.
KITTI 2015 Frame 0 Channel 0	0	255	94.2869	86.6727
KITTI 2015 Frame 1 Channel 0	0	255	93.9921	86.6926
KITTI 2015 Frame 0 Channel 1	2.76	255	100.122	86.4718
KITTI 2015 Frame 1 Channel 1	2.545	255	99.8795	86.5092
KITTI 2015 Frame 0 Channel 2	0.005	255	97.6825	90.1799
KITTI 2015 Frame 1 Channel 3	0.025	255	97.4344	90.191
KITTI 2015 Overall Mean	0.8892	255	97.2329	87.7862

Table 5.19: Image statistics for the testing data set of the MPI-Sintel Benchmark.

Pixel Values	Minimum	Maximum	Average	Stand. Dev.
MPI-Sintel Frame 0 Channel 0	0	229.33	77.4633	56.4286
MPI-Sintel Frame 1 Channel 0	0	229.194	77.5285	56.4675
MPI-Sintel Frame 0 Channel 1	0	227.005	72.7133	54.5535
MPI-Sintel Frame 1 Channel 1	0	226.822	72.7387	54.6143
MPI-Sintel Frame 0 Channel 2	0	226.194	62.2791	59.9456
MPI-Sintel Frame 1 Channel 3	0	226.027	62.2662	60.0339
MPI-Sintel Overall Mean	0	227.4287	70.8315	57.0072

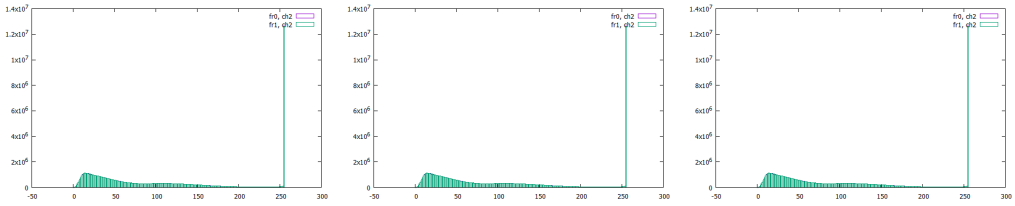

Figure 5.9: The accumulated channelwise histogram of both frames of the testing data set of the KITTI 2015 Benchmark. **Left:** First channel. **Middle:** Second channel. **Right:** Third channel.

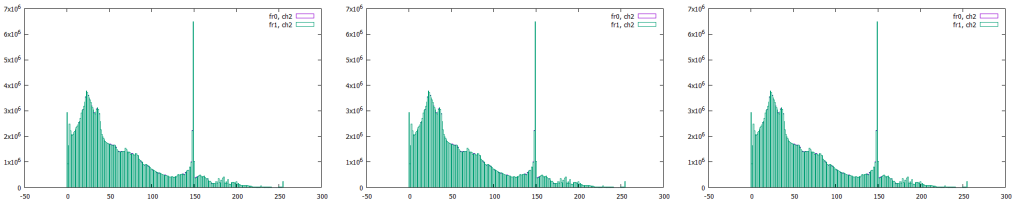

Figure 5.10: The accumulated channelwise histogram of both frames of the testing data set of the MPI-Sintel Benchmark. **Left:** First channel. **Middle:** Second channel. **Right:** Third channel.

5.2.2 *Optical Flow Statistics*

Regarding the optical flow statistics, both magnitude and angle values are computed again for the testing data sets. The minimal magnitudes are considered to be rather similar for all benchmarks. Like for the training data sets, the Middlebury has the smallest average maximal disparity which makes the computation of optical flow for scenes from it easier. The MPI-Sintel Benchmark offers quite large displacements while the two KITTI benchmarks have values of more than 100 pixels again. These values depend on the computed flow for these scenes of course but give an estimation of what to expect for the testing data sets. Considering the magnitude histogram small optical flow vectors dominate again but the KITTI 2012 Benchmark has the highest amount of large displacements in comparison to the other benchmarks again which makes it more difficult to compute the corresponding optical flow.

