UNIVERSITÄT PADERBORN
Die Universität der Informationsgesellschaft

FAKULTÄT FÜR
ELEKTROTECHNIK,
INFORMATIK UND
MATHEMATIK

Robust Motion Estimation for Qualitative Dynamic Scene Analysis

Von der Fakultät für Elektrotechnik, Informatik und Mathematik
der Universität Paderborn

zur Erlangung des akademischen Grades

Doktor der Ingenieurwissenschaften (Dr.-Ing.))

genehmigte Dissertation

von

MSc-EE. Mahmoud Ali Ahmed Mohamed

Erster Gutachter: Prof. Dr.-Ing. Bärbel Mertsching
Zweiter Gutachter: Prof. Dr. Domenec Puig

Tag der mündlichen Prüfung: 22.07.2019

Paderborn 2019

Diss. EIM-E/348

Dedication

Dedicated to my wife Rania and my kids, Youssef, Adam and Jana who experienced through all the hardships during the time of this research persistently.

Declaration

I hereby declare that I have completed the work on this PhD dissertation with my own efforts and no part of this work or documentation has been copied from any other source. It is also assured that this work is not submitted to any other institution for award of any degree or certificate.

Paderborn, July 23, 2019

Mahmoud Mohamed

UNIVERSITÄT PADERBORN
Die Universität der Informationsgesellschaft

FAKULTÄT FÜR
ELEKTROTECHNIK,
INFORMATIK UND
MATHEMATIK

Zusammenfassung der Dissertation

Robust Motion Estimation for Qualitative Dynamic Scene Analysis

Des Herrn Mahmoud Ali Ahmed Mohamed

Die Wahrnehmung und Analyse dynamischer Umgebungen ist eine zentrale Herausforderung im Bereich kognitiver Anwendungen wie Fahrerassistenzsystemen und allen Arten von autonomen Roboteroperationen. Autonome Roboter sind in der Lage, vorgegebene Aufgaben ohne kontinuierliche Kontrolle durch den Menschen durchzuführen. Voraussetzung dazu ist unter anderem eine robuste Erkennung und Verfolgung von sich bewegenden Objekten. Speziell mobilen Robotern bereiten bewegte Objekte größere Schwierigkeiten bei der Lokalisierung und Navigation als stationäre Objekte. Mobile Rettungsroboter beispielsweise steigern ihre Leistung deutlich durch die Erkennung von sich bewegenden Opfern. Die robuste Erkennung / Verfolgung von sich bewegenden Objekten von einer sich bewegenden Kamera in einer Umgebung im Freien ist aufgrund dynamischer wechselnder, unordentlicher Hintergründe, variierender Beleuchtungsbedingungen, teilweiser Okklusion von Objekten und unterschiedlichen Blickwinkeln der Objekte eine Herausforderung.

Diese Doktorarbeit beschäftigt sich mit einer robusten 2D-Bewegungsschätzung (Optische Fluss) und die Analyse für dynamische Umgebungen basierend auf Bildsequenzen und umfasst die oben genannten Probleme. Zu diesem Zweck wurde ein Verfahren entwickelt, dass die Coarse-To-Fine Ansatz verbessert einsetzt, um 2D-Bewegungen sowohl von schnellen als auch von langsamen Objekten mit weniger Rechenleistung zu schätzen. Des Weiteren wird in der vorliegenden Arbeit ein neues Optimierungsmodell für die optische Flussschätzung basierend auf der Texturbeschränkung vorgeschlagen. Bei der Texturbeschränkung wird davon ausgegangen, dass Objekttexturen wie Kanten, Gradienten oder Ausrichtung merkmale bei Objekten oder Kamerabewegungen konstant bleiben. Das Optimierungsmodell verwendet eine Zielfunktion, um die Unähnlichkeit zwischen der Bildtextur unter Verwendung lokaler Deskriptoren zu minimieren. Das vorgeschlagene Modell ist nicht auf besondere lokale Texturdeskriptoren beschränkt. Darüber hinaus stellen wir die Verwendung der Monokular Epipolaren Linienbeschränkung vor, um die Genauigkeit des geschätzten optischen Flusses in texturlosen Regionen zu verbessern. Das neue Modell schätzt den optischen Fluss in den meisten Fällen korrekt, wenn die meisten Ansätze nach dem Stand der Technik, die von der Helligkeitskonstanz eines Pixels abhängen, ausfallen. Außerdem, schlagen wir einen neuen Ansatz vor, um alle sich bewegenden Objekte zu erkennen und zu verfolgen. Der neue Algorithmus funktioniert sowohl mit einer statischen als auch mit einer sich bewegenden Kamera, und die Ergebnisse zeigen die erfolgreiche Erkennung und Verfolgung von sich bewegenden Objekten in Innen- und Außenumgebungen. Verschiedene Experimente und Anwendungen wurden durchgeführt, um die Algorithmen ausführlich zu testen und auszuwerten. Die Ergebnisse haben gezeigt, dass die vorgeschlagenen Algorithmen auf der Grundlage der Standard-Testdatensätzen den Stand der Technik übertroffen haben.

UNIVERSITÄT PADERBORN
Die Universität der Informationsgesellschaft

FAKULTÄT FÜR
ELEKTROTECHNIK,
INFORMATIK UND
MATHEMATIK

Abstract

Robust Motion Estimation for Qualitative Dynamic Scene Analysis
Mr. Mahmoud Ali Ahmed Mohamed

Dynamic scene analysis is the primary challenge for various applications such as Advanced Driver Assistance Systems (ADAS), and in any autonomous robot operation in dynamic environments. Autonomous robot/vehicle can carry out desired tasks without continuous human interaction. Distinctly, robust detection, tracking, and recognition of moving objects as well as an estimation of camera ego-motion in a scene are necessary expendables for many autonomous tasks. For instance, in mobile robotics, moving objects are possibly more insecure than stationary objects for safe navigation. In particular, rescue robot systems could increase their performance enormously if they were capable of interacting with moving victims. Robust detection/tracking of moving objects from a moving camera in an outdoor environment is a challenging task due to dynamically changing cluttered backgrounds,large motion, varying lighting conditions, less texture objects, partial object occlusion, and varying object viewpoints.

The work presented in this thesis cops with the problem of robust estimation of 2D motion and tracking of moving objects with the problems mentioned above. Therefore, this work introduce a new approach to improve the accuracy of the 2D motion estimation, which called optical flow, in case of large motion using the coarse-to-fine technique. The proposed algorithm estimates the optical flow of fast as well as slow objects correctly and with less processing cost. Moreover, the presented work proposes a novel optimization model for the optical flow estimation base on the texture constraint. The texture constraint assumes that object textures such as edges, gradients, or orientation-of-image features remain constant in case of objects or camera motion. The optimization model uses an objective function to minimize dissimilarity between image texture using local descriptors. The proposed model is not limited to any local texture descriptors, for instance, the histogram of oriented gradient (HOG), the modified local directional pattern (MLDP), the census transform, and other descriptors are used. Furthermore, we present the usage of the monocular epipolar line constraint to improve the accuracy of the optical flow in the case of texture-less regions. The new model estimates the optical flow correctly in most cases when most state-of-the-art approaches that depend on the brightness constancy of a pixel fail. Besides, we propose a new approach for detecting and tracking all moving objects. The proposed algorithm works with a static as well as a moving camera, and the results show the successful detection, estimation, and tracking of moving objects in indoor and outdoor environments. Several experiments and applications have been conducted to test and evaluate the algorithms extensively. The results have shown that the proposed algorithms outperformed the state-of-the-art approaches based on the standard benchmark datasets.

Acknowledgements

First of all, praise and thanks be to Allah who empowered me to achieve this level of academic accomplishment. Secondly, I am appreciative of my parents whose ethical support and petitions influenced it to come up to this point. I would like to thank my precious wife and kids for their proceeded with persistence and support. They are continually ready to tune in to my issues and support me in all of my undertakings. They help me cope during hard times, and keep me grounded in great circumstances.

I am extremely thankful to my supervisor Professor Bärbel Mertsching whose direction and support in every accommodating and all helpful aspects during the time of my PhD kept things progressing. Her words of encouragements and guidance have helped me through many obstacles during my thesis.

I would like to express my deepest appreciation to all my colleges in the GET Lab who provided me the possibility to complete this work.

Contents

1 Introduction

Motion perception is an essential task in our daily life. We humans perceive, understand and interact with the surrounding environment using the regular feedback provided by our visual system. We continuously perceive the motion of the environment and locate our position and those of other objects in the environment. We know the directions of cars and estimate their velocity in order to avoid a collision. Even in adverse conditions such as severe weather conditions, occlusion, illumination change and noise, we can recognize the boundaries and shapes of moving objects effortlessly.

Due to the importance of motion perception for humans, motion analysis for vision systems is considered one of the essential topics in the computer/machine vision field. Anywise, it is a nontrivial problem for computers to understand motion as humans do. For machines, the motion is ambiguous from only a local analysis due to the aperture problem. Therefore, a spatial regularization has been considered, namely neighboring pixels are likely to move together to reduce the ambiguity. Nevertheless, it is not clear to which extend pixels in a neighborhood should move together merely from local information.

1.1 Active Vision System

Recently, the development of fully or semi-autonomous robots has attracted the attention of many researchers. These robots operate independently in structured and unstructured dynamic environments without continuous human guidance. Therefore, to achieve such a purpose, a robot should have the ability to analyze, understand, and interact with these environments. Hence, a robot can use various sensors to interact with its environment, including video cameras, sonars, radar,

or thermal cameras. Among all of these sensors, video cameras are more suitable for long-term remote sensing, least hardware cost, space, and energy, delivering rich appearance information and most spatial-temporal resolution. Even humans receive most information through vision. Accordingly, a vision based motion analysis system is considered to have the highest potential to fill for the need of dynamic scene analysis over an extensive variety of applications. Subsequently, mounting an active vision system on a robot provides ongoing inputs of the present movement conditions which enables them to interact with a quickly changing dynamic condition.

The use of active vision systems enables mobile robots to analyze camera images and extract scene information, such as objects, traffic/hazard signs, roads/paths, scene model reconstruction, and motion. Correspondingly, the extracted information can be processed using higher level analysis algorithms, such as scene understanding which gives the ability to interact with different objects and do path planning. For these reasons, intelligent robots with such cognitive skills help to navigate, explore, and interact with other objects in a dynamic environment.

Admittedly, extracting reliable information from a machine vision system is a challenging task for different reasons. On one side, there are some problems related to image acquisition process, noise, and camera resolution. On the other side, there are still many reasons why it is difficult to recognize objects (vehicles and pedestrians,..., etc) and objects (roads, buildings, trees,..., etc) from image sequences, for instance, change from the object viewpoint (i.e., rotation, scale, translation). Furthermore, objects of the same semantic class have various appearances. Moreover, objects such as humans look entirely different even if they are seen from the same viewpoint due to the deformability and possess changing.

Motion estimation is a necessary task in several vision applications such as depth estimation, object detection and tracking, estimation of camera ego-motion, localization, and time-to-collision with other objects. Hence, it allows the detection of moving objects and avoid obstacles. Motion estimation makes it particularly useful in the context of fully autonomous navigation behavior for robot/vehicle.

Furthermore, motion patterns can be used in combination with machine learning approaches to allow the interpretation of human mimics and gestures.

Autonomous navigation of a vehicle through dynamic environments requires a robust motion analysis system. Henceforth, to achieve a specific goal, the dynamic models of the environment have to be maintained, and the existing information has to be updated. Motion estimation provides essential information about the dynamics of a scene. For instance, it is possible to detect and track moving objects which allows an autonomous vehicle to do motion planning in a highly dynamic environment. However, the process of acquiring and correlating images of visible light does not generally move down the point of estimating physical motion [SKS+12]. Hence, the apparent motion of the light and the physical motion in some cases are significantly disparate. Therefore, 2D motion estimation researches have to defeat several challenges concerning the motion analysis and the detection of multi-moving objects within the concepts of dynamic scene analysis.

1.2 Motion Estimation

An image sequence is a result of recording a scene for a given time using a camera. Assuming that an image frame is represented as $I : \Omega \subset \mathbb{R}^2$, the intensity value of an image's pixel $\mathrm{p} = (x, y)^T$ at time t can be considered as an intensity function $I(x, y, t)$. Due to a camera or objects move within the scene, motion estimation aims at establishing correspondence relations between positions in two consecutive frames. For every pixel p in a given frame the goal is to find the corresponding pixel in the next consecutive frame. A 2D motion vector is the projection of a 3D world motion vector onto the 2D image plane, and it defines the relative speed and direction of a pixel in the 2D image domain. Hence, an optical flow field is the relation specified as a 2D displacement vector pointing from each position in a frame to the corresponding new position in the consecutive frame. It composes of motion of objects and the camera motion itself (ego-motion of a camera), and it can be calculated based on changes in intensity values. Unfortunately, objects motion does not always yield changes in intensity values and change in intensity

values are not always due to object motion. Accordingly, although the depicted objects themselves remain stationary; intensity values may be changed due to other factors such as shadows, illumination changes, camera noise (e. g. bad illumination conditions), non-rigidity, deformability and reflections (e. g. moving light source).

1.3 Problem Formal Definition

Assume two consecutive frames $I(x, y, t)$ and $I(x + u, y + v, t + 1)$, for each pixel in frame $I(x, y, t)$ there is a corresponding pixel in the next consecutive frame $I(x + u, y + v, t + 1)$. Optical flow aims at finding the 2D displacement vector $\boldsymbol{w} = [u, v]^T$, where u and v are the optical flow components in the x and y direction respectively. Optical flow between two consecutive frames can be calculated using phase correlation [SOCM01, FZMB02], block-based matching [LZL94, NM02, ZM00, KRL10, JP13], discrete optimization [MHG15a, RWHS15b, WRHS13a] and differential techniques [BWF$^+$03, BBPW04, BM11, ZBW11]. Among all of these methods, differential techniques are the most successful approaches to calculate the optical flow [PUZ$^+$07], [BSL$^+$11] due to its accuracy and processing power.

Differential approaches use the brightness constancy assumption (BCA) by assuming a constant intensity value of a pixel if objects or a camera moves. This constraint constructs a residual function called a data term, but this is not sufficient to solve the optical flow unknowns u and v. Another constraint is needed which assumes a smoothness of the local or the global optical flow (smoothness or a regularization term). Hence, estimating the motion vector $\boldsymbol{w}$ can be achieved by optimizing an objective function combines a data and a smoothness term. The variational optical flow approach is a particularly appealing formulation of differential models [BM11]. It is based on the total variation (TV) regularization and the $L1$ or $L2$ norm in the data term. This formulation preserves discontinuities in the flow field and offers increased robustness against occlusions and noise.

Figure 1.1 shows an optical flow example for "backyard sequence" of Middlebury dataset [BSL$^+$11]. Here the estimated optical flow is represented using color mapping scheme from [BSL$^+$11]. In this visualization, the optical flow is visualized

(a) (b) (c)

Figure 1.1: Middlebury dataset [BSL+11]: (a) A blended overlay image of the "backyard sequence", scaling the intensities of two consecutive images jointly as a single image. (b) Ground truth of optical flow field. (c) Middlebury optical flow color mapping representation.

using the HSV color space in which each vector $[u, v]^T$ is decomposed into polar coordinates R, θ. The angle θ of each vector is considered as hue value, and the magnitude is considered to be the value component, while the saturation component is always equal to the maximum value. In this thesis, we follow the Middlebury visualization scheme to show the optical flow.

1.4 Challenging of Optical Flow Estimation

Estimating the optical flow using the differential approach based on the minimization of an objective function is a challenging task due to several aspects such as large displacements motion, illumination change, outliers, noise, and processing time [MG15]. The following paragraphs introduce some of these problems.

1.4.1 Large Displacement Motion

Large displacement motion caused by fast movements of objects or a camera which is a challenging problem. Hence the optical flow unknowns appear in the argument of the objective functions; the data-terms are highly non-convex

functions. Therefore, linearized data terms are used by most variational methods approximating the image sequence locally by using a linear function in space and time (i.e., using a Taylor expansion) on the cost of severe problems in cases with large displacements.

Multi-scale strategies such as [Ana89] and [ASW99] abstain from the linearization and perform gradient descent on the non-convex functions. Alternatively, a coarse-to-fine technique [BBPW04] estimates the optical flow at a coarse-levels where the motion is not large. Later, the coarse optical flow is propagated to the finer levels. The propagation of optical flow among different levels is normally done using an interpolation process and causes loss of motion details. Moreover, the bigger number of levels used for the coarse-to-fine scheme is the higher accuracy results, but at the cost of high processing power. Thus, the challenge here is to estimate the 2D motion of large objects without losing object fine detail information and inappropriate processing time. Figure 1.2 shows a large displacement sequence provided by KITTI dataset [GLSU13]. The absolute error between the ground truth and the estimated optical flow is represented by using the KITTI error color mapping for an error image. The KITTI color mapping scheme grades from a blue color for small errors through a red color for large errors. As shown in figure 1.2, a variational approach (i.e [BBPW04]) fails to estimate accurate optical flow in most of the scene. In turn, the proposed algorithm introduced in this thesis significantly increases the accuracy of the estimated optical flow.

1.4.2 Illumination Change

Illumination changes occur due to changes in the output of the real illumination source or due to a rotation of 3D surfaces or when the camera adjusts its exposure settings. Consequently, in these scenarios, the structures of the intensity-value flow over the image and their brightness change. Hence, finding corresponding point pairs is difficult. Nevertheless, most of the differential optical flow approaches use a residual function based on the brightness constraint. Accordingly, once the illumination changes or objects move to another place with a different illumination condition (i. e. into the shadow of a tree) this assumption is no longer valid. Figure 1.3 shows a KITTI [GLSU13] sequence which contains illumination changes

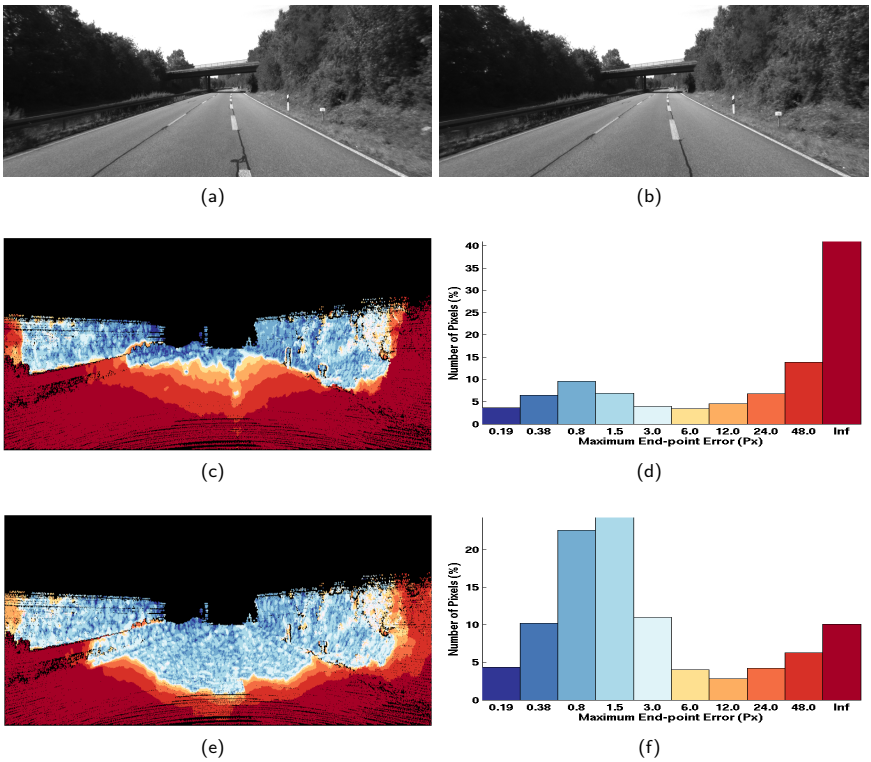


Figure 1.2: An example of large displacement optical flow from KITTI 2012 dataset sequence 181. a) and b) show frame 10 and frame 11. c) and d) show the absolute error image and the histogram of the absolute error image after applying the coarse-to-fine technique in [BBPW04]. e) and f) show the error image and the histogram of the absolute error image after applying the texture constraint proposed in this thesis.

between frame at time t and frame at time $t+1$. Here, the brightness constraint fails to reach an optimal solution in the regions which contain different illumination condition. The error image shows the absolute error after applying the algorithm [PUZ+07], which uses the brightness constraint in the data-term. Consequently, the average absolute error has increased dramatically and the percentage of outliers (points have an absolute error of more than 3 pixels) is grown to reach more than 70% using the state-of-the-art methods (i.e., [PUZ+07]) (see figure 1.3). Conversely, it is obvious that the accuracy has been significantly increased after applying the proposed algorithm in this thesis.

1.4.3 Real-Time Performance

Optical flow provides low-level information about a scene, which can be used by other image processing techniques to do high-level analysis. Unfortunately, the main shortcoming of the most of the state-of-the-art approaches for estimating optical flow is the processing time [GLSU13, BSL+11]. Consequently, most approaches consider the accuracy only and not much effort has invested in order to improve the performance. Figure 1.4 shows a histogram of processing time for all KITTI dataset method: in October 2018. As shown in figure 1.4, more than 87% of these methods need more than one second to estimate optical flow. Although there are great efforts to increase the performance using the new technology of the hardware, such as CPU, GPU, and FPGA, also faster optical algorithms gain great benefits of this technology.