Considering the angle distributions of the benchmarks, shown in figure 5.12, some differences to the training data sets are observable. The negative x-direction is more prominent than the positive one while still some small fluctuations appear in the Middlebury Benchmark. However, the distribution of angles in the KITTI 2012 Benchmark doesn't seem to be changed whereas the main directions of the KITTI 2015 Benchmark are now along all four axis directions. Most vectors in the MPI-Sintel Benchmark still point in the negative x-direction while the negative y-direction yields the second most

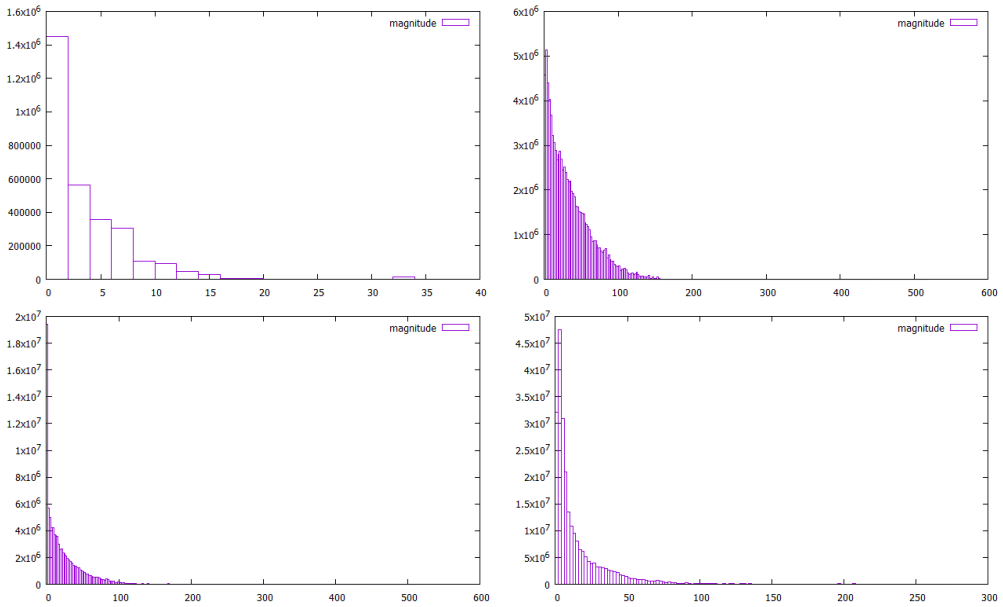

Figure 5.11: The accumulated magnitude histogram of the optical flow of the testing data sets. **Upper left:** Middlebury. **Upper right:** KITTI 2012. **Lower left:** KITTI 2015. **Lower right:** MPI-Sintel.

angles now. Additionally, the other angles are quite well distributed with some small fluctuations in this benchmark.

5.2.3 *Illumination*

Considering the average differences of the two frames, shown in table 5.21, the values don't differ much from the ones of the training data sets which means on the one hand that the data is consistent and on the other hand that small brightness differences appear between the two frames.

Table 5.20: Magnitude statistics for the testing data sets.

Magnitude Values	Minimum	Maximum	Average	Stand. Dev.
Middlebury	0.0549	15.2276	3.4674	2.7996
KITTI 2012	1.5865	136.13	33.8295	24.9982
KITTI 2015	0.0704	124.809	22.9067	20.6586
MPI-Sintel	1.6405	51.3662	15.6861	9.3344

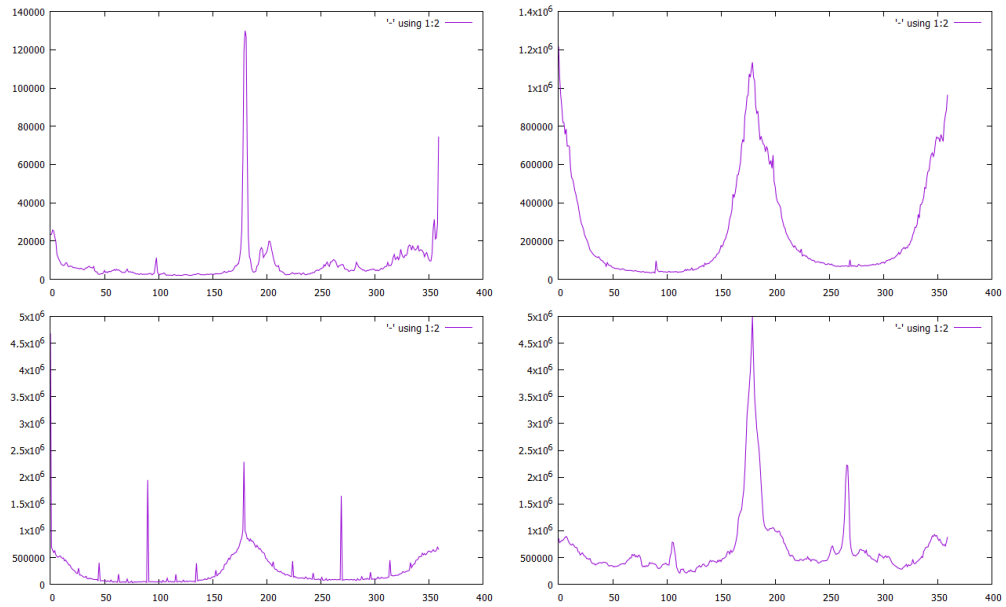

Figure 5.12: The accumulated angle graph of the optical flow of the testing data sets. **Upper left:** Middlebury. **Upper right:** KITTI 2012. **Lower left:** KITTI 2015. **Lower right:** MPI-Sintel.

By having a look at table 5.22, the multiplicative component is around 1 again which means that the brightness is nearly not changed by this component but it only darkened the image a little. Since the images have more small brightness values than high ones, the influence of the multiplicative component is small as well. Just like for the training data, the additive components' values of the non-penalised energy function are a bit higher than 0 while the values of the penalised energy function are quite close to it. This means that outliers are the cause for the higher values of the additive components' values of the non-penalised energy function and for the observed brightness

Table 5.21: Difference statistics for the testing data sets.

Difference Values	Average	Stand. Dev.
Middlebury	6.5808	11.0313
KITTI 2012	8.4249	15.658
KITTI 2015	10.9764	18.9017
MPI-Sintel	6.8425	13.942

difference. Overall, it can be said again that a global illumination change doesn't take place and the brightness is globally preserved for all four benchmarks.

Since the components of the global illumination change suggest that the brightness is preserved, the values for the components of the local illumination change are considered to verify this assumption. The range of values of the multiplicative component is not as high as for the training data sets but there are still some high extrema values. Especially the maxima for the KITTI 2015 and the MPI-Sintel benchmarks are very high. The same holds for the additive components' values. Nevertheless, the average values confirm the assumption of global brightness preservation and only some local illumination changes. The MPI-Sintel Benchmark can be considered as one of the more difficult benchmarks since it has the highest range of values.

Table 5.22: Global illumination change components for the testing data sets. The components that have been computed using a robust energy function on which a subquadratic penaliser has been applied are listed in the last two columns.

Comp. Values	Mult. comp.	Add. comp.	Mult. comp. (rob.)	Add. comp. (rob.)
Middlebury	0.9667	3.2558	0.9985	0.3892
KITTI 2012	0.9714	2.2297	1.0004	−0.3545
KITTI 2015	0.9489	3.7309	0.9992	−0.49
MPI-Sintel	0.9219	5.0276	0.9922	0.2904

Table 5.23: Statistics for the multiplicative component of local illumination changes for the testing data sets. The second row of each benchmark contains the values for the robust energy function.