1.5 Contributions of This Work

Figure 1.5 shows the proposed approach which contains two main sub-modules. The input of the first module is a sequence of images (minimum two frames), and the output is optical flow. For this purpose, we have developed a new framework for the estimation of optical flow that uses textures and epipolar constraints in a multi-level optimization for motion estimation in case of large displacements and illumination changes. The second module uses the estimated optical flow to detect

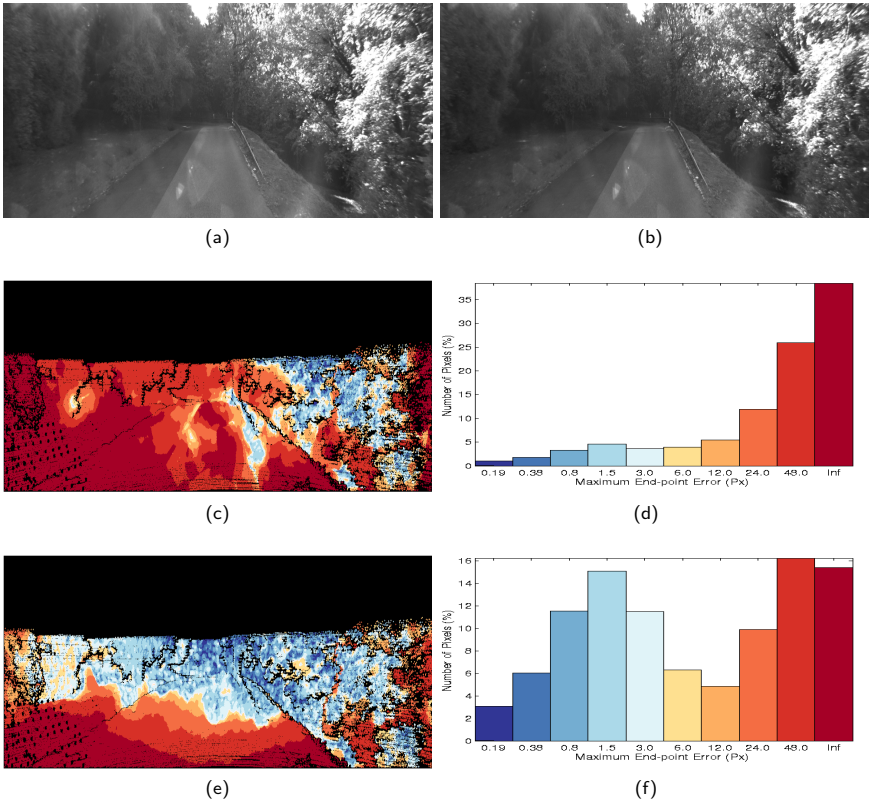


Figure 1.3: An example of illumination change from KITTI 2012 dataset sequence 74. a) and b) show frame 10 and frame 11. c) and d) show the absolute error and the histogram of the absolute error images after applying the brightness constraint using the [PUZ+07] algorithm. e) and f) show the absolute error and the histogram of the absolute error images after applying the texture constraint proposed in this thesis.

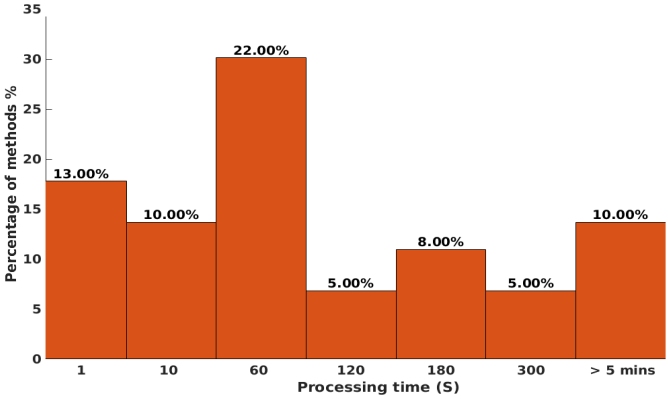


Figure 1.4: Histogram of the processing time of the optical flow methods on the KITTI 2012 [GLSU13] benchmark, October 2018.

and to track all moving objects in real time. Consequently, several algorithms have been proposed to build such approach. Moreover, we have tested the proposed algorithms and the complete approach in various outdoor and indoor environments. Furthermore, all proposed algorithms have been evaluated using the standard and well–known widely used datasets such as KITTI [GLSU13], Middlebury [BSL$^+$11], and MPI [BWSB12]. It can be concluded that the proposed algorithms provided state-of-the-art results and have higher ranks in all benchmarks. Besides, we have conducted several experiments and real application to test the proposed algorithms. The ultimate goal of this work is to use optical flow to analyze dynamic scenes. For this purpose, we have developed robust optical flow estimation approaches dealing with different kinds of environments and improved the performance of large displacement optical flow. Afterward, we propose an approach for detection and tracking all moving objects in real-time using only optical flow data. The following paragraphs explain in details each contribution:

- **Optical Flow Estimation**
  In this thesis, we proposed solutions to the challenging problems of optical flow estimation discussed above.

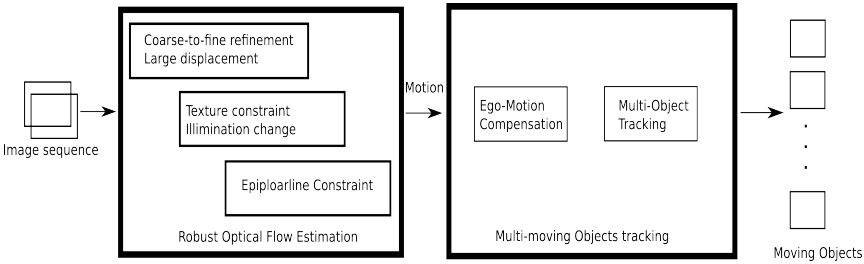


Figure 1.5: The flowchart of the proposed dynamic scene analysis approach.

– **Multi-scale optical flow optimization**
A new method for optimizing the estimation of large displacement optical flow is proposed. The proposed method estimates the optical flow using multi-scale processing without losing motion information for small objects. Primarily, the new algorithm uses the benefit of points corresponding at each level of the coarse-to-fine optimization to initialize the optical flow estimation to decrease the dependency on a large number of levels. On one side, the optimization algorithm uses the lower number of levels and lower processing time than the original optimization algorithm. On the other side, it saves the small details of small and fast objects which are affected by the linearization of the data term.

– **Robust and accurate optical flow estimation**
Concerning robust optical flow estimation for environments with different conditions and unlike most of the state-of-the-art optical flow methods, the proposed algorithms displace the brightness constraint by a new texture constraint. Alternatively, the texture constraint assumes that the textures of objects stay constant if the objects or the camera moves. Hence, we propose a novel descriptor called Modified Local Directional Pattern (MLDP), and we prove its power. The MLDP encodes texture of a region using the direction of gradient vectors in a binary pattern, which is invariant to illumination changes. Furthermore, we propose to use the Histogram of Oriented Gradient

(HOG) and the Average Distributed Gradient (DAG), which encode texture information using the direction and magnitude of the gradient vectors. Consequently, to use the texture constraint in optical flow framework, we introduce the necessary formulations to optimize an energy function based on a residual of two local descriptors. Ultimately, the proposed optimization framework is not restricted to a specific type of descriptors, and any other constraint can be easily integrated.

- **Monocular epipolar line constraint**
  We introduce the necessary formulation to augment the epipolar constraint for the calculation of optical flow using the total variational model in a multi-resolution pyramid scheme. Therefore, we minimize an objective function contains the epipolar constraint with a residual function based on different types of descriptors (BCA, HOG, Census or MLDP). For the calculation of epipolar lines, the relevant fundamental matrices are calculated based on the 7- and 8- point methods. On the other hand, SIFT and Lucas-Kanade methods are used to obtain matched features between two frames, by which we calculate the epipolar geometry. Moreover, we evaluate the effect of using different combinations of the feature matching methods, fundamental matrix calculation, and descriptors based on the challenging KITTI dataset.

- **Real-time multiple object detection and tracking**
  We developed a real-time detection and tracking of multi-moving objects using optical flow. This model separates all moving objects from the static ones and estimates models for object's motion. Therefore, the proposed optical flow algorithm has been optimized using a parallel processing technique. For segmenting moving objects, we developed a camera motion stabilization. Here, we compensate the camera ego-motion and detect moving regions after applying a motion detection algorithm. Afterward, a dense optical flow is estimated for those regions only. Later, 2D motion segmentation based on parallax constraint is applied, and a Kalman filter is modeled to track each object.

This PhD thesis makes use of material from papers by the author as the first author in [MM12c, MM12b, MRM+13, MRM+14, MBM14, MMM15, MMM17], in addition to papers as the second author such as [MM12c, RMG+13, BMM14, MMM16]. Chapter 3 uses material from references [MM12b] and [MM12c] both are co-authored with Baerbel Mertsching. Meanwhile, chapter 4 is based on references [MRM+14, MMM16] and [MRM+13] both are co-authored with H. Rashwan, B. Mertsching, M. Garcia, and D. Puig. Furthermore, chapter 5 is based on references [MMM15] and [MMM17] both are co-authored with M. H. Mirabdollah and B. Mertsching. Finally, chapter 5 is based on reference [MBM14] co-authored with C. Boeddeker, and B. Mertsching. Some materials from each of these papers have been incorporated into this introductory chapter and chapter 2.

1.6 Thesis Outline

Chapter 2 delineates on the problem of apparent motion estimation and analyzing of pixels in image sequences for monocular camera setup. Furthermore, we explain different techniques for motion estimation. Besides, we presented state-of-the-art methods for motion segmentation and tracking of multi-moving objects. Finally, we introduce a framework for evaluation.

Chapter 3 discusses the problem of large displacements optical flow and explains the coarse-to-fine technique with its advantages and disadvantages. Moreover, it presents an approach for improving the coarse-to-fine technique to be able to detect and correctly estimate the 2D motion of fast as well as slow objects. Subsequently, this chapter introduces comparisons and evaluation results.

Chapter 4 acquaints the proposed approach to replace brightness constraint with a texture constraint. Accordingly, several texture descriptors such as HOG, MLDP, and the Census are explained. Primarily, the mathematical model for the variational optical flow model which includes texture information is discussed. Eventually, evaluation and experiment results are discussed.

Chapter 5 introduces the integration of the epipolar constraint for the calculation of optical flow using the proposed optimization algorithm in a multi-resolution

pyramid scheme. On the one hand, the calculation of epipolar lines based on the relevant fundamental matrices based on different methods is presented. On the other hand, different matching schemes between two frames by which fundamental matrices can be calculated are introduced. Belatedly, the evaluation of the algorithm and the effect of using different combinations of the feature matching methods, fundamental matrix calculation and descriptors are evaluated based on the KITTI dataset.

Chapter 6 addresses a novel approach for detecting and tracking moving objects. Firstly, it considers a camera motion stabilization algorithm. Secondly, a motion detection technique to detect the hypotheses of the moving objects is presented. Afterward, the proposed optical flow algorithm based on texture constraint is applied only for those moving regions to getting dense optical flow. Ultimately, the evaluation and experiment results are presented.

Chapter 7 sums up the contributions and the achievements of dynamic scene analysis using optical flow and indicates the direction for future work.

2 Related Literature

2.1 Overview

Motion estimation is a fundamental problem in the analysis of dynamic scenes, with a diversity of applications including video surveillance [CGPP03, TMJP16], structure from motion [Tom92, Oli01, Nis05, DLH14, TZD16, LBCS18, SWH+18], object tracking [YJS06, ST08], motion segmentation [CCBK06, YP06], and advanced driver assistance systems [BWF+03, RMWF10, BHLLR14]. In this chapter, we establish a comprehensive survey of the common algorithms dealing with motion estimation and multi-objects tracking. This chapter is organized as follows: sections 2.2 discusses the motion field, while the problem of motion detection in case of stationary camera is introduced in section 2.3, whereas various algorithms such as background subtraction, kernel density, and codebook construction are briefly introduced. In turn, section 2.4 introduces the motion estimation in case of non stationary camera and objects. Section 2.5 discusses common algorithms for motion estimation using various methods such as block matching, phase correlation and differential optical flow approaches. Section 2.6 introduces different motion constraints and explain a general frame work for optimization. Section 2.7 addresses the differential optical flow estimation, while section 2.8 discusses the motion boundary problem. Coarse to fine optimization is discussed in section 2.8, while various image texture features are explained in section 2.9. Section 2.10 presents stat-of-the-art methods for motion segmentation and multi object tracking. The evaluation and accuracy metrics are discussed in section 2.11.

2.2 Motion Field

The motion field in the 2D image represents the displacements of the projected 3D points from camera- or object movements. For instance, assume a moving camera capturing two frames at time $t = 0$ and at time $t = 1$ of 3D points. Hence, image displacements can be estimated using the camera projection matrix at the last position M_0 and the camera projection matrix at the current position M_1. For simplicity, assume that the world coordinate frame coincides with the frame at time $t = 0$. Therefore, the projection matrix M_0 contains only the camera calibration matrix K: $M_0 = K[I\,|\,0]$, where I is the identity matrix. Hence, the projection matrix M_1 is $M_1 = KR[I\,|\,-\mathbf{t}]$, where R is a rotation matrix and $\mathbf{t}$ is a translation vector.

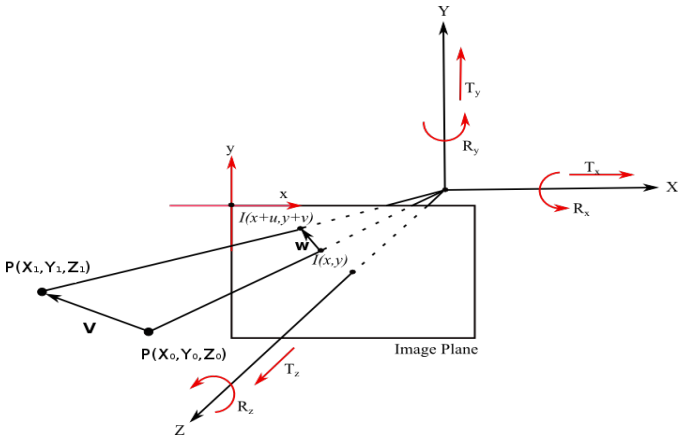


Figure 2.1: Projection of 3D motion vector on an image plane, modified from [Bla92].

The motion field is the projected motion vector $\varnothing\mathbf{w}$ of a 3D point $\mathbf{P}$ between frame at time $t = 0$ and frame at time $t = 1$, and it computes to:

$$M_0\mathbf{P} = \mathbf{p_0} \tag{2.1}$$

$$M_1\mathbf{P} = \mathbf{p}_1 \tag{2.2}$$

$$\varnothing \mathbf{w} = [\varnothing u, \varnothing v]^T = \mathbf{p_0} - \mathbf{p_1} = [u, v]^T, \tag{2.3}$$

where $\varnothing u$ and $\varnothing v$ are the x and y components of the motion vector field for the point $\mathbf{P}$, while u and v are the x and y displacements in the 2D image plane which called optical flow components. Ultimately, Eq. (2.3) concludes that establishing a correspondence between points in the image plane equals to the projection of the 3D points on the images. Unfortunately, this is not always true due to various aspects such as aperture problem, illumination changes and occlusion as explained in section 2.6.3.

Lets assume a 3D point $\mathbf{P}(X, Y, Z)$ at time $t = 0$ rotates in X, Y and Z directions with a rotation matrix $\mathrm{R} = [\mathrm{R}_z^{\gamma}\mathrm{R}_y^{\beta}\mathrm{R}_x^{\alpha}]$ and translates in X, Y and Z directions with a translation vector $\mathbf{t} = [t_x, t_y, t_z]^T$ at time $t = 1$ (see figure 2.1).

The transformation between these two frames can be represented as follow:

$$\begin{bmatrix} X_1 \\ Y_1 \\ Z_1 \end{bmatrix} = \mathrm{R} \begin{bmatrix} X_0 \\ Y_0 \\ Z_0 \end{bmatrix} + \mathbf{t} = \left[\mathrm{R}_z^{\gamma}\mathrm{R}_y^{\beta}\mathrm{R}_x^{\alpha}\right] \begin{bmatrix} X_0 \\ Y_0 \\ Z_0 \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix} \tag{2.4}$$

$$\begin{bmatrix} X_1 \\ Y_1 \\ Z_1 \end{bmatrix} = \begin{bmatrix} \cos\beta\cos\gamma & \sin\alpha\sin\beta\cos\gamma - \cos\alpha\sin\gamma & \cos\alpha\sin\beta\cos\gamma + \sin\alpha\sin\gamma \\ \cos\beta\sin\gamma & \sin\alpha\sin\beta\sin\gamma + \cos\alpha\cos\gamma & \cos\alpha\sin\beta\sin\gamma - \sin\alpha\cos\gamma \\ -\sin\beta & \sin\alpha\cos\beta & \cos\alpha\cos\beta \end{bmatrix} \begin{bmatrix} X_0 \\ Y_0 \\ Z_0 \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix} \tag{2.5}$$

Assume small rotation angels, thus $\cos(\theta) \approx 1$ and $\sin(\theta) \approx \theta$, Eq. (2.5) can be written as:

$$\begin{bmatrix} X_1 \\ Y_1 \\ Z_1 \end{bmatrix} \approx \begin{bmatrix} 1 & -\gamma & \beta \\ \gamma & 1 & -\alpha \\ -\beta & \alpha & 1 \end{bmatrix} \begin{bmatrix} X_0 \\ Y_0 \\ Z_0 \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix} \tag{2.6}$$

Applying for a 3D velocity vector of point $\mathbf{P}$, we get the following system of equations:

$$\begin{bmatrix} X_1 - X_0 \\ Y_1 - Y_0 \\ Z_1 - Z_0 \end{bmatrix} \approx \left(\begin{bmatrix} 1 & -\gamma & \beta \\ \gamma & 1 & -\alpha \\ -\beta & \alpha & 1 \end{bmatrix} - \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \right) \begin{bmatrix} X_0 \\ Y_0 \\ Z_0 \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix} \tag{2.7}$$

The 3D velocity vector of a rigid object can be formulated as:

$$\begin{bmatrix} V_x \\ V_y \\ V_z \end{bmatrix} \approx \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix} \begin{bmatrix} X_0 \\ Y_0 \\ Z_0 \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix} \tag{2.8}$$

The vector form of Eq. (2.8) can be written as:

$$\mathbf{V} = \mathbf{\Omega}.\mathbf{P_0} + \mathbf{T} \tag{2.9}$$

where $\mathbf{\Omega}$ is a 3×3 matrix containing the angular velocity vectors, and $\mathbf{T}$ is the translational velocity vector.

Under orthographic/affine projection, Eq. (2.9) can be written as follow:

$$\begin{aligned} u &= V_x = t_x + \omega_y Z_0 - \omega_z Y_0 \\ v &= V_y = t_y + \omega_z X_0 - \omega_x Z_0 \end{aligned} \tag{2.10}$$

Eq. (2.10) represents an affine motion model which has 6D of freedom. Accordingly, to estimate the 6 parameters of an affine motion model, three correspondences points are sufficient. Nevertheless, for estimating a robust model using the least square or the RANSAC, more points are required.

Under perspective projection, Eq. (2.9) can be written as follow:

$$\begin{aligned} u &= f\frac{V_x}{Z_0} - x\frac{V_z}{Z_0} = f(\frac{t_x}{Z_0} + \omega_y) - \frac{t_z}{Z_0}x - \omega_z y - \frac{\omega_x}{f}xy + \frac{\omega_y}{f}x^2 \\ v &= f\frac{V_y}{Z_0} - y\frac{V_z}{Z_0} = f(\frac{t_y}{Z_0} + \omega_x) - \frac{t_z}{Z_0}y - \omega_z x + \frac{\omega_y}{f}xy + \frac{\omega_x}{f}y^2 \end{aligned} \tag{2.11}$$

where $x = f\frac{X_0}{Z_0}$, and $y = f\frac{Y_0}{Z_0}$.

The perspective motion model in Eq. (2.11) has nine degree of freedom, and it requires a minimum four correspondence point to find the motion parameters. To that end, estimating optical flow leads to estimating an approximation of 3D motion models using two frames captured from a monocular camera.

2.3 Motion Detection

Motion detection in an image sequence is the process of identifying changes in positions of pixels relative to their surroundings [CD00], [BBV08]. The changes in positions result from appearance or disappearance, the motion, or shape changes of the object. Moreover, reflection and lighting changes afford a change in brightness or color on stationary objects. Consequently, robust motion detection approaches should differentiate between real moving regions and false moving regions which contain disturbances, such as those induced due to camera motion, lighting variation, sensor noise, or atmospheric absorption [ACPP05]. In the literature, conventional approaches for detecting motion from a given sequence of images use background subtraction or optical flow estimation as a base.

2.3.1 Background Subtraction (BS)

Background subtraction techniques contain two main steps: constructing the background model and then detecting the foreground. The background model consists of three phases [CBM03]: an initialized model, a represented model, and an updated model. The better initialization model is, the better background model. Hence, to avoid the acquisition of an incorrect background of the scene, analyzing sequences of images with a presence of moving objects should consider various initialization schemes [CBM03]. Typically, background subtraction algorithms assume a stationary camera and aim at the detection of foreground objects using the absolute difference between a frame and a background model of a scene. Nevertheless, a robust background subtraction algorithm is not a trivial task [RCH03], [BBV08] due to illumination changes, camera ego-motion or non-stationary background (e.g. moving leaves, rain, and shadows of moving objects).

In the literature, background subtraction approaches are divided into three basic categories: mixture of Gaussians [FR97], kernel density estimation [EHD00] and codebook construction [KCHD05].

2.3.2 Mixture of Gaussians (MoG)

This approach describes each pixel by its intensity. Afterwards, the probability of observing a pixel's intensity $I(x, y, t)$ in a multidimensional state is expressed utilizing a Gaussian probability density function, which can be expressed in [FR97] as follows:

$$pdf(I(x, y, t)) = \sum_{i=1}^{k} \omega_{\mathrm{i,t}} \eta(I(x, y, t), \mu_{i,t}, \sigma_{i,t}), \tag{2.12}$$

where k is the number of Gaussian distributions, $\omega_{\mathrm{i,t}}$ is a weight for the i Gaussian at time t with mean $\mu_{i,t}$ and standard deviation $\sigma_{i,t}$. η is a Gaussian probability density function [FR97] described as follows:

$$\eta(I(x, y, t), \mu_{i,t}, \sigma_{i,t}) = \frac{1}{(2\pi)^{n/2}|\sigma|^{0.5}} e^{-\frac{1}{2}(I(x,y,t)-\mu)\sigma^{-1}(I(x,y,t)-\mu)}. \tag{2.13}$$

The MoG technique is robust against illumination changes [SCK04]. Unfortunately, mixture Gaussian model performs badly in presence of dynamic textures such as moving leaves, rain and moving shadows. Furthermore, it provides non-coherence foreground objects that contain empty regions [RSPMB16].