Comp. Values	Minimum	Maximum	Average	Stand. Dev.
Middlebury	−2.4226	16.462	0.9974	0.5261
Middlebury (rob.)	−0.0169	20.849	0.9979	0.5372
KITTI 2012	0.0026	10.6812	0.8918	0.5361
KITTI 2012 (rob.)	0.0019	10.6228	0.8704	0.5481
KITTI 2015	−3.79938	44.4765	0.9559	0.6963
KITTI 2015 (rob.)	−0.0267	50.855	0.9412	0.7133
MPI-Sintel	−9.7835	33.046	0.9895	0.7209
MPI-Sintel (rob.)	−1.2059	52.5513	1.0254	0.8264

Table 5.24: Statistics for the additive component of local illumination changes for the testing data sets. The second row of each benchmark contains the values for the robust energy function.

Comp. Values	Minimum	Maximum	Average	Stand. Dev.
Middlebury	−0.1995	30.2288	0.0747	0.4551
Middlebury (rob.)	0.1738	26.2091	0.0221	0.1712
KITTI 2012	−0.0926	0.8484	0.0066	0.0119
KITTI 2012 (rob.)	−0.0124	0.67	0.0188	0.0212
KITTI 2015	−0.3964	110.149	0.0981	0.9049
KITTI 2015 (rob.)	−0.2431	71.3163	0.0304	0.2402
MPI-Sintel	−0.7294	173.029	0.9704	4.8501
MPI-Sintel (rob.)	−0.7917	168.386	0.1155	1.7202

5.2.4 *Movement*

The values of the components of the global movement assumption are listed both for the constant movement model and the affine movement model in the tables 5.25 and 5.26. Considering the constant movement model first, the values of the constant parameters are rather small for all benchmarks. Only the KITTI 2015 and the MPI-Sintel benchmarks have slightly higher values for the non-penalised energy function while the KITTI 2012 and the MPI-Sintel benchmarks have a bit higher values for the penalised energy function. It is important to mention that these values depend on the computed flow and hence the result is affected by the assumptions which were used to compute this optical flow. When assuming global affine movement, the multiplicative components for the x- and y-values of u and v are all very close to 0 again which means that the constant model suffices like for the training data sets to describe the

Table 5.25: Average value of the movement components for the testing data sets, assuming global constant movement.

Comp. Values	u_{const}	v_{const}	u_{const} (rob.)	v_{const} (rob.)
Middlebury	0.1252	0.5698	−0.0644	0.4405
KITTI 2012	−0.2225	0.6602	1.2474	3.6899
KITTI 2015	1.4689	3.3834	−0.8572	0.6756
MPI-Sintel	−2.6454	0.016	−2.6371	−0.1548

global movement, although even the values of the constant movement are quite small. The Middlebury has overall the smallest component values which is why it contains less global movement in comparison with the other benchmarks.

Just like for the training data sets, the amount of points with a small gradient or Hessian are regarded when categorising the type of local movement. The threshold T for small gradients and Hessians is again defined to be half of the median of all gradient respectively Hessian values. Assuming local constant movement first, around

Table 5.26: Average value of the movement components for the testing data sets, assuming global affine movement.

Comp. Values	$u_{x-\text{dir}}$	$u_{y-\text{dir}}$	u_{const}	$v_{x-\text{dir}}$	$v_{y-\text{dir}}$	v_{const}
Middlebury	0.0003	−0.0088	0.121	−0.0026	0.0023	0.5697
Middlebury (rob.)	−0.0013	−0.0081	0.182	−0.002	0.0016	0.5799
KITTI 2012	0.0914	0.0086	2.4308	0.001	0.0824	3.7203
KITTI 2012 (rob.)	0.0849	0.0035	1.729	0.0006	0.0801	2.9475
KITTI 2015	0.0583	−0.0022	1.4973	0.0004	0.0552	3.3587
KITTI 2015 (rob.)	0.0513	−0.0035	1.1234	−0.0004	0.0533	2.7617
MPI-Sintel	0.0124	0.0003	−2.4191	0.0035	0.0117	0.2419
MPI-Sintel (rob.)	0.0121	0.0003	−2.3993	0.0029	0.0093	−0.067

Table 5.27: Average number of all points with a small gradient or Hessian for the testing data sets given in percent, assuming local constant movement. A small gradient or Hessian depends on the threshold T which is half the median of all gradient respectively Hessian values.

Average percentage	$\lvert\nabla u\rvert^2 \leq T_{\nabla u}$	$\lvert\nabla v\rvert^2 \leq T_{\nabla v}$	$\lVert \mathrm{H}(u)\rVert_F^2 \leq T_{\mathrm{H}(u)}$	$\lVert \mathrm{H}(v)\rVert_F^2 \leq T_{\mathrm{H}(v)}$
Middlebury	37.3708	37.4409	35.8435	35.1193
Middlebury (rob.)	38.4361	37.4028	36.6263	35.1572
KITTI 2012	42.9435	42.8275	42.393	42.287
KITTI 2012 (rob.)	44.1986	45.2324	45.5321	43.932
KITTI 2015	44.2089	43.8902	43.5631	43.2313
KITTI 2015 (rob.)	45.5306	43.9928	46.4812	42.3635
MPI-Sintel	39.5618	38.4778	38.5484	37.9237
MPI-Sintel (rob.)	41.5247	39.3009	41.304	39.1411

$38 - 45\%$ of all pixels have a small gradient in both directions. The same holds for the amount of points with a small Hessian which is why it can be assumed that these points really have constant movement. However, this can be explained by the choice of the computed flow since a first order computed flow is used which assumed smooth gradients. Therefore, the computation of points with affine movement suggests that all points have constant movement as well. Hence, this metric can only determine the average percentage of points with a constant movement for this computed flow which is around $37 - 44\%$.

The amount of pixels with a large displacement relative to the maximal displacement in percent is shown in table 5.29. About $4.5 - 7\%$ of all points for all benchmarks have

Table 5.28: Average number of all points with a small gradient or Hessian for the training data sets given in percent, assuming local affine movement. A small gradient or Hessian depends on the threshold T which is half the median of all gradient respectively Hessian values.

Average percentage	$\|\nabla u\|^2 \leq T_{\nabla u}$	$\|\nabla v\|^2 \leq T_{\nabla v}$	$\|\mathrm{H}(u)\|_F^2 \leq T_{\mathrm{H}(u)}$	$\|\mathrm{H}(v)\|_F^2 \leq T_{\mathrm{H}(v)}$
Middlebury	37.5441	37.5695	36.3732	35.5168
Middlebury (rob.)	37.7697	37.8089	38.1129	37.2098
KITTI 2012	42.929	42.7643	42.2449	42.0749
KITTI 2012 (rob.)	43.0005	42.8521	43.4916	43.606
KITTI 2015	44.2915	43.9248	43.7576	43.3135
KITTI 2015 (rob.)	43.9055	43.4094	43.8306	43.6573
MPI-Sintel	39.7242	38.6058	39.2864	38.2868
MPI-Sintel (rob.)	39.8096	38.6438	41.1455	40.1924

Table 5.29: Average number of all points with a large displacement for the testing data sets given in percent. A large displacement is considered if the magnitude of the flow vector is equal or larger than a factor times the maximal displacement in the image sequence.

Average percentage	Factor 0.5	Factor 0.8	Factor 0.9
Middlebury	5.7231	0.9113	0.4776
KITTI 2012	6.6806	1.8672	0.8074
KITTI 2015	4.6008	1.071	0.3727
MPI-Sintel	6.9782	1.7363	0.638

a displacement magnitude of half of the maximal magnitude while the optical flow vector of about $0.5 - 1\%$ of all pixels has a magnitude of 0.9 times the maximal length. These values are quite similar to the ones for the training data sets. Due to the fact that the maximal magnitude of the optical flow vectors of the Middlebury Benchmark are again short in comparison with the other benchmarks, the difficulty of the other benchmarks is higher.