2.3.3 Kernel Density Estimator (KDE)

Elgammal et al. [EHD00] proposed the KDE technique to estimate the probability density function *pdf* for every pixel using latest frames of a video stream. Typically, KDE uses a Parzen-window for every pixel and defines 1D kernel. Ultimately, a classification of foreground and background pixel is applied based on the likelihood of the current pixel using a predefined threshold for the *pdf* [SM10]. This approach can analyze sequences with multi-modal backgrounds, and it is robust to noisy input. Nevertheless, it suffers from the problem of dynamic textures and outdoor conditions as reported in [RSPMB16].

2.3.4 Codebook Construction (CC)

Codebook construction is an adaptive background subtraction technique. In fact CC models a background from a training sequence [KHDL04], [KCHD05]. Since change is only due to the brightness, the CC model assumes that the background pixel intensities are laying along the principal axis of the codeword bounded by the low and high pixel intensity [KCHD04]. The CC model is not only faster but also more robust than the MoG model [EHD00] and the KDE model [FR97] in several background modeling problems. Nevertheless, it suffers from those mentioned above outdoor environmental issues [RSPMB16].

2.4 Motion Estimation for Moving Camera and Objects

In literature, motion estimation form a moving camera problem is addressed in three different categories. Firstly, 3D-3D motion estimation by calculating point correspondences based on sets of 3D points. The shortcomings of this category are the difficulty to have valid 3D points particularly in a case of an outdoor environment and the massive cost of data processing. Secondly, 2D-3D motion estimation determines 3D motion based on finding correspondence between 3D model and 2D image projections which needs 3D model analysis. Finally, 2D-2D calculates correspondences between 2D image projections and estimates 3D motion models from such 2D correspondences. Constructing 2D correspondences can be accomplished using point, line, curve, texture, or region correspondences.

2.5 Optical Flow

Motion estimation using optical flow is an example of the 2D-2D category. Typically, it describes the apparent motion projected on an image plane of a camera as local gray value displacements. Hence, it results from the motion of objects in a scene, the motion of the observer camera, or illumination changes. Excluding the illumination changes, a flow field describes the dynamics of a scene and involves the camera ego-motion and the motion of the object. Nevertheless, it is

a challenging task to segment two independent motion without pre-knowledge. Figure 2.1 illustrates the projection of 3D motion vector on an image plane.

Although optical flow calculation has been investigated for a long time, it is still known as an open problem especially if the flows are due to large motions and for low textured regions [MG15]. Various methods based on block-based minimization, phase correlations, discrete optimization, and differential estimation exist. In the coming sections, we introduce the basic concepts and the main categories of optical flow methods. For more details, the reader is directed to [BSL$^+$11, FBK15].

2.5.1 Block Matching

The Block Matching algorithm is an example of pixel-accurate optical flow where a spatial-domain search procedure is applied to find the best match point by evaluating a pixel matching score based on gray values of a pixel's neighborhood. Normally, a specific search strategy is used to match a block around a pixel in the current frame with blocks of the previous frame using a specific criterion. However, several criteria can be used such as mean squared error (MSE), minimum absolute difference (MAD) and sum absolute difference (SAD) . For a certain pixel, the number of possible flow vectors or displacement vectors is bounded by the image size and the outliers are less likely. On the one hand, pixel accurate optical flow algorithms are robust due to the limited solution space. On the other hand, these approaches can be parallelized using dedicated hardware which yields real-time performance. Nevertheless, a major drawback of these approaches are accuracy limitations, i. e. sparse optical flow and pixel accurate displacements.

2.5.2 Phase Correlation

On the contrary to the block matching approaches described above which searches the blocks using intensity matches, a phase-correlation approach estimates the motion between two frames from their phases in the frequency domain. In fact, a translational shift between two frames in a spatial domain is reflected as a phase change in the spectrum domain. Therefore, the normalized cross-correlation

NCC between two frames can be applied to find the motion vectors. Accordingly, it yields a robust estimate of a motion field with much lower entropy.

2.5.3 Differential Optical Flow

The most exciting approach for the calculation of optical flow is the differential optical flow based on total variational optimization. In fact, a differential optical flow approach cast different constraints as a single cost function resulting in well established elegant models which take care of all objectives simultaneously. For instance, several differential optical flow approaches minimize an objective function based on the brightness constancy assumption or gradient constancy assumption as well as a smoothness constraint.

Differential optical flow is an example of sub-pixel accurate optical flow categories. The basis of a differential optical flow approach is the motion constraint equation which has to be minimized after applying differentiations of the image in the spacial and the time domain. Typically, motion constraint assumes a gray value image sequence has a continuous domain, and intensity variations in the images are due to the motion of the objects present in the depicted scenes [RPG12]. Therefore, the image gradient can be used to estimate the change of a pixel's gray value between two consecutive frames.

2.6 Data Conservation For Differential Optical Flow

The intensity (brightness) captured by a camera sensor at a specific pixel usually is proportional to the amount of the reflected light from the depicted point in the environment. The amount of the reflected light depends on the reflectance property of the surface and the illumination [Sha12]. The following subsections conclude various data conservation assumptions underlying most optical flow methods.

2.6.1 Brightness Constancy Assumption (BCA)

The brightness constraint assumption assumes a constant intensity of a pixel, if objects or the camera move. Assuming two consecutive gray images $I(t)$ and $I(t+1)$, the goal of motion estimation is to map each pixel $\mathrm{p} = (x, y)^T$ in image $I(t)$ to the corresponding pixel $\mathrm{p}' = (x + dx, y + dy)^T$ in the image $I(t+1)$ such that

$$I(x, y, t) = I(x + dx, y + dy, t + dt). \tag{2.14}$$

Eq. (2.14) is nonlinear in terms of dx and dy. Using the Taylor expansion, the brightness consistency assumption in Eq. (2.14) can be approximated as follows:

$$I_x(x, y)\frac{dx}{dt} + I_y(x, y)\frac{dy}{dt} + I_t(x, y) + HOT = 0 \tag{2.15}$$

where $I_x(x, y) = \frac{\partial I_1(x,y)}{\partial x}$, $I_y(x, y) = \frac{\partial I_1(x,y)}{\partial y}$ and $I_t(x, y) = I_2(x, y) - I_1(x, y)$. Assuming $dt = 1$, $u = \frac{dx}{dt}$, $v = \frac{dy}{dt}$ and neglecting the high order terms (HOT), Eq. (2.15) can be written as follows:

$$I_x(x, y)u + I_y(x, y)v + I_t(x, y) = 0 \tag{2.16}$$

For each pixel, Eq. (2.16) yields only one linear equation to solve for two unknowns u and v which results in an under-determined equation system that yields an infinite number of solutions. Possible solution are presented in section 2.7. The brightness constancy assumption fails to find a good match in case of rotation around the lens axis. Moreover, a shadow of a moving object lying on another object or the background confuses approaches that use brightness constraint.

2.6.2 Gradient Constancy Assumption (GCA)

It is convincing to evade the problem by considering a constant image gradient when objects or cameras move [RPG12]. This yields an assumption namely

gradient constancy assumption between two frames $I_1(x, y, t)$ and $I_2(x + u, y + v, t + dt)$ which is formed as follows:

$$\nabla_3 I_1(x, y) - \nabla_3 I_2(x + dx, y + dy) = 0, \tag{2.17}$$

where $\nabla_3 = (\frac{\partial}{\partial x}, \frac{\partial}{\partial y}, \frac{\partial}{\partial t})$.

Although the gradient constancy is robust with respect to illumination changes, it is sensitive to noise. In chapter 4, we introduce a new motion constraint called the texture constancy assumption which uses image gradients to gain robustness with illumination changes as well as image noise.

2.6.3 Differential Optical Flow Violation

Aperture Problem

As mentioned in section 2.2, the optical flow is not perpetually corresponding to the motion field due to several reasons such as the aperture problem, sensor noise, and illumination changes. Typically, the aperture problem arises due to the ambiguity of one-dimensional motion viewed through an aperture [Wed09]. Thus, the brightness constancy assumption provides only optical flow in the same direction of the spatial image gradient, and it is not possible to determine optical flow perpendicular to the image gradient.

As shown in figure 2.2, the background of the stripe is hidden by an occluding bull's aperture [Wed09]. Hence, in case that stripes move upwards, the line pattern shifts in the aperture. Similarly, if the stripes move to the left, the pattern shifts in the same way. Therefore, we are facing a motion ambiguity which can only be solved if we know the motion of the boundary of the pattern [Wed09]. Accordingly, the optical flow does not always represent the motion field, while it represents the apparent motion of the scene. Ultimately, the estimation of unique optical flow requires other assumptions and constraints. In the literature, there are two basic assumptions which can be applied; the global [HG81] and local smoothness constraint [LK81].

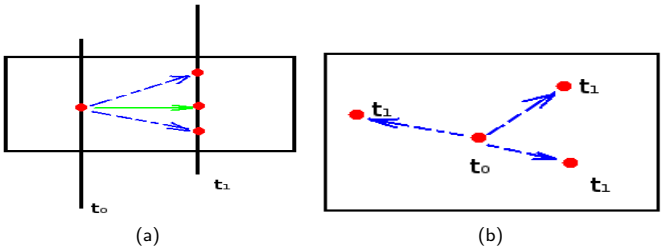


Figure 2.2: Aperture problem. (a) Only the orthogonal component of the flow to $\nabla_2 I$ is computable (green arrow). (b) No information available. Correspondences may lie everywhere.

Occlusion

Occlusion occurs when a 3D point in the background/foreground seen in one frame has disappeared in the next frame due to a moving foreground object. As a result, there exists no corresponding point pair associated with that 3D point in the two frames. For an algorithm, the occlusions are not known a priori, and it tries to find matching gray-value structures between frames and may find wrong matches [Jen08].

2.7 Differential Optical Flow Estimation

The differential optical flow techniques can be divided into three main categories.

- Local methods filter image gradients in a local neighborhood around a pixel and assume that the velocity field of a small patch of pixels changes gradually.
- Global methods use a global optimization procedure based on a regularization term for estimating flow field.
- Combined local global methods combine the advantages of local and global methods. Local methods produce robust flow fields in the case of image noise; however, they fail to obtain a dense optical flow field.

In contrast, global methods present dense optical flow fields, yet they are more sensitive to noise. Next, we recall the most outstanding and state-of-the-art differential approaches for optical flow.

2.7.1 Local Smoothness Based Methods

Lucas/Kanade (LK-OF) [LK81] is the primary approach that uses local methods. It assumes a uniform optical flow in the local neighborhood of every pixel and estimates the flow by applying the least squares minimization technique for a local group of pixels as follows:

$$\nabla_2 I \boldsymbol{w} = -\boldsymbol{I_t}, \tag{2.18}$$

where I is an image, $\nabla_2 I$ is the spatial gradients and I_t the temporal gradient . In practice, the weighted version of the least squares equation is often used to assign a higher weight for the closer pixel to the center pixel and to apply a Gaussian kernel around it. Since the input gradients are filtered out, these approaches yield a good noise tolerance [RPG12]. Nevertheless, local smoothness methods produce no flow fields in homogeneous image regions due to the not existing gradients. Hence, the estimated optical flow is sparse flow, and it is valid only for features point that does not have gradients different from zero.

2.7.2 Global Smoothness Based Methods

The method of Horn/Schunk (HS-OF), [HG81] is an instance of the global smoothness category. It uses two assumptions: The brightness constancy assumption by assuming a constant gray value of objects over time and a homogeneous regularization which assumes that the resulting flow field is smooth everywhere. Therefore, it optimizes an objective function by minimizing the spatial variation of the resulted flow field over the whole image to solve the aperture problem.

By formulating these two assumptions mathematically, the following objective function can be formulated:

$$\mathcal{E} = \int (I_t + I_x u + I_y v)^2 + \alpha(||\nabla u|| + ||\nabla v||) d\Omega. \tag{2.19}$$

Global smoothness based methods yield dense flow fields even in homogeneous image regions. However, the main shortcoming of these methods is the boundaries of the motion segments as they do not allow discontinuities in the optical flow field and do not handle outliers in the data-term robustly [Wed09].

Many algorithms have been proposed to eliminate the drawbacks of simple HS-OF [HG81]. For instance, the total variational optical flow algorithm penalizes the derivative of the optical flow field, yielding an objective function which is minimizing the summation of the amount of optical flow variation or fluctuation in the whole image [Wed09]. Eq. 2.19 uses the integral of the L_2 norm of the gradient which is also called the total variation TV norm. In contrast to the original quadratic L_2 -regularity suggested by Horn and Schunck [HG81], the L_1 -regularity preserves motion discontinuities better [PUZ+07].

2.7.3 Combined Local Global (CLG) Based Methods

To avoid the drawbacks and achieve the advantages of both local and global approaches, Bruhn et al. [BWS05] proposed a combination between the local and global optical flow methods as described in the pervious subsection. This work has introduced a unifying multi-grid approach to variational optical flow computation in a real-time and has examined the smoothing effects in local and global differential methods. The proposed method [BWS05] extracts local information by applying a Gaussian filter on the pixel's neighborhood and assumes global smoothness of the flow field. CLG combines the advantages of both local and global algorithms. Typically, it is robust against noise and provides an accurate dense flow field based on multi-grid techniques to speed up the minimizing of the main optimal procedure with a regularization.

$$\mathcal{E}(\mathbf{w}) = \int_{\Omega} \left[\psi \left(\mathbf{w}^T J_\rho \left(\nabla_3 f \right) \mathbf{w} \right) + \lambda \psi \left(\nabla \mathbf{w} \right) \right] d\Omega \tag{2.20}$$

where ψ is a convex function, J_ρ is a Gaussian kernel and λ is a weight between the optical flow constraint and the regularization term.

The shortcoming of this method is the usage of a Gaussian kernel with structure tensors which is an anisotropic filter and does not preserve discontinuities and boundaries of the motion of the objects.

2.7.4 Variational Optimization Framework

A variational optical flow algorithm contains typically two terms; a data-term which contains an optical flow constraint, and a regularization term which penalizes high variations in the optical flow field to obtain smooth fields.

A common framework for the refinement of the optical flow based on the differential optical flow methods starts with an evaluation of a data-term. This evaluation can be performed independently for each pixel. The advantage of such an algorithm is that it can be accelerated by utilizing multiple processors and parallel computing power, e. g. graphics processing units (GPUs). However, the disadvantage is that the pixel itself only contributes locally to the solution. This leads to noisy flow fields as a result corrupted image data due to sensor noise, low entropy (information content) in the image data, and illumination changes [Wed09]. Assuming uncorrelated noise with zero mean, a common approach for noise reduction effect is a subsequent smoothing operation [Wed09]. However, flow field smoothing leads to bluring discontinuities that exist in the correct flow field, especially at motion boundaries. Hence, there is a need to employ discontinuity preserving filters which yields a piecewise-smooth flow field [Wed09].

To solve for the aperture problem, information from a region's boundary should be propagated into the interior. Hence, the local smoothing does not wholly solve the aperture problem, unless the filter mask is chosen large enough. In turn, global techniques are propagating information across the whole image and they can be used to solve the aperture problem. The three primary objectives for the smoothing step are discarding outliers in the flow field due to corrupted image data (denoising), preserving edges (i. e. do not smooth over edges) and propagating information into areas of low texture. However, illumination changes are different

from the above mentioned problems. Therefore, denoising or smoothing [Wed09] cannot remove them.

2.8 Coarse-To-Fine Optimization

Most of the differential based approaches use the Taylor approximation to linearize the brightness constraint. As a result, the solution is only valid for small displacement vectors [BGM04]. Larger displacements are solved by embedding the method into a coarse-to-fine pyramid approach. A Coarse-to-fine strategy find solutions for low frequency structures at low resolution images and refines the solution on higher resolution images (see figure 2.3). The maximum track length of the coarse-to-fine levels depends on image content; large displacements flow vectors are less likely to be found in images than those within a small pixels displacement [Wed09].

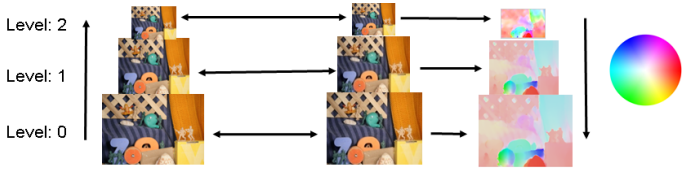


Figure 2.3: An example of the coarse-to-fine approach using 3 levels applied on the "Army" sequence from Middlebury dataset.

2.9 Image Texture

Recently, approaches have been trying to use image features to calculate robustly optical flow. In the following, we explain some of these image features that can be used to estimate a robust optical flow field.

2.9.1 Structure Texture Decomposition via Total Variation (ROF)

The ROF [ROF92] model assumes that images are contaminated by noise. The decomposing purpose is to remove the noise by separating the image into signal and noise parts. Certain assumptions are taken such as the piecewise smooth nature of the image, which enables good approximations of the clean original image [AGCO05]. The successful primary approaches for image denoising depend on the minimization of an energy function based on norms of the image gradient by solving nonlinear partial differential equations (PDE). Therefore, ROF algorithm preserves the edges of the original image and removes most of the noise by decomposing an image I into two components I_u and I_v and minimizes the following objective:

$$\min_{(I_u,I_v)/I=I_u+I_v} \left(\int \mid DI_u \mid +\lambda \|I_v\|^2 \right), \tag{2.21}$$

where $\int \mid DI_u \mid$ is the total variation of I_u. By applying ROF, an algorithm can gain robustness against illumination changes.

2.9.2 Normalized Cross Correlation (NCC)

A normalized cross-correlation is a conventional approach to feature detection. It is usually used as a metric to evaluate the similarity between two matched feature vectors or images. Hence, it is less sensitive to linear changes in the amplitude of illumination in the two compared images. Furthermore, the NCC is restricted to the range between -1 and 1. Typically, the NCC is widely used for finding matches of a reference template $T_1(x, y)$ with size $m \times n$ in an image $I_1(x, y)$ of size $M \times N$, and a matching template $T_2(x + d1, y + d2)$ of size $m \times n$ in a scene image $I_2(x, y)$.

$$C(x, y, d) = \frac{1}{n} \sum_{x,y \in \mathcal{N}} \frac{(T_1(x, y) - \bar{T_1})(T2(x + d1, y + d2) - \bar{T_2})}{\sigma_{T_1} \sigma_{T_2}}, \tag{2.22}$$

where n is the number of pixels, $\bar{T_1}$ and $\bar{T_2}$ are the averages of $T_1(x, y)$ and $T_1(x, y)$, and σ_{T_1} and σ_{T_2} are the norms of $T_1(x, y)$ and $T_1(x, y)$, respectively.

2.9.3 Census Transform (CT)

Recently, variational optical flow methods used local image descriptors such as the census transform to achieve robustness against illumination changes [Ste04a, MRR+11]. The census transform is a texture descriptor which initially has been used for face detection. It is a form of non-parametric local transform based on the relative ordering of local intensity values, and not on the intensity values themselves. Census transform maps the intensity values of the pixels within a block to a bit string. The center pixel's intensity value is replaced by a string composed of a set of boolean comparisons. For each comparison the bit is shifted to the left, forming an 8 bit string for a census window of size 3×3 and a 24 bit string for a census window of size 5×5, based on the following equation:

$$\xi(p, p_{i,j}) = \left\{ \begin{array}{ll} 1 & I(p) - I(p_{i,j}) >= \varepsilon \\ 0 & \text{otherwise.} \end{array} \right\}, \tag{2.23}$$

where $I(p) = I(x, y)$ and $I(p_{i,j}) = I(x + i, y + j)$ and ε is a threshold to deal with noise.

Census transform reduces effects of global illumination changes and the variations caused by camera gain and bias. Moreover, it is robust to outliers points which are located near depth discontinuities and encodes the local spatial structure. However, the census transform is sensitive to non-monotonic illumination changes. Furthermore, it fails to solve the affine motion problem (i.e., rotatory motion). Besides, it cannot handle the problem of blocks with saturated center pixels, which means all neighbors are greater or smaller than the value of the center pixel [MRM+13].

The modified census transform [FE04] is a modified version of the census transform, and it represents pixels which have an intensity value higher than the mean or the median pixel intensity value within a particular block. It forms a 9 bits string for a census window of size 3×3 and a 25 bits string for a census window of size 5×5. The modified census is used to distinguish between the darkness and brightness regions that the original census transform fails to detect it.

Stein et al. [Ste04a] introduced a ternary signature inspired from the census transform. The ternary census transform maps a local neighborhood surrounding a pixel p to a ternary string representing a set of neighbors pixels. A ternary census signature $\xi(I(p), I(p_{i,j}))$ is defined as:

$$\xi(p, p_{i,j}) = \left\{ \begin{array}{lll} 0 & I(p) - I(p_{i,j}) & > \varepsilon \\ 1 & |I(p) - I(p_{i,j})| & \leq \varepsilon \\ 2 & I(p_{i,j}) - I(p) & > \varepsilon \end{array} \right\} \tag{2.24}$$

For all census variants, the choice of an optimal threshold is always a challenging problem, and it is an experimental issue based on the application.