5.2.5 *Fundamental Matrix Estimation*

The results of the Fundamental Matrix estimation can be seen in table 5.30 where the average distances of the correspondence points to their epipolar lines are presented. Like in the case for the training data set, the Middlebury Benchmark has the smallest values due to the fact of containing mainly camera motion which causes the optical flow. This eases the computation of optical flow again because assumptions for stereo matching algorithms can help. Since the MPI-Sintel Benchmark contains many scenes both with camera motion and independently moving objects, the average distance is still smaller than or around 1 pixel. Even though the KITTI benchmarks consist of basically only stereo images due to the fact that the camera was moving during the image capturing process, their values are a bit higher. This can be explained again with the problem of points close to the epipoles. Another reason is that computed flow is used and these distances are only estimations.

Table 5.30: Average distance of a point to its epipolar line for the testing data sets given in pixels.

Average distance	Total Least Squares Fit	M-Estimators
Middlebury	0.50721	0.4475
KITTI 2012	0.93339	1.40807
KITTI 2015	1.11368	1.82801
MPI-Sintel	0.84305	1.16856

6. CONCLUSION AND FUTURE WORK

In the following chapter, the proposed metrics and their results regarding the four benchmarks are summarised and in the end a few ideas for future work are given.

6.1 CONCLUSION

This thesis proposed some metrics to investigate the difficulties and regularities of optical flow benchmarks. These metrics can be assigned to the categories image statistics, optical flow statistics, illumination, movement and stereo. The statistics regarding the images themselves are not useful enough to determine the difficulty since most images use the full range of pixel values. However, the optical flow statistics are quite helpful because the maximal magnitude as well as the main flow directions can be determined. The histograms of these values can give additional information on the amount of large flow vectors and how well the flow directions are distributed. This way it is possible to say that the two KITTI benchmarks have the largest displacements, especially the KITTI 2012 Benchmark has the most large flow vectors, while the Middlebury has only very short displacements. This can be confirmed by computing the amount of points with large displacements relative to the maximal displacement. Nevertheless, the Middlebury and the MPI-Sintel benchmarks' directions are better distributed whereas the angles of the flow vectors of the two KITTI benchmarks are more predictable.

Considering the illumination metrics, the brightness is almost preserved for all benchmarks, especially regarding the global illumination changes. But in the consideration of local illumination changes, the MPI-Sintel Benchmark has the largest range of component values which makes it hard to integrate this into the computation of optical flow while the KITTI 2012 Benchmark has nearly no local illumination changes.

By evaluating the movement metrics, it can be assumed that the two KITTI benchmarks have a small amount of global constant movement, especially the KITTI 2012 Benchmark whereas the MPI-Sintel and the Middlebury benchmarks are nearly free of any global movement. Global movement eases the effort to compute optical flow. However, about $30 - 40\%$ of all points in all four benchmarks have local constant movement but only the MPI-Sintel Benchmark seems to have points with affine movement.

The last metric is about stereo which is why a Fundamental Matrix has been estimated from the scene and the average distance of points to their epipolar lines is computed. The Middlebury Benchmark almost only consists of stereo scenes while the other benchmarks contain stereo scenes as well but they still have some independently

moving objects. These observations hold true both for the training and the testing data sets.

6.2 FUTURE WORK

There are some possibilities to extend the evaluation of optical flow benchmarks which has been done in this thesis. One of them could be to evaluate other than the four benchmarks and generate a general overview and ranking of the difficulties of optical flow benchmarks. Additionally, it might be possible to design new benchmarks on criteria which increase the measured difficulty, depending on which metrics are considered more important than others. Furthermore, the suggested metrics can be extended in many ways. Some approaches are suggested in the following sections.

6.2.1 *Image Statistics*

The first metrics category was about image statistics. In this thesis, the focus was only on the image information itself which means the pixel values. The information which was not considered here was the relation of the pixels and their values to their neighbours. This means that the pixel values should be evaluated in a bigger context. To do this, it could be possible to extract information from the image about structures. This could be done by transforming the image into the Fourier domain which reveals image structures. Additionally, the Fourier spectrum can be computed from it. Image structures can either help or be difficult to compute optical flow from. Another possibility which gives a clue about image structures are the derivatives of the image. Images with objects that have a high contrast and can be clearly separated from each other can be expressed as having many high derivatives whereas many low derivatives mean that objects can't be distinguished very well due to the fact that the boundaries are rather smooth. Segments are somehow related to this topic. By computing a segmentation of an image, it might be possible to give a clue how many objects appear in the image and how many smooth or flat regions exist. With the help of these metrics, the difficulty of the images themselves can be measured.

6.2.2 *Illumination*

Considering illumination changes, only the affine model has been used in this thesis. It might be insteresting to assume other brightness transfer functions and compute their weights. Another possibility would include the learning of brightness transfer functions first and using them to measure the difficulty of the given benchmark. Illumination effects like specular reflections and shadows and shading were mentioned before which increase the difficulty to compute optical flow. Therefore, it might also

be interesting to apply such methods on the evaluation process of optical flow benchmarks.

6.2.3 *Movement*

Just like for the measurement of illumination changes, different movement types might be assumed and therefore other models which represent the considered movement can be applied. The possibility to learn movement models exists as well as for illumination change models. The large displacements were measured by counting pixels with a movement larger or equal to the maximal displacement multiplied by a certain factor. A common difficulty of the computation of optical flow fields are small objects with a large displacement since they are hard to track. Therefore, the measurement of large displacements should include the size of the considered object. To do this, the image should be transformed into a Gaussian scale-space using difference of Gaussians. This way, certain objects of a specific size exist only in some levels and can be localised better. Small objects can be found by doing so and their displacement relative to their size can be measured.

6.2.4 *Fundamental Matrix Estimation*

The Fundamental Matrix was estimated either by a total least squares fit or by M-estimators. The quality of these methods can be increased by normalising the x- and y-values such that large values don't have a much bigger impact on the result as small values. However, there exist other methods to estimate the Fundamental Matrix like using a geometrical approach. A more robust algorithm is RANSAC which estimates several Fundamental Matrices by choosing various subsets of correspondences and then evaluates the computed matrices with the help of the remaining correspondences.

6.2.5 *Computed Flow*

The difficulty of the testing data sets of the benchmarks can only be estimated since the ground truth is withheld. Therefore, optical flow which has been computed from the image scenes is used to investigate the benchmarks' difficulties. The methods to compute optical flow integrate some assumptions which have an influence on the resulting flow field. Hence, it might be necessary to use optical flow from different methods to get a better estimation of the difficulties.

BIBLIOGRAPHY

[1] S. Baker, D. Scharstein, J. Lewis, S. Roth, M. J. Black, and R. Szeliski, "A Database and Evaluation Methodology for Optical Flow," *International Journal of Computer Vision*, vol. 92, no. 1, pp. 1–31, 2011.

[2] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, IEEE, 2012, pp. 3354–3361.

[3] M. Menze and A. Geiger, "Object Scene Flow for Autonomous Vehicles," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3061–3070.

[4] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets Robotics: The KITTI Dataset," *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1231–1237, 2013.

[5] D. J. Butler, J. Wulff, G. B. Stanley, and M. J. Black, "A Naturalistic Open Source Movie for Optical Flow Evaluation," in *European Conference on Computer Vision*, Springer, 2012, pp. 611–625.

[6] J. L. Barron, D. J. Fleet, and S. S. Beauchemin, "Performance of Optical Flow Techniques," *International journal of computer vision*, vol. 12, no. 1, pp. 43–77, 1994.