2.9.4 Histogram of Oriented Gradients (HOG)

The HOG descriptor proposed in [DT05] uses the dominant edge orientations to construct a robust local descriptor. It is calculated after applying a gradient operator $\frac{d}{d_x}$ and $\frac{d}{d_y}$, within a local window ($N \times N$) using a centered derivative mask (i.e. $[-1, 0, 1]$). Afterwards, the magnitude and orientation of the resulting gradient vector at every pixel are computed. The range of orientations angle between $[0 \ldots 2\pi]$ is divided into some bins n. At each pixel, we count the number of angles associated with each bin in a local window and calculate the probability distribution of the angles. Regularly, the histogram is normalized eventually using the following norm: L_2, L_1 or $L_2 - sqrt$ [DT05]. Figure 2.4 shows an example of HOG descriptor.

2.9.5 Distributed Average Gradient (DAG)

DAG [MMM16] is a modified version of HOG, which calculates the averages of gradients in four surrounding windows about each pixel. It is a short descriptor with the length of 8 bins which is shown in figure 2.5. DAG uses an overlapping window in which their size can be changed, but the number of windows always remains four. This make DAG robust against abrupt changes of the gradients

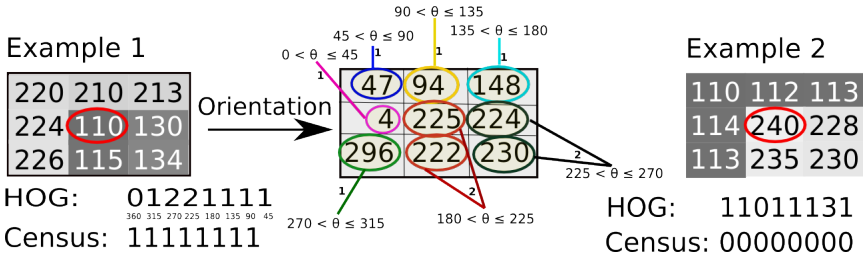


Figure 2.4: An example of a 3×3 HOG descriptor vs. a 3×3 census descriptor (from [RMG+13] with permission from Springer).

at the borders of the object. The average gradient v_i for each sub-window $w_i : i = 1, 2, 3, 4$ is calculated as follows:

$$\mathbf{v}_i = [v_{i,x} \; v_{i,y}]^T = \frac{1}{N} \sum_{(x,y) \in w_i} \mathbf{v}(x, y) \tag{2.25}$$

where $N = (S/2 + 1)^2$. By concatenation of the four vectors, a descriptor vector as following is formed:

$$\mathbf{d} = [v_{1,x}, \; v_{1,y}, \; v_{2,x}, \; v_{2,y}, \; v_{3,x}, \; v_{3,y}, \; v_{4,x}, \; v_{4,y}]^T \tag{2.26}$$

A normal vector in the direction of the average of gradients in a neighborhood of each point is taken into account: $\mathbf{g} = [g_x \; g_y]^T$, to obtain a rotation invariant DAG. Hence, the orthogonal vector to $\mathbf{g}$, which is called $\mathbf{k}$ is used to build a local coordinate system based on $\mathbf{g}$ and $\mathbf{k}$. The four windows about a point are considered to be using four sets of vectors.

$$\{\mathbf{g}_1 = -\mathbf{g}, \; \mathbf{k}_1 = \mathbf{k}\}, \; \{\mathbf{g}_2 = \mathbf{g}, \; \mathbf{k}_2 = \mathbf{g}\}, \; \{\mathbf{g}_3 = -\mathbf{g}, \; \mathbf{k}_3 = -\mathbf{g}\}, \; \mathbf{g}_4 = \mathbf{g}, \; \mathbf{k}_4 = -\mathbf{g}\} \tag{2.27}$$

To address pixels in surrounding windows, the following equation is used:

$$\mathbf{p} = [x \; y]^T = h\mathbf{g}_i + w\mathbf{k}_i \tag{2.28}$$

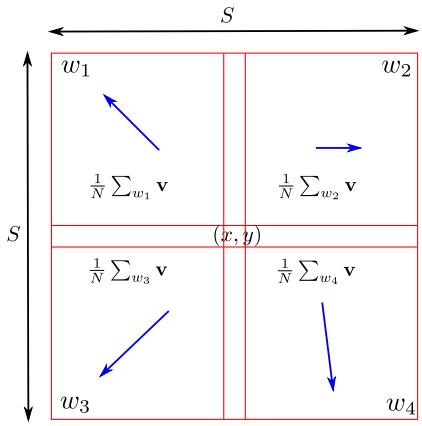


Figure 2.5: DAG descriptor. Each arrow indicates the average of gradients vector in a square window (from [MMM16] with permission from Springer).

where i is the index of a window, $h = 0, .., \frac{S}{2} + 1$ and $w = 0, ..., \frac{S}{2} + 1$. After the calculation of the averages of gradients in the four windows, it is necessary to project the averages on the axis of the rotated coordinate system ($\mathbf{g}$ and $\mathbf{k}$).

2.9.6 Local Directional Pattern (LDP)

The LDP operator proposed by [JKC10] describes a gray-scale textures. It encodes the directional information of edges in a window, instead of the intensity values. Hence, the LDP descriptor represents the local primitives such as different types of curves, corners, and junctions. The LDP descriptor is computed based on the responses of edges ($\mathcal{ER}$) in all directions at each pixel after applying the compass Kirsch masks or Gaussian mask centered on the current pixel center [JKC10]. Corners, feature points or edges cause high response values in some directions. Therefore, the LDP descriptor uses the k ($k = 3$) most predominant directions to construct a binary string. These top k directional bit responses are set to 1, and the other directions are set to 0, resulting in an 8-bit binary string. Unlike the census transform, the LDP can cope with image blocks with a saturated center

pixel. In turn, the census transform fails to distinguish between different textures containing dark and light regions.

A modification of the LDP descriptor is called Local Directional Number Pattern (LDNP) [RRCC13] which is a 6-bit binary string. Unlike the LDP mentioned above, LDNP is more robust to noise since it encodes the direction in a number instead of bit strings to describe the information of the neighborhood. Consequently, LDNP computes the edge responses of the neighborhood using a modified version of a compass mask based on a Gaussian filter.

Relying only on the k most prominent directions LDP is a proper feature descriptor for some applications such as face detection [RRCC13]. However, LDP yields structural information loss in a neighborhood in case of optical flow estimation. Moreover, LDP features depend on edge responses while the directions of edges are missing. Besides, LDP and its modified versions consider all different directions in the same manner. Further, the location of the maximum and minimum responses are not sufficient to describe a neighborhood that can be efficiently used to estimate a dense optical flow [MRM$^+$14]. Ultimately, LDP and its modified versions are not suitable descriptors to be used for the calculation of optical flow. In chapter 4, we proposed a new version of LDP which is robust and well suited for optical flow estimation.

2.10 Tracking of Multiple moving Objects

A variety of methods have been proposed to track multiple moving objects with stationary cameras based on background modeling [MBM14]. For instance, Zhu et al. [DZ12] presented a real-time approach for short-term tracking of multiple objects. This approach detects moving objects in stationary scenes after applying a motion detection approach. For a moving camera, an adaptive background algorithm is used to manage illumination variations, a dynamic background, and the camera ego-motion. However, this algorithm fails in complex scenes and when the camera moves fast. In turn, Stalder et al. [SGG09] proposed an algorithm to track a single object after constructing a model consisting of low-level features and searching for its new location in each frame.

Enzweiler et al. [EG11] proposed a multilevel mixture-of-experts approach. This approach aims at improving the pedestrian classification by combining information from multiple features and cues. On the contrary, Talukder et al. [TM04] combined dense optical flow and stereo which yields an estimation of the background motion and objects motion. Moreover, Henriques et al. [HCB11] proposed a graph-based structure that allows the computation of the solution to be in polynomial time. It encodes multiple-match events as standard one-to-one matches. Furthermore, Wojek et al. [WWR+13a] proposed a probabilistic 3D scene model that integrates a geometric 3D reasoning with multi-class object detection/tracking, and scene labeling.

Most approaches for real-time object-tracking from a moving camera used sparse features or assumed flat scene structures [ST94], [ML11]. Sparse optical flow is valid only at few feature points (i.e. corners). However, it is hard to infer an object's shape and its boundaries from a set of sparse feature points [MBM14]. In turn, dense optical flow reveals important information about objects by distributing all pixels through a regularization [BM11], [SBK10] and [RMG+13]. Although dense optical flow is a substantial input for motion segmentation and flow-based object tracking, the high computational load that affects the real-time performance, is a big hindrance, especially for high-resolution images. Peter et al. [ST06] proposed an algorithm that produces a set of spatially dense and temporally smooth trajectories by combining feature points tracking and dense optical flow fields. Additionally, Rubinstein et al. [RLF12] suggested an algorithm to use an initial estimate for a global solution based on long-range motion trajectories by leverage local trajectories.

2.11 Metrics for Accuracy and Performance

To estimate the accuracy of the computed optical flow, we compare the estimated optical flow to some provided ground truth data. For this purpose, we apply the standard Middlebury [BSL+11] and KITTI [GLSU13] evaluation metrics. In this section, we shade the light on the standard evaluation benchmarks used for the evaluation of optical flow in this these.

2.11.1 Angular Error AE

The angular error AE is the angle in $3D$ space between the estimated optical flow (u, v) and the corresponding ground truth vector (u_{GT}, v_{GT}):

$$\mathrm{AE} = cos^{-1}\left(\frac{1 + u \cdot u_{GT} + v \cdot v_{GT}}{\sqrt{1 + u^2 + v^2} \cdot \sqrt{1 + u_{GT}^2 + v_{GT}^2}}\right). \tag{2.29}$$

As the error for the whole image is relevant instead of calculating it only for a single pixel, the average angular error AAE for the optical flow of the entire image is computed as:

$$\mathrm{AAE} = \frac{1}{N} \cdot \sum_{i=0}^{N} \mathrm{AE}_i \ , \tag{2.30}$$

where N is the number of pixels per image.

2.11.2 End-point Error EE

The End-point Error EE expresses the absolute error between two vectors. Hence, it considers the lengths of the optical flow vectors. The EE is formulated as:

$$\mathrm{EE} = \sqrt{(u - u_{GT})^2 + (v - v_{GT})^2}. \tag{2.31}$$

The average point error AEE is computed as:

$$\mathrm{AEE} = \frac{1}{N} \cdot \sum_{i=0}^{N} \mathrm{EE}_i \ , \tag{2.32}$$

where N is the total number of pixels in an image.

2.11.3 Percentage of Outliers AEE_{out}

The KITTI benchmark [GLSU13] introduced an additional metric based on the EE, to consider the individual optical flow values of each pixel. The percentage of outliers (also called bad pixels) returns the relative amount of pixels in percentage

which have an EE greater than a given threshold. Consequently, with a proper threshold, this metric gives an impression about the percentage of outliers, which have a significantly worse EE than the AEE:

$$\mathrm{AEE}_{\mathrm{out}} = \frac{100}{N} \cdot \sum_{i=0}^{N} [\mathrm{EE}_i > threshold] \ . \tag{2.33}$$

The AEE and AAE is not always sufficient to evaluate the algorithm; therefore the histogram of errors and the error image [GLSU13] are used to evaluate the optical flow algorithms.

2.11.4 Interpolation Error IE

The interpolation error IE is defined as the root-mean-square (RMS) difference between the ground-truth image and the estimated interpolated image [BSL+11].

$$\mathrm{IE} = \left[\frac{1}{N} \sum_{(x,y)} (I(x,y) - I_{GT}(x,y))^2 \right]^{\frac{1}{2}} \tag{2.34}$$

where N is the number of pixels. For color images, the $L2$ norm of the vector of RGB color differences can be used. A second measure of the interpolation performance called a gradient-normalized RMS error can also be computed.

2.11.5 Normalized Interpolation Error NIE

The normalized interpolation error [BSL+11] NIE between an interpolated image $I(x,y)$ and a ground-truth image $I_{GT}(x,y)$ is given by:

$$\mathrm{NIE} = \left[\frac{1}{N} \sum_{(x,y)} \frac{(I(x,y) - I_{GT}(x,y))^2}{\|\nabla I_{GT}(x,y)\|^2 + 1} \right]^{\frac{1}{2}} \tag{2.35}$$

For color images, the $L2$ norm of the vector of RGB color differences is used and compute the gradient of each color channel separately.

2.11.6 Performance Metrics

A reasonable performance metric for a program, performed on a non-real-time operating system, is the average wall-clock execution time. It is the real elapsed time between the start and the end of the program. As a non-real-time operating system may execute a non-deterministic amount of other tasks in between and additionally the status of the CPU caches is non-deterministic, the average value over several measurements has to be taken.

An alternative metric is to measure the CPU time, which only counts the time where the CPU is executing instructions of this program. As this neglects the times of any interrupts by the operating system, the CPU time is typically significantly lower than the wall-clock time. Although this metric might be useful to measure the average CPU utilization of a program, it is not sufficient to determine, how many FPS the program can process on a system. Moreover, it is impossible to guarantee a specific amount of FPS on a non-real-time operating system. Only a real-time operating system allows the specification of hard deadlines and ensures that they are met under all circumstances.

3 Proposed Multi-Resolution Optimization

The variational optical flow approach belong to the state-of-the-art family of approaches for the optical flow estimation. This approach optimizes an objective function after applying the Euler-Lagrange to a linear partial differential equation containing a data and smoothness terms. Afterward, the problem is simplified by solving a sequence of large and structured linear systems of equations. Most of the optical flow algorithms including the variational algorithm assume small motion between two consecutive frames and use the Taylor expansion in the linearization step by neglecting the high order terms. Consequently, a coarse-to-fine scheme is used to deal with large motion resulting in the loss of motion information for image details and small objects. In this chapter, we demonstrate the feasibility of this problem and introduce a solution for recovering small motion details and using a lower number of levels for the coarse-to-fine optimization to achieve better accuracy in less execution time.

The organization of this chapter is as follows: section 3.1 introduces the large displacement optical flow problem, while section 3.2 presents some of related work. Section 3.3 explains the proposed optical flow model and represents the image details recovering module by using descriptor matching. The experimental results with synthetic and real sequences, including a comparison with classical and state-of-the-art methods are shown in section 3.4. Finally, conclusions and future works are given in section 3.5.

3.1 Large Displacements Optical Flow Problem

The use of the first order Taylor expansion approximation Eq. (2.15) in the optical flow estimation restricts the capability to estimate fast movements of

objects or the camera. Therefore, if the motion between two consecutive frames is notable (i.e. more than one pixel), the high order terms in Eq.(2.15) will have a significant influence on the estimated optical flow. Consequently, the estimation of optical flow using a differential optical flow approach fails, when the camera or objects move fast. Hence, a coarse-to-fine scheme [BBPW04] is proposed to overcome such a problem of the cost of losing small details in the interpolation process between different levels where initial values are propagated from a coarse optical flow level to a finer one. Furthermore, the number of levels required for the coarse-to-fine technique is an essential factor influencing the accuracy and the performance of the algorithm. Accordingly, the more significant number of levels is the more accurate results, but high processing power.
As described in chapter 2, the most widely known methods of differential optical flow estimation were developed by K.P. Horn and B.G. Schunck [HG81] as well as B. D. Lucas and T. Kanade [LK81]. The Lukas-Kanade method is a local operation assuming that small groups of pixels are moving together and having the same optical flow. In turn, the Horn-Schunck method uses a global optimization using a variational approach to the optical flow estimation. The Horn-Schunck method minimizes the following energy error function:

$$\mathcal{E} = \sum_{\Omega} \left[\left(I_1 \left(x, y \right) - I_2 \left(x + u, y + v \right) \right)^2 + \lambda \left(|\nabla u|^2 + |\nabla v|^2 \right) \right] \tag{3.1}$$

where $\mathcal{E}$ is an energy error function that has to be minimized and (u, v) are the displacement values in the x and y direction, respectively. $I_1(x, y)$ is the frame at time t while $I_2(x, y)$ is the consecutive frames at time $t + 1$. $\nabla = (\partial/\partial_x, \partial/\partial_y)^T$ is first order derivative which called gradient vector as well while λ is a regularization parameter. Ω is the $2D$ image domain. An optimal solution of Eq. (3.1) is calculated using the Euler-Lagrange optimization and the least square minimization within an iterative scheme [HG81].
Alternatively, [Cha04] proposed to use the L_1 total variation minimization [Cha04] which is formulated as:

$$\mathcal{E}_{TV-L1} = \sum_{\Omega} \left[\lambda |I_1 \left(x, y \right) - I_2 \left(x + u, y + v \right)| + \left(||\nabla u|| + ||\nabla v|| \right) \right] \tag{3.2}$$

Here, the data and the smoothness terms represent the isotropic total variation. Eq. (3.2) is decomposed into three parts [Cha04] as follows:

$$\mathcal{E}_{TV-1} = \sum_{\Omega} \left[\lambda |I_1(x,y) - I_2(x+u, y+v)| + \frac{1}{2\theta}(u-\hat{u})^2 + \frac{1}{2\theta}(v-\hat{v})^2 \right] \tag{3.3}$$

$$\mathcal{E}_{TV-u} = \sum_{\Omega} \left[\frac{1}{2\theta}(u-\hat{u})^2 + \|\nabla u\| \right] \tag{3.4}$$

$$\mathcal{E}_{TV-v} = \sum_{\Omega} \left[\frac{1}{2\theta}(v-\hat{v})^2 + \|\nabla v\| \right] \tag{3.5}$$

To solve the three equations above, [Cha04] proposed a numerical scheme. The original solution was developed to solve an image denoising problem and is subjected to an optimization of a convex function. Most of the optical flow approaches approximate $[(I_1(x,y) - I_2(x+u, y+v)]$ by using the Taylor expansion as in Eq.(3.6) and ignoring all terms of order higher than two .

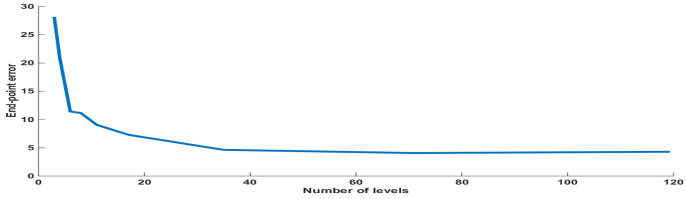
$$I_1(x,y) - I_2(x+u, y+v) = u\frac{\partial I_1}{\partial_x} + v\frac{\partial I_1}{\partial_y} + I_t(x,y) + HOT \tag{3.6}$$

Where $I_t(x,y) = I_1(x,y) - I_2(x,y)$ and HOT is the terms higher than two. Ignoring the high order terms yields the following linear equation:

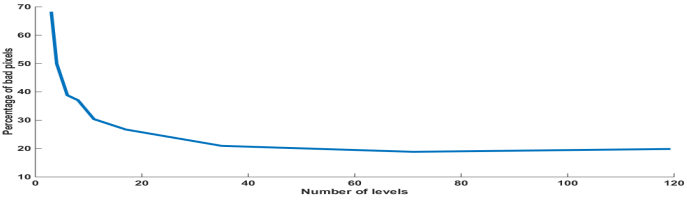
$$I_1(x,y) - I_2(x+u, y+v) = I_t + I_x u + I_y v \tag{3.7}$$

where $I_x = \frac{\partial I_1}{\partial_x}$ and $I_y = \frac{\partial I_1}{\partial_y}$ are the image derivatives in the x and y direction. As a result, the solution of the data-term Eq. (3.7) is only valid for small displacement motion (i.e., one pixel). In turn, high-order Taylor expansion terms can be used to deal with larger displacements resulting in a highly complex system which is difficult to be optimized. A coarse-to-fine scheme [BBPW04] solves the optical flow of different resolutions in a pyramid style and propagates the flow among different levels of the pyramid through an interpolation process. The pyramid is constructed using a pyramid factor which define the number of levels. The interpolation of the optical flow across different levels of the pyramid causes the loss of motion details such as small objects and the motion of image texture. Accordingly, large pyramid factor can be chosen to overcome this problem by increasing the number of levels, ending with high processing cost. Figure 3.1

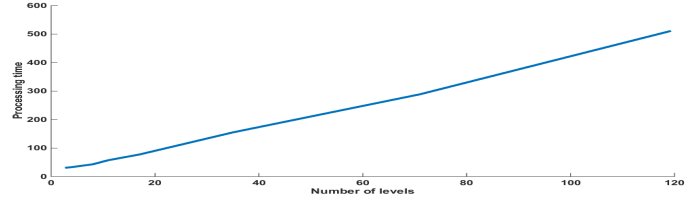
shows the effect of increasing the level numbers on the end-point errors figure 3.1a, the percentage of outliers figure 3.1b and the processing time (in *second*) figure 3.1c after applying the approach [BBPW04]. The larger number of levels is the lower average errors, but high execution cost is required.



(a) End-point error against number of levels



(b) Percentage of outliers against number of levels



(c) Processing time against number of levels

Figure 3.1: The effect of the number of levels on the accuracy and processing time after applying the variation optical flow approach [BBPW04]. (a) End-point error. (b) Percentage of outliers. (c) Processing time in *second*.

3.2 Related Work

Since the work of Horn and Schanck [HG81], research has concentrated on relieving the drawbacks of this method. Therefore, a series of improvements were proposed over the years. To handle large displacements, Bruhn et al. [BWS05] proposed a variation approach named CLG which combines local and global optical flow methods. However, the CLG approach produces wrong optical flow at motion boundaries as well as the motion of small image texture. Liu et al. [LYT11] proposed the usage of SIFT to estimate a dense optical flow field and used a discrete optimization algorithm. However, to introduce the regularity constraint, [LYT11] considers all the possible matches for the SIFT correspondents which requires heavy calculation cost. Moreover, Drulea et al. [DN11] proposed a parallel numerical scheme using the CLG [BWS05] and integrated the total variation L_1 norm. Furthermore, this approach replaced the Gaussian filter in the data-term with a bilateral filter and a diffusion mask to limit the propagation of the optical flow among the adjacent pixels.