[7] M. Otte and H.-H. Nagel, "Optical Flow Estimation: Advances and Comparisons," in *European Conference on Computer Vision*, Springer, 1994, pp. 49–60.

[8] B. McCane, K. Novins, D. Crannitch, and B. Galvin, "On Benchmarking Optical Flow," *Computer Vision and Image Understanding*, vol. 84, no. 1, pp. 126–143, 2001.

[9] D. Butler, J. Wulff, G. B. Stanley, and M. J. Black, "MPI-Sintel Optical Flow Benchmark: Supplemental Material," in *MPI-IS-TR-006, MPI for Intelligent Systems (2012*, Citeseer, 2012.

[10] M. D. Grossberg and S. K. Nayar, "Modeling the Space of Camera Response Functions," *IEEE transactions on pattern analysis and machine intelligence*, vol. 26, no. 10, pp. 1272–1282, 2004.

[11] O. Demetz, M. Stoll, S. Volz, J. Weickert, and A. Bruhn, "Learning Brightness Transfer Functions for the Joint Recovery of Illumination Changes and Optical Flow," in *European Conference on Computer Vision*, Springer, 2014, pp. 455–471.

[12] T. Nir, A. M. Bruckstein, and R. Kimmel, "Over-Parameterized Variational Optical Flow," *International Journal of Computer Vision*, vol. 76, no. 2, pp. 205–216, 2008.

[13] D. Scharstein and R. Szeliski, "High-Accuracy Stereo Depth Maps Using Structured Light," in *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, IEEE, vol. 1, 2003, pp. I–I.

[14] D. J. Fleet and A. D. Jepson, "Computation of Component Image Velocity from Local Phase Information," *International journal of computer vision*, vol. 5, no. 1, pp. 77–104, 1990.

[15] D. Scharstein and R. Szeliski, "A Taxonomy and Evaluation of Dense Two-Frame Stereo Correspondence Algorithms," *International journal of computer vision*, vol. 47, no. 1-3, pp. 7–42, 2002.

[16] S. M. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski, "A Comparison and Evaluation of Multi-View Stereo Reconstruction Algorithms," in *Computer vision and pattern recognition, 2006 IEEE Computer Society Conference on*, IEEE, vol. 1, 2006, pp. 519–528.

[17] L. Priese, *Einführung in die Verarbeitung und Analyse digitaler Bilder*. Springer, 2015.

[18] S. Negahdaripour and C.-H. Yu, "A Generalized Brightness Change Model for Computing Optical Flow," in *Computer Vision, 1993. Proceedings., Fourth International Conference on*, IEEE, 1993, pp. 2–11.

[19] O. Faugeras, Q.-T. Luong, and T. Papadopoulo, *The Geometry of Multiple Images: The Laws That Govern the Formation of Multiple Images of a Scene and Some of Their Applications*. MIT press, 2004.

[20] Q.-T. Luong and O. D. Faugeras, "The Fundamental matrix: Theory, algorithms, and stability analysis," *International journal of computer vision*, vol. 17, no. 1, pp. 43–75, 1996.

[21] H. Zimmer, A. Bruhn, J. Weickert, L. Valgaerts, A. Salgado, B. Rosenhahn, and H.-P. Seidel, "Complementary Optic Flow," in *International Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition*, Springer, 2009, pp. 207–220.

DECLARATION

Ich versichere, diese Arbeit selbstständig verfasst zu haben. Ich habe keine anderen als die angegebenen Quellen benutzt und alle wörtlich oder sinngemäß aus anderen Werken übernommene Aussagen als solche gekennzeichnet. Weder diese Arbeit noch wesentliche Teile daraus waren bisher Gegenstand eines anderen Prüfungsverfahrens. Ich habe diese Arbeit bisher weder teilweise noch vollständig veröffentlicht. Das elektronische Exemplar stimmt mit allen eingereichten Exemplaren überein.

I hereby declare that the work presented in this thesis is entirely my own. I did not use any other sources and references that 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