An optimization approach in [XJM12] is used to reduce the dependency of the flow on their initial values which are propagated from the coarser to the finer in order to recover motion details in each scale. This approach refines the flow initialization at each level by integrating matching of SIFT [BSGF10] features at the cost of heavy processing time for the fusion steps. In turn, Brox et al. [BM11] proposed a solution to estimate large motions with small structures by integrating matched correspondences in a variational approach. They showed that an integration of a descriptor matching term in the variational approach allows handling better large displacements flow [WRHS13a]. This approach combines the ability to estimate arbitrarily large displacement which can be achieved using region-based descriptor matching with the strengths of variational optical flow methods.

Leordeanu et al. [LZS13] proposed an alternative approach to the coarse-to-fine scheme. This approach uses a total variation approach and refines the flow using a sparse matching with locally affine constraint for dense matching. Weinzaepfel et al. [WRHS13a] integrated a variational approach with a matching algorithm for optical flow estimation and proposed a descriptor matching algorithm

resulting in large displacements. The matching algorithm expands upon a multi-scale architecture with six levels, interleaving convolutions, and max-pooling. Consequently, the integration of dense flow into an energy minimization framework for optical flow estimation allows to retrieving correspondences and smoothing effect on descriptors matches.

[XT13] propose a non-iterative multi-resolution motion estimation strategy which involve block-based comparisons in each band of a Laplacian pyramid and combined the matching scores across resolutions. However, block-based matching results in high processing time and outliers. In turn, [WRHS13b] proposed a descriptor matching algorithm to the large displacement optical flow problem. The authors claim that the proposed approach allows to boost performance on fast motions as it builds upon a multi-stage architecture with a deep neural network with 6 layers. In their new work the authors [RWHS15a] introduced a matching algorithm using a deep multi layer convolution architecture. However the accuracy of the deep-learning based optical flow estimation algorithms is highly depending on the training data. Hence, there are not too much label training data that can be used for optical flow estimation, the accuracy is declined. Furthermore, [RWHS15b] proposed an approach for optical flow estimation dealing with large displacements with occlusions. The authors integrated dense matching using edge-preserving interpolation from a sparse set of matches with a variational energy minimization. [BYJ14a] proposed a fast randomized edge-preserving algorithm which approximate the nearest neighbor field. However, all the method mentioned above are facing the problem of providing accurate as well as fast processing at the same time. In the result section we show details comparisons with these methods.

3.3 The Proposed Approach

The CLG based models [BWS05, DN11] model used the squared L_2 norm which results in many drawbacks such as high sensitivity to noise and over-propagation. Furthermore, They does not preserve motion boundaries where it applied an anisotropic filter on the image gradient which blurs the motion boundaries.

Despite, the major drawback of relying on matching descriptors such as [XJM12, LYT11, BM11, LZS13, WRHS13a] to handle massive displacement optical flow is that local descriptors are not reliable at the locations of non-rigid objects. Besides, feature matching has low precision and can produce false or ambiguous matches. Moreover, integrating a matching component in the variational approach affect the formulation of the energy function as it could break down the performance at small displacement locations.

The coarse-to-fine scheme aims at finding a coarse level on which the motion is small (i.e., one pixel) and estimates the optical flow at that level. Hence, on the higher levels the results are more accurate. In fact, at the coarsest level, the motion is small, usually close to one pixel. For optimizing the energy function which is in a differential equation format, an initial solution is required. Therefore, in the original coarse-to-fine scheme [BBPW04], the initial solution for the coarser level customarily is considered to be equal to zero, while the interpolated optical flow at the previous coarse level is used as an initial solution for the next fine level. The proposed solution in this thesis is finding an appropriate initial solution at fine levels can save the cost of estimating the flow at coarse levels; hence the number of required levels can be decreased and a better accuracy achieved. In Figure (3.2) we introduces the proposed algorithm which uses a new method for doing initialization of the solution of the objective function without use of the coarse levels.

The proposed method utilizes the advantages of points matching to obtain an initial solution at the fine levels of the coarse-to-fine optimization scheme. Afterwards, an energy function is minimized using the proposed initial solution. As a result, the effect of the interpolation step in the coarse-to-fine model is significantly degraded, and the overall accuracy of large and small displacements is improved. Moreover, small image details are preserved, and the performance of the estimated optical flow is significantly increased since the number of levels for the coarse-to-fine scheme is reduced.

The proposed algorithm combines the total variational approach using the CLG with an image details recovering module based on point matching. Accordingly, the missing image details during the interpolation between different scales of the coarse-to-fine levels can be recovered.

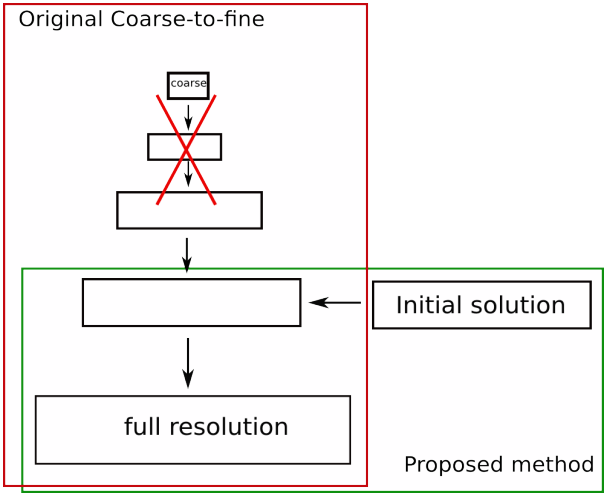


Figure 3.2: The proposed coarse-to-fine approach. The coarse levels are replaced with the results from the initialization step.

3.3.1 Image Details Recovering Module

Extracting and recovering image details requires a descriptor that provides a rich texture density in areas with complex structural information. Therefore, this module needs a robust descriptor. For that purpose, many descriptors can be used such as SIFT [Low04], SURF [BTVG06], BRIEF [CLSF10], ORB [RRKB11], BRISK [LCS11], LIOP [WFW11], MRRID [FWH12], census [ZW94] and MCT [FE04]. However, the proposed optical flow method in this chapter is not restricted to a specific descriptor. The evaluation of these descriptors is out of the scope of this thesis, and the reader is directed to read an evaluation of these descriptors in [MM12a]. For instance, the modified census transform MCT [FE04] (explained in section 2.10.3) has been selected as an example in this work due to its simplicity, its robustness concerning outliers, its allowance of a large extent of displacement vector lengths and its computational efficiency.

Features Matching

The purpose of the image details recovering module is to obtain a set of promising point to point correspondence hypotheses. These correspondences are accomplished by calculating a descriptor (i.e., the modified census transform operator MCT) on the two images and extract a signature vector for each feature in each image. Each vector has a fixed length and contains information about all the neighbor pixels. Hence, the matching can be performed using a basic indexing scheme. Consequently, the occurrence frequency c for each signature vector is calculated, and only the vectors which have $c < c_{max}$ are considered, to reduce the limits of the search area. In the proposed approach, we use a small value for the occurrence frequency which yields complexity of $O(c * n)$ where n is the number of pixels to get only robust features. We followed the algorithm in [Ste04a] for calculating the correspondences between two images. Figure 3.3 show the result after applying the matching algorithm on the "Army" sequence from the Midellburry dataset.

Figure 3.3: Feature points correspondences between two consecutive frames applied on the "Army" sequence from the Midellburry dataset. Here we used the modified census transform MCT as an example to calculate matching correspondences between feature points.

3.3.2 Optical Flow Model

The proposed algorithm was named TV-L1-MCT. It was published in [MM12b] and used in several methods published in [MRM+14], [MRM+13], [MMM15], [MMM16], and [MMM17]. In this section, we summarized briefly the proposed approach. It uses the CLG model in [DN11] to gain its robustenes against large displacement. Hence the CLG model is sensitive to outliers, we replaced the L_2 with the L_1 total variational. Moreover, we did not use the bilateral filter and the diffusion filter which proposed by [DN11] due their high processing time. The new objective function can be written as follows:

$$\mathcal{E} = \sum_{\Omega} \left[\psi \left(w^T J_\rho \left(\nabla_3 f \right) w \right) + \lambda_1 \left(\psi \left(\nabla u \right) + \psi \left(\nabla v \right) \right) + \lambda_2 \left((u - \hat{u})^2 + (v - \hat{v})^2 \right) \right] \tag{3.8}$$

where $\psi(x^2) = \sqrt{x^2 + \varepsilon^2}$ with $\varepsilon = 0.001$. $w = (\hat{u}, \hat{v}, 1)^T$ is the optical flow vector with the dual variational auxiliary variables $\hat{u}$ and $\hat{v}$. f is a convolved version of $I(x, y)$ with a Gaussian $K_\sigma(x, y)$ of standard deviation σ. $\nabla_3 f = (f_x, f_y, f_t)^T$. J_ρ is Gaussian $K_\rho(x, y)$ with a standard deviation ρ and $J_\rho(\nabla_3 f) = K_\rho * (\nabla_3 f \nabla_3 f^T)$. The term $\lambda_2((u - \hat{u})^2 + (v - \hat{v})^2)$ is the dual variational model used to enforce $(\hat{u}, \hat{v})$ and (u, v) to be equal. λ_1 and λ_2 are regularization parameters.
To solve the function in Eq. (3.8), we followed the solution suggested in [Cha04] by doing decompose of the function into three parts:

$$\mathcal{E}_M = \sum_{\Omega} \left[\psi \left(w^T J_p \left(\nabla f \right) w \right) + \lambda_2 \left((u - \hat{u})^2 + (v - \hat{v})^2 \right) \right] \tag{3.9}$$

$$\mathcal{E}_u = \sum_{\Omega} \left[\lambda_2 \left(u - \hat{u} \right)^2 + \lambda_1 \psi \left(\nabla u \right) \right] \tag{3.10}$$

$$\mathcal{E}_v = \sum_{\Omega} \left[\lambda_2 \left(v - \hat{v} \right)^2 + \lambda_1 \psi \left(\nabla v \right) \right] \tag{3.11}$$

To solve Eq. (3.9), assume u and v are constants and have initial values. Therefore, $\mathcal{E}_M$ has only two unknowns $\hat{u}$, $\hat{v}$. The solution for $\hat{u}$, $\hat{v}$ does not depend on the spatial derivative of $\hat{u}$ and $\hat{v}$, hence the solution can be calculated pointwise by using the least square minimization $A\mathbf{x} = \mathbf{B}$.

$$A = \begin{bmatrix} 2\lambda\theta + \sum I_x^2 & \sum I_x I_y \\ \sum I_x I_y & 2\lambda\theta + \sum I_y^2 \end{bmatrix}. \tag{3.12}$$

and

$$B = \begin{bmatrix} \left(2\lambda\theta + \sum I_x^2\right)\hat{u} + \sum I_x \sum I_y \hat{v} - \sum I_x \sum I_t \\ \left(2\lambda\theta + \sum I_y^2\right)\hat{v} + \sum I_x \sum I_y \hat{u} - \sum I_y \sum I_t \end{bmatrix}. \tag{3.13}$$

Similarly, equations $\mathcal{E}_u$ and $\mathcal{E}_v$ have two unknowns u and v, while the variables $\hat{u}, \hat{v}$ are known and constants from above. The numerical scheme in [DN11] can be used to optimize $\mathcal{E}_u$ and $\mathcal{E}_v$ and solve for the unknowns. The Euler-Lagrange equation for $\mathcal{E}_u$ is:

$$(u - \hat{u}) - \lambda div\left[\frac{(\nabla u)}{\psi(\nabla u)}\right] = 0 \tag{3.14}$$

If $P_u = \nabla u/\psi(\nabla u)$, then the Eq. (3.14) can be written as:

$$u = \lambda div(P_u) + \hat{u}, \tag{3.15}$$

The above equation can be solved iteratively using the fixed-point iteration scheme as illustrated in [DN11].

$$P_u^{n+1} = \frac{P_u^n + \tau\nabla(div(P_u^n) + \frac{\hat{u}}{\lambda})}{1 + \tau||\nabla(div(P_u^n) + \frac{\hat{u}}{\lambda})||}, \tag{3.16}$$

where τ is a time step, experimentally ($\tau \leq \frac{1}{8}$). The same scheme can be applied to optimize the function P_v.

As mention above the CLG model fails to calculate correct optical flow at the motion boundaries because the objective function Eq. (3.8) is an isotropic function which propagates the flow in all directions regardless of local properties which causes motion blur on the boundaries. Therefore, in order to reduce the effect of the propagation, a weighted median filter can be applied as recommended in [BCM05, GO09]. However, in the proposed model [MM12b], applying a median filter can eliminate not in all cases the recovered image details. Hence, to handle this problem, the algorithm in [SRB10] is combined with the spatial-temporal image segmentation approach introduced in [RPG11] to calculate the weighted function based on the image texture as follows:

$$\hat{u}_{i,j} = \min \sum \omega_{i,j,i',j'} |\hat{u}_{i,j} - u_{i',j'}| \tag{3.17}$$

where $(i', j') \in N_{i,j} \cup \{i, j\}$ which $N_{i,j}$ is the $N \times N$ local window, and $\omega \in [0, 1]$. The approach in [RPG11] segments the image into three different regions (texture

moving, homogeneous moving and static regions) based on the spatial and the temporal image derivatives.

$$I(x,y,t) \in \begin{cases} Texture - Moving & SNR \leqslant \tau, |\cos(\delta)| \simeq 1, |\cos(\beta)| \simeq 0 \\ Homogenous - Moving & SNR > \tau, ||\cos(\delta) \simeq 1 \\ Stationary & Otherwise \end{cases} \tag{3.18}$$

where SNR is the signal-to-noise ratio of the gradient's magnitude and δ is the angle between the spatiotemporal image gradient $(I_x, I_y, I_t)^T$ and a unit vector $(0,0,1)^T$. In the case that the image gradient is minimal, $|\cos(\beta)|$ is set to one. To illustrate the proposed algorithm: if two feature points belong to the same type of region (homogeneous or textured), but the states are different, i.e., one is a moving region and the other is a static region, formerly $\omega = 0$. Likewise, if a feature point belongs to a textured region while a neighboring point belongs to a homogeneous region, the propagation does not affect that pixel and thus $\omega = 0$, otherwise $\omega = 1$.
The proposed algorithm works as follows: at each pyramid level of a coarse-to-fine scheme, the solution of the Eq. (3.15), and Eq. (3.16) are calculated iteratively. Afterward, the initial values of u and v are propagated from each coarser level into the finer one. The matching correspondences algorithm provides a set of hypothesis points. These points used to refine the propagated values from the coarser level and provide an initial solution for the optical flow at the fine levels. A weighted median filter is applied every level to reduce the outliers.

Figure 3.4 shows the optical flow results applied on the "Army" sequence of Middlebury dataset. We have used a coarse-to-fine scheme with pyramid factor equal to 0.5 to estimated the optical flow. It can be seen that using the initialization scheme the fine details of the motion are significantly improve (see figure 3.4e and 3.4f).

Until that point, we have two initial sources of matching: (a) matching correspondences which neglect regularity and (b) propagated values from the coarser level which neglect image details. The proposed algorithm combines both sources of information to estimate the optical flow at each level. In case that there is no matching correspondence, only the propagated value from the coarser level

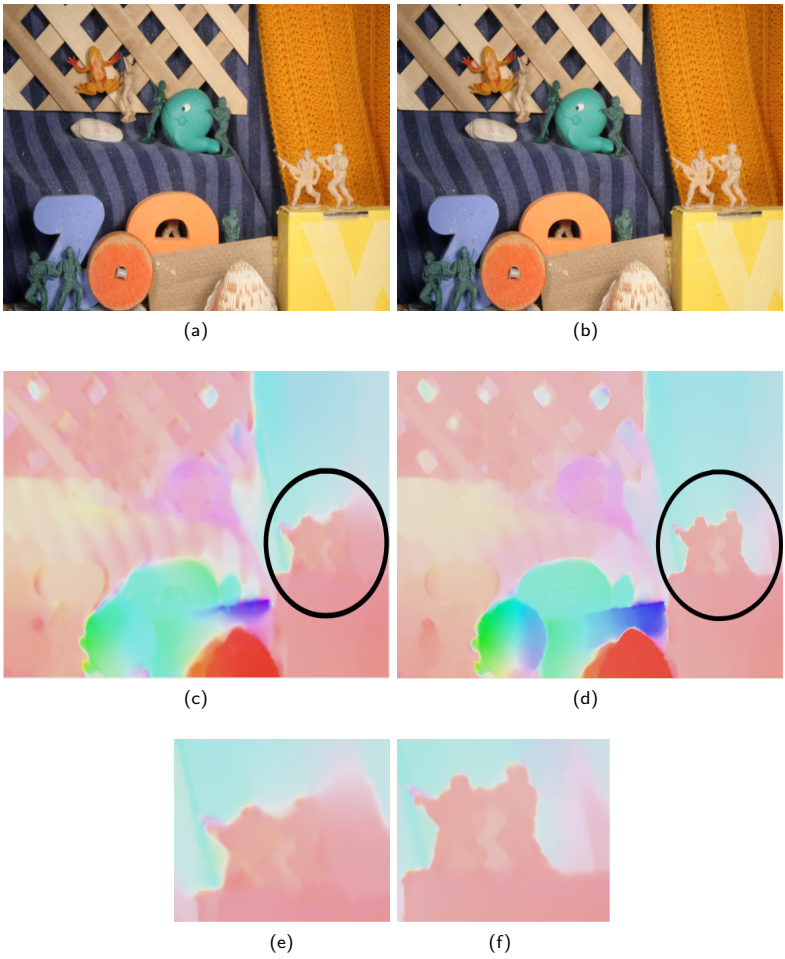


Figure 3.4: Optical flow results applied on the "Army" sequence from Middlebury dataset. (a) and (b) show $frame10$ and $frame11$ from "Army" sequence. (c) The estimated optical flow using the original coarse-to-fine with a pyramid factor equal to 0.5 (6 levels). (d) The estimated optical flow after applying the proposed coarse-to-fine algorithm using only 3 levels. (e) Part of the optical flow using the original coarse-to-fine. (f) Part of the optical flow using the proposed algorithm.

is considered. Otherwise, a fusing function is used to verify the motion vector values based on the analyzing of the neighborhood pixels. This decision is done via comparing the mean value of the vector lengths $\bar{d}_p$ of the propagated $(N \times N)$ window values and the vector length d_c of the matching correspondences. Hence, d_c is assumed to be an outlier in case that the difference between d_c and $\bar{d}_p$ is more significant than a threshold and then only the propagated value is considered. In turn, if the propagated value is similar to the neighbor pixels, while its location is not homogeneous, then its probably, the motion information has been lost in the interpolation process. In such case, we consider only the matching correspondences. The initial values for the first optical level are set to the values of matching correspondences because the initial values of the propagated optical flow are set to zero.

The overall approach of the optical flow estimation is illustrated in figure 3.5. For simplicity, we used a coarse-to-fine scheme with only 3 levels. At each level we calculate the matching between 200 feature points and integrate them with the dense optical flow. It can be seen that the quality of optical flow is improved from one level to the next.

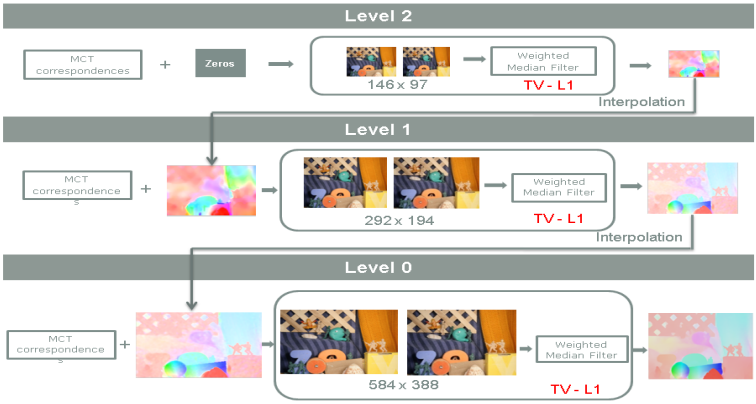


Figure 3.5: The flowchart of the proposed coarse-to-fine approach. At each level, the point correspondence has been used to provide an initial solution to solve the objective function at each pixel.

3.4 Evaluation and Experimental Results

To evaluate the proposed approach, we have used Middlebury optical flow benchmark [BSL+11] which contains two different datasets for the evaluation of optical flow. The first dataset called "training" and it contains eight sequences with known ground-truth. The second dataset called "test", and it contains twelve testing sequences without known ground-truth. In the following section, we introduce the evaluation of the proposed algorithm based on the two given datasets and present comparisons with stat-of-the-art methods.

3.4.1 Middlebury Training Dataset

We have evaluated the proposed approach on the training database from Middlebury using the average end-point error (AEE) and the average angular error(AAE). The results are shown in table 3.1. We compared the proposed method with the baseline method [DN11] which uses the CLG model with L_2 norm. Based on the same training data, we have chosen the parameters for the algorithms which give the best accuracy in most cases, see [MM12b] for more detail. As shown in table 3.1, the proposed method significantly outperform the baseline method in all cases, especially in the sequences which have large displacement such as Urban2 and Urban3.

Table 3.1: The average angular error AAE and the end-point error AEE of the proposed approach applied on the training dataset from Middlebury benchmark compared with the baseline method CLG-TV [DN11].

	Error	Venus	Dimetrod	Hydrangea	Rubber	Grove2	Grove3	Urban2	Urban3
CLG-TV	AAE	7.060	3.951	2.271	3.332	2.844	7.756	3.407	13.302
[DN11]	AEE	0.500	0.195	0.187	0.103	0.217	0.894	0.551	1.501
The proposed approach	AAE	3.320	2.013	1.834	2.472	1.440	5.492	2.071	2.997
	AEE	0.238	0.103	0.156	0.078	0.099	0.509	0.223	0.407

3.4.2 Middlebury Test Dataset

We have used the on-line evaluation service provided by Middlebury to evaluated the proposed algorithm on the test dataset which does not have known ground truth. At the time of submission, the proposed approach has been ranked on as 12^{th} position for angular error and 12^{th} for the end-point error out of 120 method. The proposed algorithm out-performed most of the approaches dealing with large displacements optical flow such as [XT13], [DHW13], [WRHS13b] , [RWHS15b], [ADB14], [RWHS15a], [BYJ14a], and [LYMD13]. Table 3.2 shows a comparison among the proposed method and the mentioned methods. It can be seen that the proposed algorithm outperformed these method in most of the sequences and provided the best rank. In the following sections we provide details comparisons of the proposed approach with these methods based on motion discontinuities, interpolation/normalized interpolation errors, and percentage of outliers.

Table 3.2: Middlebury on-line comparison of the AEE among the proposed method and the state-of-the art algorithms which are dealing with large displacement optical flow.

AEE all	Rank	Army	Mequon	Schefflera	Wooden	Grove	Urban	Yosemite	Teddy
2bit-BM-tele [XT13]	101.7	0.21	0.39	0.60	0.38	1.01	1.39	0.31	1.11
DeepFlow [WRHS13b]	65.3	0.12	0.28	0.44	0.26	0.81	0.38	**0.11**	0.93
EpicFlow [RWHS15b]	59.1	0.12	0.25	0.39	0.19	0.89	0.53	0.10	0.67
DeepFlow2 [RWHS15a]	55.6	0.10	0.25	0.40	0.21	0.80	**0.36**	**0.11**	0.82
EPPM w/o HM [BYJ14a]	42.9	0.11	**0.19**	**0.29**	0.17	**0.63**	0.60	0.19	**0.45**
The proposed approach	**36.2**[1]	**0.08**[1]	0.24[2]	0.32[2]	**0.14**[1]	0.72[2]	0.54[4]	**0.11**[1]	0.54[2]

Motion Discontinuities

The idea of the proposed algorithm is to provide a solution for the problem of the large displacement as well as small displacement optical flow. Therefore, in this experiment, we test the accuracy of the motion details and motion boundaries. Table 3.3 shows a comparison between the proposed method and the state-of-the-art methods mentioned above. As shown in table 3.3, the proposed method provides the minimum AEE error at motion boundaries for most of the sequences among all state-of-the-art methods. figure 3.6 shows qualitative results of some

sequences of Middlebury dataset. As can be seen, the optical flow errors at the boundaries and the image textures are significantly reduced.

Table 3.3: The average endpoint error of the discontinuities AEE disc on Middlebury evaluation.

AEE disc	Rank	Army	Mequon	Schefflera	Wooden	Grove	Urban	Yosemite	Teddy
2bit-BM-tele [XT13]	101.7	0.42	1.04	1.30	1.49	1.41	1.68	0.23	2.09
DeepFlow [WRHS13b]	65.3	0.31	0.82	1.00	1.34	1.21	1.55	**0.11**	1.82
EpicFlow [RWHS15b]	59.1	0.36	0.85	1.00	1.01	1.31	1.31	**0.11**	1.43
DeepFlow2 [RWHS15a]	55.6	0.29	0.79	0.96	1.08	1.18	1.45	**0.11**	1.68
EPPM w/o HM [BYJ14a]	42.9	0.30	**0.67**	**0.71**	0.78	**0.93**	1.35	0.15	**0.94**
The proposed algorithm	**36.2**[1]	**0.23**[1]	0.77[2]	0.76[2]	**0.69**[1]	1.03[2]	**1.10**[1]	0.12[2]	1.04[2]

Figure 3.6 shows the quality of the motion boundary using the proposed algorithm applied on Middlebury dataset. It can be seen from the error images (see figure 3.6 col.4) that the proposed method succeeded to estimate correct motion at the boundaries.

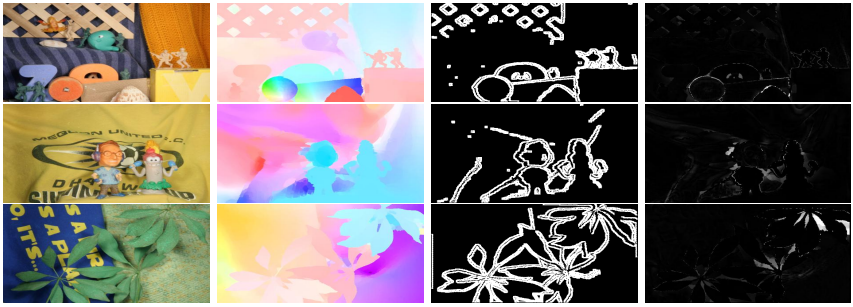


Figure 3.6: Motion boundaries. Col.1: sequences Army, Mequon, and Schefflera. Col.2: the estimated optical flow. Col.3: boundaries. Col.4: error.

Interpolation and Normalized Interpolation Error

In this experiment, we calculate the interpolation and the normalized interpolation errors.

Table 3.4: The interpolation error IE of some of the state-of-the-art methods on Middlebury benchmark.

IE	Rank	Mequon	Schefflera	Urban	Teddy	Backyard	Basketball	Dumptruck	Evergreen
DeepFlow2 [RWHS15a]	55.6	3.29	**3.79**	4.96	**5.08**	11.18	6.45	8.11	7.68
DeepFlow [WRHS13b]	65.3	3.31	3.82	5.00	5.34	11.21	6.55	8.11	7.82
2bit-BM-tele [XT13]	96.10	3.31	4.53	6.23	5.94	11.30	7.72	12.2	7.76
EPPM w/o HM [BYJ14a]	76.50	3.35	3.85	7.03	6.15	**10.60**	7.00	8.85	8.42
EpicFlow [RWHS15b]	61.30	**3.17**	**3.79**	**4.28**	6.37	11.20	6.23	8.11	8.76
The proposed algorithm	**50.60**[1]	3.17[1]	3.87[4]	4.48[2]	5.37[3]	11.60[6]	**6.08**[1]	**8.07**[1]	**7.68**[1]

Table 3.5: The normalized interpolation error NE of some of the state-of-the-art methods on Middlebury benchmark.

NE	Rank	Mequon	Schefflera	Urban	Teddy	Backyard	Basketball	Dumptruck	Evergreen
DeepFlow2 [RWHS15a]	55.6	0.29	0.79	0.96	1.08	1.18	1.45	**0.11**	1.68
DeepFlow [WRHS13b]	65.3	0.31	0.82	1.00	1.34	1.21	1.55	**0.11**	1.82
2bit-BM-tele [XT13]	101.5	0.70	0.87	2.82	1.13	1.16	1.59	1.90	**0.82**
EpicFlow [RWHS15b]	62.8	0.62	0.70	**1.06**	1.09	1.18	1.10	1.00	1.04
EPPM w/o HM [BYJ14a]	51.7	0.60	0.67	2.36	1.01	**1.00**	1.14	1.18	0.87
The proposed algorithm	**38.80**[1]	0.62[4]	0.71[4]	1.21[3]	**0.95**[1]	1.19[6]	1.70[7]	**0.71**[1]	**0.82**[1]

Figure 3.7 shows the interpolation IE and the normalized interpolation NE error of optical flow estimation using the proposed method applied on Middlebury dataset. It can be seen from the error images that the proposed method succeeded to estimate correct motion at the boundaries. However, the error images still contain errors at some points due to the limitation of the brightness constraint. A solution to replace the brightness constraint will be introduce in the next chapter.

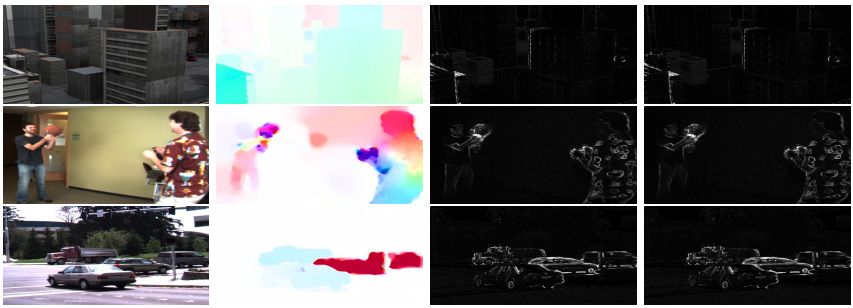


Figure 3.7: The IE and NE. Col.1: sequences Urban, Basketball, and Dumptruck. Col.2: the estimated optical flow. Col.3: IE. Col.4: NE.

Percentage of the Outliers

In this experiment, we evaluate the robustness of the proposed approach concerning the outliers. Therefore, we measure the percentage of points which have EE more than $0.5, 1$ and 2 pixels for each sequence in the Middlebury dataset. Table 3.6 shows a comparison with the baseline method [DN11]. As shown in table 3.6, the percentage of outliers is decreased significantly, and the proposed method outperforms the baseline method in the most sequence of the Middlebury dataset.

Table 3.6: The percentage of outliers for the proposed algorithm and the CLG-TV method [DN11] on the Middlebury dataset.

Outliers %EE $> \varepsilon$ Pixels	Rank	Army	Mequon	Schefflera	Wooden	Grove	Urban	Yosemite	Teddy
$\varepsilon > 0.5$ Proposed algorithm	**33.8**	**2.09%**	**9.67%**	**7.11%**	**2.94%**	**28.40%**	16.00%	**1.27%**	**10.80%**
$\varepsilon > 0.5$ CLG-TV [DN11]	57.2	2.80%	14.00%	12.70%	8.13%	32.00%	**14.10%**	6.45%	19.00%
$\varepsilon > 1.0$ Proposed algorithm	**37.1**	**0.90%**	**3.73%**	**4.61%**	**2.16%**	**17.60%**	11.00%	**0.36%**	**9.73%**
$\varepsilon > 1.0$ CLG-TV [DN11]	52.60	1.01%	4.16%	7.52%	3.33%	20.90%	**8.94%**	1.27%	11.10%
$\varepsilon > 2.0$ Proposed algorithm	**33.20**	**0.22%**	**1.64%**	**2.86%**	**1.14%**	**10.60%**	8.81%	**0.08%**	**5.93%**
$\varepsilon > 2.0$ CLG-TV [DN11]	50.30	0.31%	2.10%	5.33%	1.71%	19.90%	**5.20%**	0.20%	7.85%

Performance Evaluation

Figure (3.8) show the effect of the number of levels on the average end-point error and the percentage of outliers. As shown in figure (3.8), the more significant number of levels is the better accurate results and more considerable processing power as well. Applying the proposed approach to initialize the optical flow increases the accuracy and reduces the processing power. Hence, using only ten levels with initialization produces accuracy which is obtained, if we use 40 levels without initialization. It can be seen that if use only one level with initialization the percentage of outliers are decreased 10% and the AEE is decreased 10.

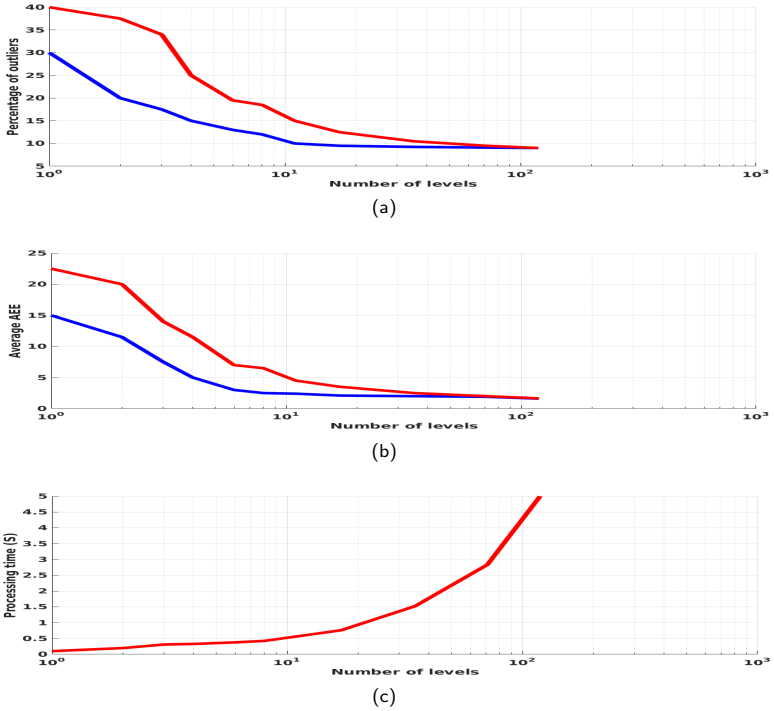


Figure 3.8: The effect of the number of levels on the errors after applying the refinement (blue) and without refinement (red) applied on all sequences of the KITTI dataset. (a) Percentage of outliers. (b) AEE. (c) Processing time in *Second*.

3.4.3 Large Displacements Optical Flow Dataset

We have evaluated the proposed approach on real and large-scale displacement sequences using the available data sets at Karlsruhe university[1] and Freiburg university[2]. figure 3.9 shows some results of our approach compared with the results after applying the original CLG-TV algorithm. Unfortunately, no ground truth for these data set is available. Therefore, we a compared the optical flow visually. It is clear from figure 3.9 that the proposed approach provide smooth flow and the motion boundaries are more precise than the baseline method.

3.4.4 Real Application

In this section, a real scenario is used to prove the robustness of the proposed algorithm. The proposed optical flow approach is used to obtain accurate optical flow that can be used by a robot to detect, localize and grasp moving objects. The optical flow algorithm is tested using real image sequences from the Lemgo model factory [MM12c]. Some images from the first image sequence are shown in figures 3.10 (a), (b) and (c). In this sequence, the camera is in a horizontal position on the top of moving objects. As shown in figure 3.10 (d), (e) and (f) the algorithm correctly estimated the motion of the object. The object moved from right to left, which is represented by the blue color in the color representation. The second sequence is shown in figure 3.10 (g), (h) and (i). In this sequence, the camera is portable and facing many moving objects in the scene. The objects were moving from left to right, The proposed algorithm succeeded to detect and estimate the motion of the objects, the motion is display by the red color based on the visualization scheme. The global color which appears in this sequence represents the global motion due to the camera motion which is also called ego-motion.

Figure 3.11 Shows another example of a fast moving object from left to right. The proposed algorithm correctly estimated the optical flow.

[1] $http://i21www.ira.uka.de/image\_sequences/$
[2] $http://lmb.informatik.unifreiburg.de/resources/datasets/sequences.en$

Figure 3.9: Optical flow results of the proposed approach compared with the original CLG-TV, the first row shows frames at time t from Ettlinger-Tor, MIT, tennis, and *marple*2 sequences, and the second row shows the frames at $(t+1)$. while the third row shows the estimated optical flow produced by using the original CLG-TV and the fourth row is our estimated optical flow.

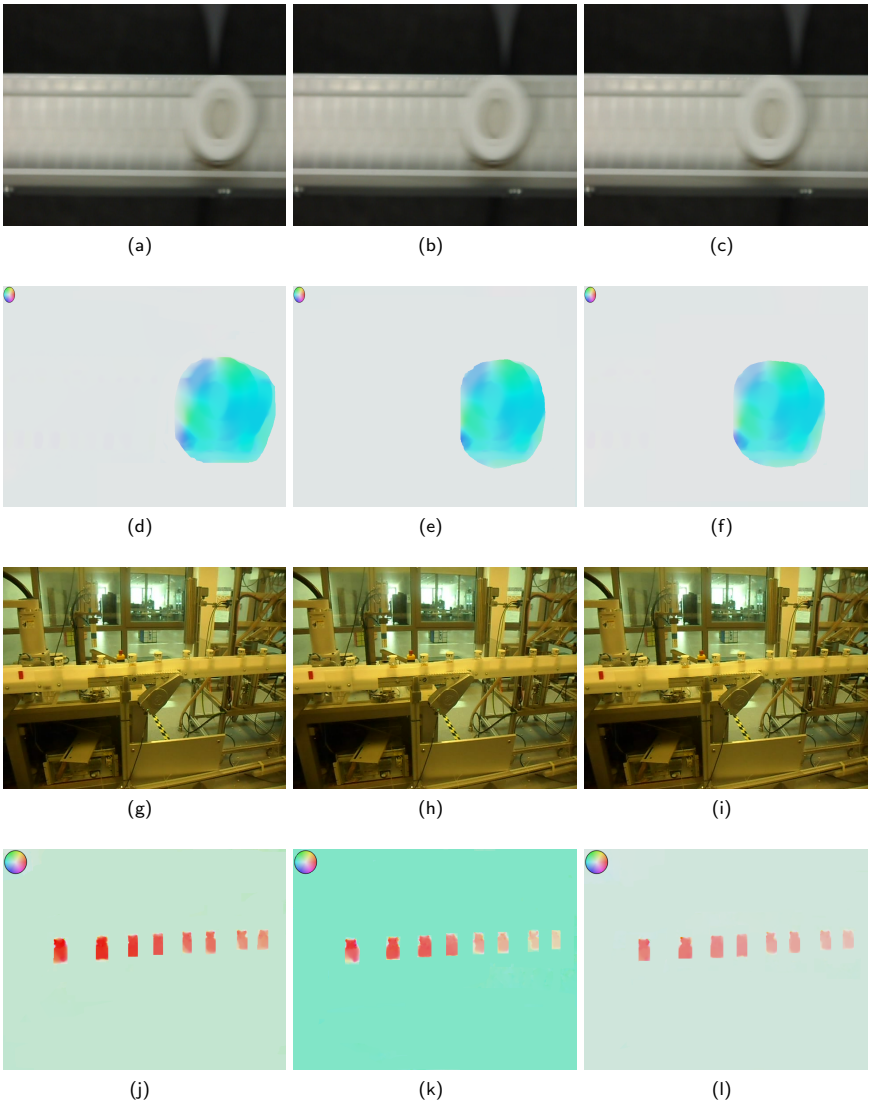


Figure 3.10: Results of optical flow estimation applied on a real application. Odd rows show the images while the even rows show the optical flow between each image and the next image in this sequences.

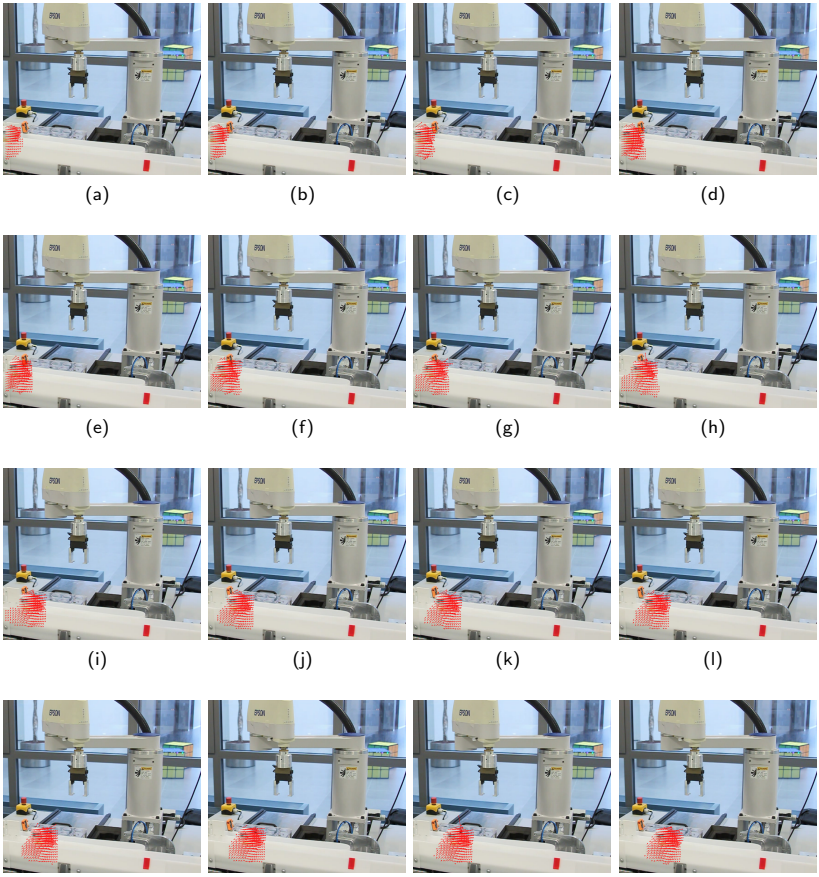


Figure 3.11: Results of optical flow estimation applied on a sequence of images of a fast moving object. The optical flow here is shown using arrows pointed at the direction of the motion.

3.5 Summary

In this chapter, we introduced a method for improving the estimation of optical flow in case of fast motion. Therefore, we combined the L_1 total variational and CLG optical flow approach integrating image details recovering module in a coarse-to-fine scheme. A local descriptor is used to calculate matching correspondences of feature points. Correspondingly, the proposed approach uses the matching correspondences output to build initial solutions for solving the variational minimization equation during the coarse-to-fine scheme. Besides, the possible matching correspondences are limited by considering only robust matching to improve the overall computational time. Equally, the matching correspondences have been used, are combined with a weighted median filter to recover the lost image information and preserve image details during the interpolation process to improve both large and small displacements motion. The proposed method has been tested on *Middlebury* benchmark, large displacement optical flow dataset, in addition to a real application. Admittedly, the new method gives competitive results in both the end-point and the average angular errors. It has been shown that the proposed algorithm decresed the processing time about 30% as well as increased the accuracy with about 40%. The proposed method has been successfully used in a real application.

4 Proposed Robust Optical Flow Estimation

In chapter 3, we considered the problem of large displacement optical flow, and we proposed an approach to improve the accuracy and performance of the coarse-to-fine scheme. The proposed method used the brightness constraint as a data conversation in a sequence of consecutive images and optimized an objective function based on the TV-L1 approach. In fact, in an environment with constant brightness, the optical flow can be accurately estimated using the brightness constancy assumption (BCA) or using a high order constancy such as Gradient. Conversely, once the illumination changes or objects move to another place with a different illumination condition (e. g. into the shadow of a tree), these assumptions are no longer valid.

In this chapter, we illustrate the feasibility of robust optical flow estimation in case of illumination change by introducing a local texture constraint which is more robust against illumination changes than others assumptions. Hence, we formulate an optimization model for integrating the image texture as a constraint to describe edges, gradients, or orientation of image features. Accordingly, we optimize an energy function that maximizes the similarity between images features. Moreover, several image texture descriptors have been integrated such as the histogram of oriented gradients (HOG),the modified local directional pattern (MLDP) or the local binary pattern (LBP) (census signature) and other descriptors.

The organization of the chapter is as follows: section 4.1 focuses on the illumination change problem and presents some of the state-of-the-art methods. Section 4.2 deals with the extraction of image texture and presents a new robust descriptor for describing image features. Moreover, it describes the texture constancy assumption and explains the optimization of an energy function based on the texture constraint using different kinds of texture descriptors. Section 4.3

contributes a solution to the integration of color information using different color spaces to increase the accuracy of the estimated optical flow. The evaluation and experiment results including comparisons with the state-of-the-art methods are given in section 4.4. Finally, section 4.5 highlights the conclusions.

4.1 Related Work

Most optical flow methods, such as [BBPW04, SRB10, XJM12, ZBW11, RPG12, MM12b, RGP13], have focused on estimating accurate flow fields under ideal conditions [BSL+11, GLSU13] rather than improving the robustness in realistic scenes under various conditions [MRM+14]. Furthermore, most variational optical flow approaches concerning the accuracy are mainly dependent on the brightness constancy assumption (BCA) or high-order constancy assumptions, such as gradient constancy assumption (GCA) [BBPW04, ZBW11, RPG12, RGP13]. However, the brightness of a pixel on an object may significantly change, if the object moves to another location with a different illumination condition in the scene. Moreover, the image gradient is sensitive to noise. In this manner, those assumptions become strongly limited in the case of illumination changes [MRM+14].
In the literature, many techniques have proposed different optical flow models that are robust toward illumination changes. For instance, [KMK05] proposed a robust energy function that takes into account multiplicative and additive illumination factors as well as optical flow unknowns. However, this integration affects the accuracy of the optical flow estimation negatively. Moreover, the optimization of such complex energy function becomes troublesome as it deals with motion estimation and illumination variations concurrently. In turn, [MBW07] proposed an algorithm appropriate only for color images with brightness variations by using photometric invariants of the dichromatic reflection model.
Recently, local image descriptors have been introduced to estimate the optical flow. E.g., Stein et al. [Ste04a] used the census transform and produced a sparse optical flow estimation model. Moreover, [MRR+11] proposed an illumination-invariant objective function that uses a total duality variation with the L1 norm (TV-L1) after utilizing the Hamming distance between two descriptors using the census transform. In turn, Ranftl et al. [RGPB12] proposed a variational model for stereo

images that integrates an objective function applying the census transform and a smoothing term using the total generalized variation (TGV) proposed in [BKP10]. Accordingly, to avoid thresholding for the census transform, Vogel et al. [VRS13] proposed a data-term that uses the sum of the centralized absolute differences CSAD. Hence, to avoid the artifacts produced by the TV-L1, the proposed data term [VRS13] uses the total generalized variation regularizer. Nevertheless, the census transform is sensitive to non-monotonic illumination variation and random noise. Furthermore, census transform cannot distinguish between dark and bright regions in a neighborhood, and it discards essential information casting from neighbors. Consequently, Oliver et al. [DHW13] proposed a data-term using the complete rank transform and used the TV-L1 optical flow model. The complete rank transform is a modified version of the census transform, and it encodes the intensity rank of every pixel in a neighborhood. In addition, Ali et al. [ADB14] proposed a method to consider the structure information of the image to estimate the optical flow. To summarise, Vogel et al. [VRS13] evaluated several data costs such as the census transform, ternary census transform, normalized cross-correlation, and mutual information.

Discrete optimization has been successfully used to estimate optical flow that works robustly in case of illumination changes as well as large displacement motion. For instance, Liu et al. [LYT11] proposed the SIFT flow, which estimates a dense correspondence field between two frames using the SIFT descriptor [Low04] and a belief propagation approach. The SIFT flow calculates a matching of features and yields a pixel accurate optical flow. In turn, Brox et al. [BM11] proposed a continuous variational energy function which integrates discrete pixel matching using a HOG/SIFT-like descriptor with a data-term using the brightness and the gradient constancy assumptions. In addition, József [MCF10] proposed an approach that uses the normalized cross-correlation. The resulted data term leads to increasing the robustness against multiplicative illumination changes.

To tackle the problems of poorly textured regions, occlusions, and small-scale image structure, [WPB10] incorporated a low-level image segmentation based on a non-local total variation regularization in a unified variational framework. Furthermore, Werlberger et al. [WPB10] proposed a truncated normalized cross-correlation that is robust against illumination changes. Moreover, Drulea et al. [DN13] proposed a method based on the zero-mean normalized cross-correlation

transform. Color information can be used to improve the accuracy of optical flow estimation. For instance, Zimmer et al. [ZBW11] presented a data-term that is robust against outliers and varying illumination conditions based on the HSV color space and a normalization constraint. This model uses a gradient constancy assumption and integrates an isotropic smoothness term that works complementary to the data-term. In turn, Sun et al. [SRB10] proposed a weighted non-local term that uses a color similarity together with an occlusion state of pixels and integrates flow estimates over large spatial neighborhoods.
Multi-constraints such as the image gradient constancy and the brightness constancy has been integrated into a single objective function. For instance, the algorithm [RGP13] uses a discontinuity-preserving filtering stage based on a stick tensor voting which is robust against noise and illumination changes. However, the gradient constancy assumption does not work correctly with massive illumination changes.
The contribution of this chapter is developing an optimization framework based on a texture constraint, which has provided to be robust in the context of illumination changes and large displacements. Hence, we illustrate an adapted variational optical flow model which integrates several types of texture descriptors (i.e., HOG, LDP, Census,..., and so forth.). Accordingly, the proposed algorithm integrates other constraints such as color and epipolar constraints. To our knowledge, this is the first attempt to integrate a residual function based on the texture features directly in the differential optical flow model. Moreover, we propose the use of a novel descriptor namely the modified local directional pattern MLDP [MRM$^+$14] which outperforms other local descriptors in the calculation of the optical flow.

4.2 Texture Constancy Assumption

A texture constraint assumes that the connection between neighborhood pixels (e. g. edges, gradients, curves,..., and so forth) stays constant if an object or the camera moves, while the brightness might vary. As mentioned in chapter 2, there exist several local descriptors that can be utilized to describe a texture such as SIFT, SURF, Census, LDP, HOG, DAG, and other descriptors. On the one hand, extended descriptors of the feature matching methods such as

SIFT and SURF are solutions to achieving high correct matching rates. However, in the case of dense matching based on differential techniques, the use of such extended descriptors increase the high computation cost. Considerably, compact versions of the descriptors such as Census and its modified versions, LDP and its modified versions, DAG and HOG mostly with a lengths of one byte have been used for dense optical flow calculations. In this chapter, we propose a new compact descriptor called the MLDP. The extracted features are then utilized in a data-term to formulate a new texture constancy assumption.

4.2.1 Modified Local Directional Pattern (MLDP)

As mentioned in chapter 2, the local direction pattern does not encode the direction of edges. Therefore, it is not a reliable choice for optical flow estimation. Therefore, we present here a new modification for the LDP descriptor called, Modified Local Directional Pattern, which has been published in [MRM+14]. The proposed MLDP, which depends on the edge responses and their signs (directions), but does not encode the edge magnitude, yields a very robust descriptor against illumination changes. In this chapter, a brief explanation of this descriptor is given. For more details, the reader is directed to the articles [MRM+14] and [MRM+13]. Likewise LDP, MLDP encodes eight edge responses relating to eight mask operations. However, the resulting descriptor is in the form of 8-bit string encoding the directions of edges by setting 1 to the corresponding bits of the positive edge responses and 0 to the negative responses. The MLDP response is written as:

$$s = \begin{cases} 1 & \mathcal{ER} > 0 \\ 0 & \text{otherwise.} \end{cases} \tag{4.1}$$

where s is a bit in the MLDP string $\mathcal{S}$ and $\mathcal{ER}$ is the edge response at the current pixel. The descriptor is robust to illumination changes as it depends on the edge responses instead of the difference between intensity values (i.e., census transform). To calculate the edge response $\mathcal{ER}$, we used a compass mask proposed in [JKC10] which yields a noise-robust descriptor. Every pixel intensity is replaced by an 8-bit string (see figure 4.1). After applying MLDP to a gray

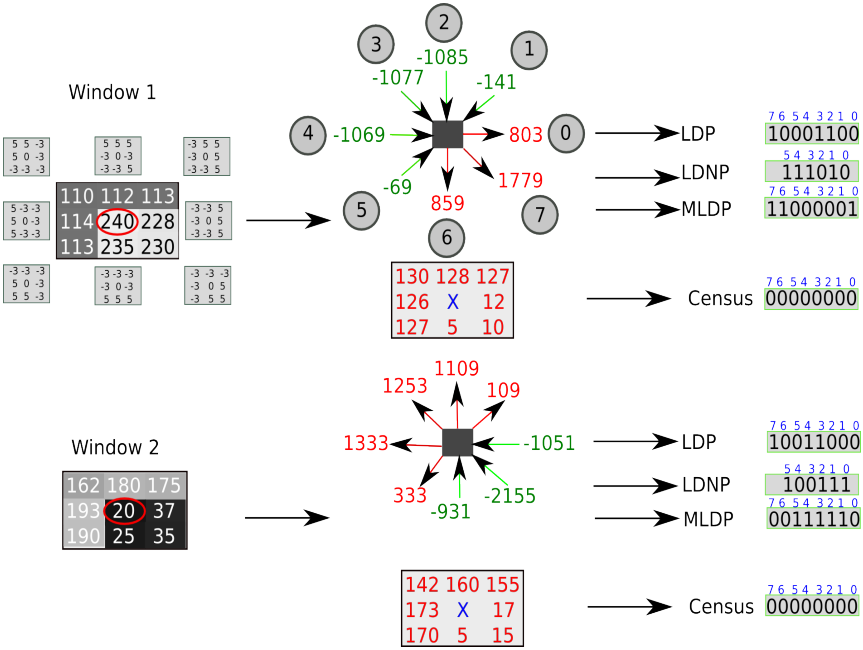


Figure 4.1: Different versions of local descriptors such as LDP, LDNP, MLDP and the census transform for two examples (from [MRM+14] with permission from IEEE).

image, the result is an 8 bit binary image with eight channels, with each channel representing a response of a directional mask operation. For determining the direction responses at a multi-scale, the Kirsch masks [JKC10] are applied for computing the eight edge responses for each pixel at each scale. Alternatively, the shifted derivative of the Gaussian filter G'_σ can be utilized to calculate the edge responses. Here, no convolution with the Gaussian filter G_σ is needed, since it is already applied at every scale level in the pyramid scheme during rescaling as described in [BBPW04].

4.2.2 Optical Flow Model for the Texture Constraint

In the literature, the usage of a texture descriptor to estimate optical flow is usually done by applying a discrete optimization scheme or by using a matching procedure. Conversely, in this chapter, an explanation of how to develop an objective function that uses the similarity of texture descriptors for the estimation of optical flow [MRM+14]. Consequently, the new formulation makes it easy to use several types of local descriptors and at the same time to use several kinds of smoothness constraints which was not feasible before. In the following paragraphs, an explanation of how to construct an objective function based on the TV-L1 model in Eq. (3.3) is given. However, the proposed method is not restricted to any residual function.

For two consecutive frames $I(x, y, t)$ and frame $I(x + u, y + v, \text{t+1})$, the proposed objective function for estimating the optical flow $\mathbf{w} = [u, v]^T$ at a point $\mathbf{p} = [x, y]^T$ is divided into three different parts. Hence, it is formulated as follows:

$$\min_{u,v} \; \mathcal{E}(u, v) = \sum_{\Omega} \left(\lambda \mathcal{E}_{data} + \gamma \mathcal{E}_{smooth} + \mathcal{E}_{dual}\right), \tag{4.2}$$

where

$$\mathcal{E}_{data} = \rho(x, y, u, v) \tag{4.3}$$

$$\mathcal{E}_{smooth} = \| \nabla u \| + \| \nabla v \| \tag{4.4}$$

$$\mathcal{E}_{dual} = \frac{1}{2\theta} \left((u - \hat{u})^2 + (v - \hat{v})^2\right) \tag{4.5}$$

where ρ is a residual function between two images features, u and v are the horizontal and vertical optical flow components, and λ, γ, and θ are weights of the data, the smoothness, and the dual terms. The energy function Eq. (4.2) can be solved iteratively after dividing it into two parts using the quadratic coupling term $\mathcal{E}_{dual}$.

$$\mathcal{E}_d = \lambda \mathcal{E}_{data} + \mathcal{E}_{dual} \tag{4.6}$$

$$\mathcal{E}_s = \gamma \mathcal{E}_{smooth} + \mathcal{E}_{dual} \tag{4.7}$$

Typically, in the case of the brightness constraint, the residual function is $\rho = (I_1(x, y) - I_2(x + u, y + v))$ and the optimization model Eq. (4.2) leads

to the original dual TV-L1 which was proposed by [PUZ+07]. In the following sections, we explain how to adapt this model to include a texture constraint.

The first part of Eq. (4.2) $\mathcal{E}_{data} = \rho(x, y, u, v)$ is called the data-term and it is used together with the coupling term to estimate u and v as follows:

$$\min_{u,v} \ \mathcal{E}_d(u, v) = \sum_{\Omega} \left(\lambda \rho(x, y, \mathbf{u}, \mathbf{v})^2 + \frac{1}{2\theta} \left((u - \hat{u})^2 + (v - \hat{v})^2 \right) \right) \tag{4.8}$$

where $\hat{u}$ and $\hat{v}$ are the auxiliary optical flow variables and θ is a threshold. To illustrate the algorithm, assume $\mathbf{S}_1(x, y)$ and $\mathbf{S}_2(x + u, y + v)$ are two strings represented two descriptors extracted from the two images $I_1(x, y)$ and $I_2(x + u, y + v)$, respectively. Thus, the residual function ρ is rewritten as:

$$\rho(x, y, u, v) = \mathbf{S}_2(x + u, y + v) - \mathbf{S}_1(x, y), \tag{4.9}$$

Eq. (4.9) is a nonlinear with respect to u and v. For simplicity, lets assume $\mathbf{w} = [u, v]^T$ and $\hat{\mathbf{w}} = [\hat{u}, \hat{v}]^T$. Thus, a linearization of $\mathbf{S}_2(x+u, y+v)$ or $\mathbf{S}_2(x, y, \mathbf{w})$ around the starting value of $\mathbf{w}$ is used as presented in [MRM+14]. Hence, a first order Taylor expansion realizes this task as follows:

$$\mathbf{S}_2(x, y, \mathbf{w}) \approx \mathbf{S}_2(x, y) + \nabla^T \mathbf{S}_2(x, y, \hat{\mathbf{w}})(\mathbf{w} - \hat{\mathbf{w}}). \tag{4.10}$$

The derivative $\nabla^T \mathbf{S}(x, y, \hat{\mathbf{w}}) = \left[\frac{\partial \mathbf{S}}{\partial x} = \mathbf{S}_x, \frac{\partial \mathbf{S}}{\partial y} = \mathbf{S}_y \right]^T$ is computed by applying a derivative mask (i.e. Sobel) to the eight channels image in the x and y directions. Therefore, the residual ρ function leads to:

$$\begin{aligned} \rho(x, y, \mathbf{w}) &\approx \tilde{\rho}(x, y, \mathbf{w}) \\ &= \mathbf{S}_2(x, y) - \mathbf{S}_1(x, y) + \nabla^T \mathbf{S}_2(x, y, \hat{\mathbf{w}})(\mathbf{w} - \hat{\mathbf{w}}). \end{aligned} \tag{4.11}$$

Assuming constant values of $\hat{u}$ and $\hat{v}$, the intention is to minimize the objective function Eq. (4.11) with respect to u and v. Therefore, the partial derivatives with respect to u and v are calculated as follows:

$$\begin{aligned}\frac{\partial}{\partial u}\left(\lambda\tilde{\rho}(x,y,\mathbf{w})^2+\frac{1}{2\theta}(u-\hat{u})^2\right)=0,\\ \frac{\partial}{\partial v}\left(\lambda\tilde{\rho}(x,y,\mathbf{w})^2+\frac{1}{2\theta}(v-\hat{v})^2\right)=0.\end{aligned} \tag{4.12}$$

The above equations are expressed in vector form as:

$$\begin{aligned}2\lambda\left(\mathbf{S}_t+\nabla\mathbf{S}_2\left(x,y,\hat{\mathbf{w}}\right)^T\left(\mathbf{w}-\hat{\mathbf{w}}\right)\right)\nabla\mathbf{S}_2(x,y,\hat{\mathbf{w}})+\\ \frac{1}{\theta}(\mathbf{w}-\hat{\mathbf{w}})=0,\end{aligned} \tag{4.13}$$

where $\mathbf{S}_t=\mathbf{S}_2(x,y)-\mathbf{S}_1(x,y)$. Eq. (4.13) is linear with respect to u and v and is written as:

$$A\mathbf{w}=\mathbf{B}. \tag{4.14}$$

Eq. (4.14) is solved using the least square minimization as follows:

$$\mathbf{w}=A^{-1}\mathbf{B}. \tag{4.15}$$

The data-term of the texture constancy in the case of non-binary descriptors such as HOG and DAG includes a residual of two texture features extracted from the input images. The residual function sums up all the variations among eight channels and reflects the similarity between the two strings. On the contrary, for binary descriptors, such as LDP, Census transform, and MLDP, the data-term involves the hamming distance between the 8-bit channel descriptors extracted through a local descriptor. Accordingly, at every pixel, the Hamming distance is computed by counting the number of differences between the two descriptors.

Generally, the residual function ρ between two N-channel descriptors can be described as:

$$\begin{aligned}\rho(x,y,u,v)^2 &= \sum_{i=1}^{N} \rho_i(x,y,u,v)^2 \\ &= \sum_{i=1}^{N} \left(\mathbf{S}_{2,i}(x+u, y+v) - \mathbf{S}_{1,i}(x,y)\right)^2, \end{aligned} \tag{4.16}$$

In practice, the summation over all ρ_i^2 measures the residual between two descriptors. Therefore, the final data-term is formulated as:

$$\min_{u,v} \ \mathcal{E}_{Id}(\mathbf{w}) = \sum_{\Omega} \left(\lambda \sum_{i=1}^{N} \rho(x,y,\mathbf{w})^2 + \frac{1}{2\theta}(\mathbf{w} - \hat{\mathbf{w}})^2 \right), \tag{4.17}$$

The matrices A and $\mathbf{B}$ of the linear system described in Eq. (4.15) are written as:

$$A = \begin{bmatrix} 2\lambda\theta + \sum \mathbf{S}_x^2 & \sum \mathbf{S}_x \mathbf{S}_y \\ \sum \mathbf{S}_x \mathbf{S}_y & 2\lambda\theta + \sum \mathbf{S}_y^2 \end{bmatrix}. \tag{4.18}$$

and

$$\mathbf{B} = \begin{bmatrix} \left(2\lambda\theta + \sum \mathbf{S}_x^2\right)\hat{u} + \sum \mathbf{S}_x \sum \mathbf{S}_y \hat{v} - \sum \mathbf{S}_x \sum \mathbf{S}_t \\ \left(2\lambda\theta + \sum \mathbf{S}_y^2\right)\hat{v} + \sum \mathbf{S}_x \sum \mathbf{S}_y \hat{u} - \sum \mathbf{S}_y \sum \mathbf{S}_t \end{bmatrix}. \tag{4.19}$$

The smoothness term of Eq. (4.2) contains the regularization term:

$$\min_{\mathbf{w}} \ \mathcal{E}_s(\hat{\mathbf{w}}) = \sum_{\Omega} \left(\frac{1}{2\theta}(\mathbf{w} - \hat{\mathbf{w}})^2 + \parallel \nabla \hat{\mathbf{w}} \parallel \right), \tag{4.20}$$

The smoothness term Eq. (4.20) represents an isotropic total variation [PUZ+07]. Correspondingly, Eq. (4.20) is decomposed into two parts, and it can be rewritten as:

$$\mathcal{E}_{\hat{u}} = \sum_{\Omega} \left[\frac{1}{2\theta}(u - \hat{u})^2 + \parallel \nabla \hat{u} \parallel \right], \tag{4.21}$$

$$\mathcal{E}_{\hat{v}} = \sum_{\Omega} \left[\frac{1}{2\theta}(v - \hat{v})^2 + \parallel \nabla \hat{v} \parallel \right]. \tag{4.22}$$

Assuming constant values of u and v after solving the data-term Eq. (4.15), the aim is to minimize the differential equations $\mathcal{E}_{\hat{u}}$ and $\mathcal{E}_{\hat{v}}$ for the two unknowns, $\hat{u}$ and $\hat{v}$. Therefore, the Euler-Lagrange equation is applied as follows:

$$- div \left[\frac{\nabla u}{\| \nabla u \|} \right] + \frac{1}{\theta}(u - \hat{u}) = 0 \tag{4.23}$$

Let $P_u = \frac{\nabla u}{\|\nabla u\|}$. Thus:

$$u = \lambda div(P_u) + \hat{u}, \tag{4.24}$$

By using Eq. (4.23) and Eq. (4.24), P_u can be solved iteratively using the following formula:

$$P_u^{h+1} = \frac{P_u^h + \tau \nabla (div(P_u^h) + \frac{\hat{u}}{\theta})}{1 + \tau \| \nabla (div(P_u^h) + \frac{\hat{u}}{\theta}) \|}, \tag{4.25}$$

where h is the iteration number, and τ is a time step experimentally set to $\tau \leq \frac{1}{8}$. P_v is solved in the same way.

$$P_v^{h+1} = \frac{P_v^h + \tau \nabla (div(P_v^h) + \frac{\hat{v}}{\theta})}{1 + \tau \| \nabla (div(P_v^h) + \frac{\hat{v}}{\theta}) \|}, \tag{4.26}$$

Eq. (4.21) and Eq. (4.22) are solved using a fixed-point iteration scheme (see [MRM+14] for more details). The overall performance of the optimization framework using the new version of the coarse-to-fine scheme is presented in chapter 3. It allows the estimation of the optical flow for large displacements and improve the accuracy. Hence, at each pyramid level, a texture descriptor is calculated, and scaled images are warped based on the estimated optical flow at each scale.

The motion discontinuity is usually problematic due to occlusion and over-smoothing. Moreover, the use of the isotropic TV $L1$ in the regularization term causes an accuracy loss at motion boundaries as well as small image details. To handle this problem, the resulting flow field at each pyramid level requires a de-noising stage to preserve edges and small details. Afterward, the objects boundaries are detected using an edge detection algorithm (i.e. Canny edge detection) and morphological image processing algorithms are used to dilate a

$N \times N$ mask. At each pixel $p = (x, y)$ in a local region, a $N_{x,y}$ weighted median filter [SRB10] is applied:

$$\mathcal{E}_w = \sum_{x,y} \sum_{(\acute{x},\acute{y}) \in N_{x,y}} \varpi_{p,\acute{p}}(|u_{x,y} - u_{\acute{x},\acute{y}}| + |v_{x,y} - v_{\acute{x},\acute{y}}|). \tag{4.27}$$

where $(\acute{x}, \acute{y})$ are the x and y positions of a pixel $\acute{p}$ in a neighborhood of pixel p in a $N_{x,y}$. $\varpi_{p,\acute{p}}$ is a weighting function that takes into account the occlusion state of pixels $\mathcal{O}(p)$ [ST08], color similarity and spatial distance. Thus $\varpi_{p,\acute{p}}$ is written as:

$$\varpi_{p,\acute{p}} \propto exp\left(-\frac{(p-\acute{p})^2}{2\sigma_s^2} - \frac{(I(p) - I(\acute{p}))^2}{2\sigma_r^2}\right) \frac{\mathcal{O}(\acute{p})}{\mathcal{O}(p)}, \tag{4.28}$$

where $I(p)$ and $I(\acute{p})$ are the intensity values of points p and $\acute{p}$, respectively, and σ_s and σ_r are standard deviations.

4.3 Color Texture

In the previous sections, only gray images were used to calculate the optical flow. Therefore, in this section, we discuss how to integrate other constraints such as color constancy to get more accurate and robust results. Although many color spaces such as RGB, HSV, and CIE − LAB can be used, we restrict this work to color spaces which are robust against illumination changes. Thus, HSV and CIE − LAB are used to represent the color information and produce a color texture descriptor. HSV color space separates the intensity (V channel) and the color (S and H channels). Normally, illumination changes alter the intensity channel and do not influence the chromaticity channels. Therefore, we apply a texture descriptor to the intensity channel only and concatenate the values in the color channels to that descriptor. E.g., in the case of HSV, the texture descriptor is applied to describe only the V channel while the color channels are integrated using S and H channels. In turn, for the CIE-LAB color space, A and B channels are used to integrate the color while texture descriptor is applied only to the L channel.

4.4 Experiments and Evaluation

The proposed optical flow model was tested on synthetic and real images using several texture descriptors. Furthermore, several training datasets with various challenging problems were used to evaluate the performance. Moreover, for making a fair comparison with the state-of-the-art methods, we conducted on-line evaluations of the most challenging optical flow datasets such as KITTI 2012, MPI, and Middlebury. Our results were publicly available [BSL$^+$11, GLSU13]. Besides, we performed a thorough analysis of each step to evaluate the strengths and the weaknesses of the proposed approach.

4.4.1 Synthetic Illumination Changes

The proposed variational optical flow model was tested with different feature descriptors after applying a synthetic illumination change to the sequence GROVE2 from the Middlebury dataset which comes with freely available ground-truth data [BSL$^+$11]. The proposed illumination change model aimed at changing a multiplication factor m, addition factor a, and gamma correction γ. The illimination model assumes that $m > 0, a > 0$, and $\gamma > 0$. We applied the synthetic illumination model to one frame of the two consecutive frames. The model is represented as:

$$I_o = \chi \left[255 \left(\frac{mI_i + a}{255} \right)^{\gamma} \right], \tag{4.29}$$

where I_i is the input and I_o the output image. The function χ is used to quantize the resulting value to an 8-bit unsigned integer format [MRM$^+$14]. Figure 4.2 shows a qualitative comparison of the average end-point error (AEE) and the average angular error (AAE) between the flow fields obtained with LDP, LDNP, and MLDP, in addition to both the census transform and gradient constancy. Furthermore, the effects of different values of m, a and γ have individually been assessed. The LDNP and gradient constancy are robust against small changes of m and γ, while they are affected significantly by big variations (see figure 4.2). In turn, LDP, MLDP and the census transform increase the robustness against both small and large changes of m and γ. However, MLDP yields the smallest AEE and AAE for changing values of m, a and γ. In addition, the AEE and AAE with gradient constancy have increased with big values ($a > 20$) of a. In turn,

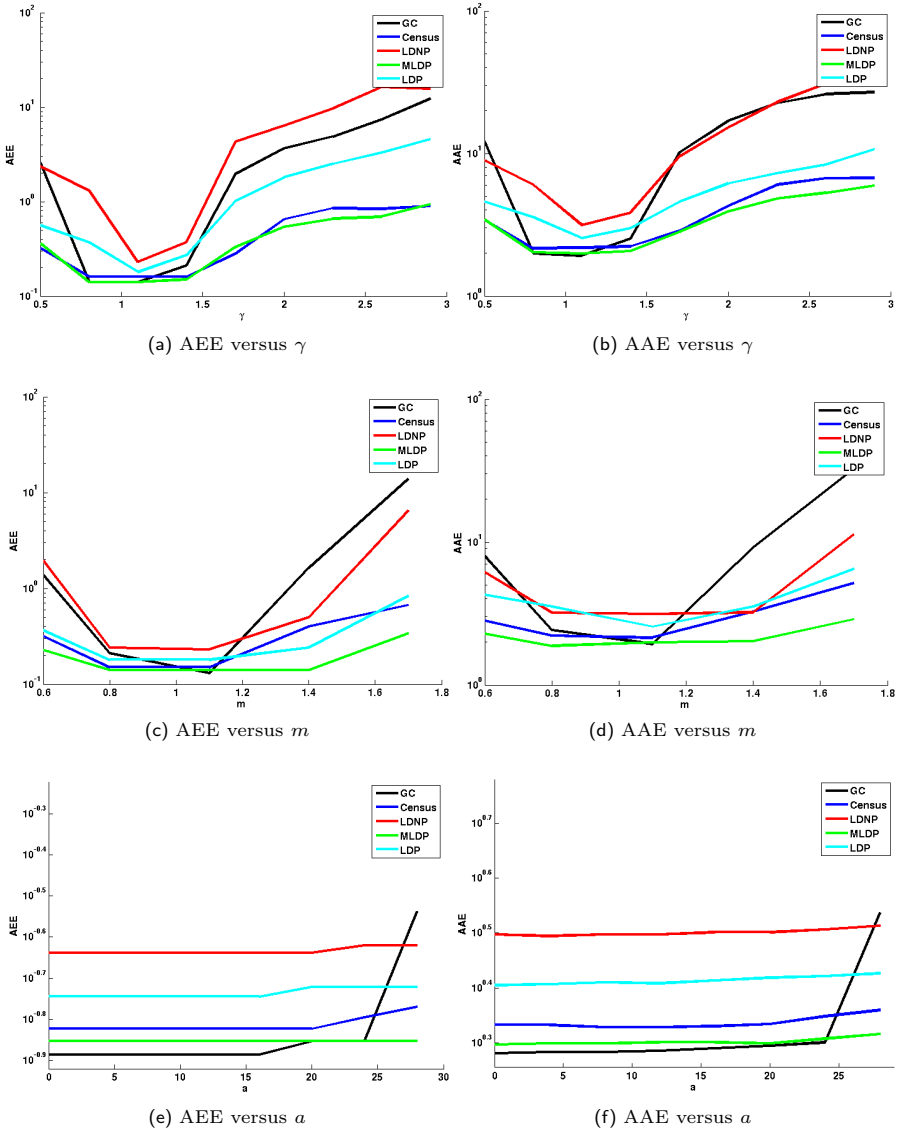
LDNP yields the worst values for both AEE and AAE among the tested data due to its depending on a decimal number of encoding the position of the maximum and minimum edge responses of a neighborhood [MRM+14]. In summary, the proposed approach using the MLDP descriptor surpassed all other approaches.

4.4.2 KITTI 2012 Datasets

For quantitative evaluation, the proposed model was tested on the public KITTI 2012 datasets [GLSU13, MHG15b, MHG18]. The datasets provide challenging scenarios for the evaluation of optical flow algorithms. It contains on the one side dataset called training which has public available ground truth and on the other side there are test sequences images for which ground truth data is not publicly available. The images have a resolution of 1240×376 pixels. The displacements of pixels are generally large, and in some scenarios they are exceeding 250 pixels. The images contain fewer textured regions and strongly varying lighting conditions. Furthermore, the images exhibit many non-Lambertian surfaces such as translucent windows, specular glass, and metal surfaces. Moreover, the images manifest large regions on the image boundaries that move out of the field of view between frames, and they are occluded due to the high speed of the forward motion, such that no correspondence can be established [VSR13].

KITTI 2012 On-line Comparison with the State-of-the-art

We evaluated the results of the proposed model with the HOG and MLDP descriptor (MLDP-OF and TVL1-HOG) with the on-line KITTI vision benchmark. The proposed methods were ranked in the seventh and eighth positions respectively against 60 state-of-the-art optical flow algorithms [GLSU13] with an average of 7.91 and 8.67% incorrect pixels (percentage of pixels with the endpoint error (EE) above 3 pixels). Table 4.1 shows a comparison with the top rank methods and the proposed algorithm using HOG and MLDP. Furthermore, the proposed approach using HOG descriptor was ranked at the first positions of the differential optical flow methods while the proposed approach using MLDP was ranked as



(a) AEE versus γ

(b) AAE versus γ

(c) AEE versus m

(d) AAE versus m

(e) AEE versus a

(f) AAE versus a

Figure 4.2: The effect of synthetic illumination changes on the estimated optical flow using various descriptors.

Table 4.1: The evaluation of the top state-of-the-art methods for optical flow estimation on the the KITTI 2012 benchmark.

Rank	Method	Outliers (%)	AEE
1	PR-Sf+E [VSR13]	4.08 %	0.9 px
2	PCBP-Flow [YMU13b]	4.08 %	0.9 px
3	MotionSLIC [YMU13b]	4.36 %	1.0 px
4	PR-Sceneflow [VSR13]	4.48 %	1.3 px
5	TGV2ADCSIFT [BZDB13]	6.55 %	1.6 px
6	Data-Flow [VRS13]	8.22 %	2.3 px
7	**Proposed method + HOG**	7.91 %	2.0 px
8	**Proposed method + MLDP**	8.67 %	2.5 px
12	fSGM [HK12]	11.03 %	3.2 px
13	TGV2CENSUS [WPB10]	11.14 %	2.9 px
14	C+NL-fast [SRB14b]	12.42 %	3.2 px
25	DB-TV-L1 [PUZ$^+$07]	30.75 %	7.8 px

the second positions. Conversely, the original TV-L1 method [PUZ$^+$07] had 30.75% and the method based on the census transform and the total generalized variation (TGV) [WPB10] had 11.03%, in addition [SRB10] had 24.64%, while the complete rank transform (CRTflow) [DHW13] had 9.43% average incorrect pixels. Nevertheless, most of the top six methods used stereo images to calculate the optical flow, while the proposed method used only monocular images. The KITTI 2012 [MHGow] on-line evaluation table 4.1 sorts all methods according to the percentage of non-occluded bad pixels (outliers) using the end-point error and a threshold. For each method, the following parameters are calculated: Out-Noc is the percentage of outliers in non-occluded areas. Out-All is the percentage of outliers in total. Avg-Noc is the end-point error in non-occluded areas. Avg-All is the end-point error in total. To reduce the amount of data, in table 4.1, we named the Out-Noc as Outliers (%) and the Avg-Noc as the AEE. For details comparison with all criteria and all methods, the reader is directed to visit the KITTI on-line website [GLSU13]. On the KITTI benchmark, the proposed method using HOG is named as TVL1-HOG [RMG$^+$13], while the proposed method with MLDP is named as MLDP-OF [MRM$^+$14].

We evaluated the proposed approach based on the MLDP and HOG descriptors.

Table 4.2: The percentage of outliers using different thresholds for the estimated optical flow model using MLDP and HOG on the KITTI 2012 benchmark.

	MLDP	HOG
2 px	11.10 %	12.06 %
3 px	08.67 %	07.91 %
4 px	07.55 %	06.20 %
5 px	06.84 %	05.26 %

Table 1.2 shows a comparison between MLDP and HOG using different thresholds for the outliers. It can be seen that MLDP provide lower percentage of outliers than HOG in the case of small threshold values.

We compared the proposed algorithm using HOG and MLDP with the state-of-the-art methods which provide solutions for the illumination change problem. The results are shown in table 4.3. As shown in table 4.3 the proposed method using HOG and MLDP outperformed many of the state-of-the-art methods.

Table 4.3: KITTI 2012 on-line evaluation among the proposed approach using HOG/MLDP and some of the state-of-the-art methods dealing with illumination change problem.

Method	Outliers (%)	AEE (pixel)
FlowFields [BTS15]	5.77 %	1.4 px
NLTGV-SC [RBP14a]	5.93 %	1.6 px
BTF-ILLUM [DSV$^+$14]	6.52 %	1.5 px
Proposed method + HOG	7.91 %	2.0 px
Proposed method + MLDP	8.67 %	2.4 px
CRTflow [DHW13]	9.43 %	2.7 px
C++ [SRB14a]	10.04 %	2.6 px
ROF-NND [ADGB16]	10.44 %	2.5 px
DSPyNet [SW18]	10.64 %	2.4 px
AggregFlow [FBK16]	12.23 %	3.1 px
CPNFlow [YS18]	13.01 %	2.0 px
Grts-Flow-V2 [ZLS17]	15.63 %	3.2 px
UnsupFlownet [JHD16]	34.85 %	4.6 px
FSDEF [GM17]	36.85 %	8.8 px
FlowNetS+ft [DFI$^+$15]	37.05 %	5.0 px
RLOF(IM-GM) [SGS16]	37.49 %	8.2 px
DIS-FAST [KTDVG16]	38.58 %	7.8 px

Robust Optical Flow for Challenging Sequences

GCPR 2013 [BW13] organized a special session for robust optical flow evaluation to encourage research on robust optical flow with a focus on challenging real-world scenes. Therefore, the conference's committee prepared challenging sequences from KITTI 2012 training dataset that include common difficulties including illumination changes, large displacements, low-textured areas, reflections, and specularities [BWml]. In order to test the robustness of the proposed method, I have participated in this special session with the results of these sequences obtained using the proposed algorithm using HOG descriptor and it outperformed all the state-of-the-art and other competitors (i.e. [WPB10], [SRB10], [ZBW11], and [VRS13]). Moreover, the same image sequences have been used to evaluate the proposed method using MLDP, census transform and gradient constancy (GC) and we compared the results to the state-of-the-art methods [ZBW11], [SRB10], [BW05], [HG81] [WPB10]. The results obtained by [VRS13] are not public available, however this method used a discrete optimization method based on the census transform to estimate the optical flow. Tables 4.4 and 4.5, show comparisons between the proposed method and state-of-the-art methods based on the percentage of outliers. Among the evaluated approaches, the optical flow model based on MLDP yielded the most accurate flow fields concerning the state-of-the-art methods for real images that include both illumination changes, reflections, large displacements, and other difficulties.

4.4.3 MPI Dataset

The proposed algorithm can be used for video processing application. Therefore, we tested the proposed algorithm on the MPI-sintel datasets [BWSBts] ("clean" and "final" datasets) which has long sequences, large motions, specular reflections, motion blur, defocus blur, and atmospheric effects. The "clean" and the "final" datasets are good test data to illustrate the effect of the proposed approach in a video sequence. We evaluated the proposed method using the MLDP descriptor (MLDP-OF) [MRM$^+$14], with the MPI-Sintel benchmark. MPI-Sintel benchmark

Table 4.4: The percentage of outliers of the proposed methods and state-of-the-art methods using four challenging illumination changes sequences [BW13] from the KITTI 2012 datasets.

Method	Seq 44	Seq 11	Seq 15	Seq 74
MLDP	**20.42%**	**29.67%**	**23.85%**	**56.01%**
Gradient constancy	29.25%	35.72%	26.41%	59.20%
OFH 2011 [ZBW11]	23.22%	37.26%	32.20%	62.90%
Census (5 × 5)	35.23%	33.93%	29.04%	57.57%
Census (3 × 3)	29.55%	37.54%	33.74%	57.43%
SRB 2010 [SRB10]	26.58%	40.61%	32.85%	62.94%
SRBF 2010 [SRB10]	31.83%	40.34%	35.13%	64.89%
BW 2005 [BW05]	32.44%	33.95%	47.70%	71.44%
HS 1981 [HG81]	42.96%	38.84%	58.08%	82.14%
WPB 2010 [WPB10]	49.09%	49.99%	67.28%	88.67%

Table 4.5: The percentage of outliers of the proposed methods and state-of-the-art methods using four challenging sequences large displacement [BW13] from the KITTI 2012 datasets.

Method	Seq 147	Seq 117	Seq 144	Seq 181
MLDP	**11.79%**	18.67%	**41.05%**	**58.25%**
OFH 2011 [ZBW11]	15.04%	**16.26%**	42.04%	63.86%
Gradient constancy	12.28%	17.70%	44.51%	67.63%
SRB 2010 [SRB10]	14.59%	24.71%	50.67%	67.11%
SRBF 2010 [SRB10]	14.79%	24.41%	50.66%	68.41%
BW 2005 [BW05]	16.98%	28.80%	46.98%	69.04%
Census (5 × 5)	13.98%	27.33%	47.68%	73.85%
Census (3 × 3)	14.76%	28.80%	48.97%	73.63%
HS 1981 [HG81]	24.84%	43.24%	51.89%	74.11%
WPB 2010 [WPB10]	32.72%	46.80%	52.25%	76.00%

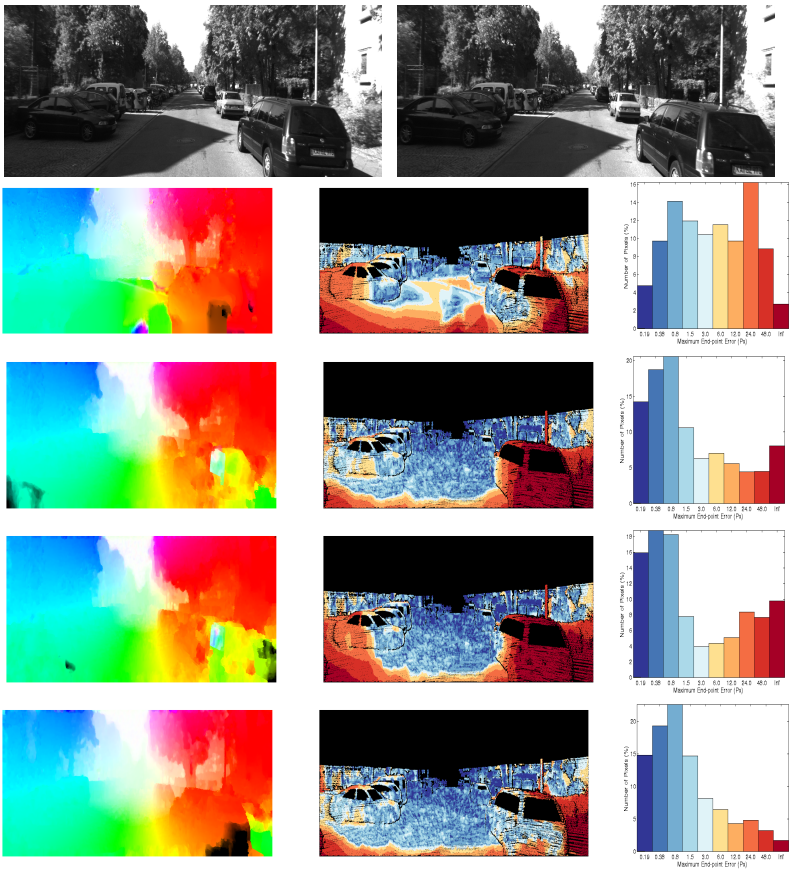


Figure 4.3: Row 1: Sequence 44 of the KITTI 2012 datasets. Optical flow field, error image and error histogram with: Row: 2 Brightness constancy, Row 3: 3×3 census transform, Row 4: 5×5 census transform, Row 5: MLDP. ((from [MRM+14] with permission from IEEE))

Table 4.6: MPI On-line evaluation. AEE of the top ranked methods.

Method	AEE
DiscreteFlow [MHG15a]	6.07 px
EpicFlow [RWHS15b]	6.28 px
SPM-BP [LMB+15]	7.32 px
PH-Flow [YL15b]	7.42 px
TV-L1-MLDP [MRM+14]	8.28 px
Classic+NLP [SRB14b]	8.29 px
EPPM [BYJ14b]	8.38 px
NLTGV-SC [RBP14b]	8.74 px
HCOF+mult [KT15a]	8.80 px
Channel-Flow [SLSLMB14]	8.83 px
LDOF [BM11]	9.11 px
DF [MSP16]	9.19 px
ROF-NND [ADGB16]	9.29 px

uses the average end-point error AEE and not the percentage of outliers to sort the different methods. The proposed method has been ranked in the 8^{th} position out of 62 methods regarding the "final" sets. Table 4.6 shows a comparison between the proposed approach and the state-of-the-art methods. The proposed model provides satisfactory flow fields and copes with illumination changes, motion blur and defocus blur, especially in the "final" dataset. In another experiment, we compared the estimated optical flow fields with MLDP, the census transform, as well as the gradient and brightness constancy on some of the MPI challenges sequence [MRM+14]. Figure 4.4 shows the estimated flow fields for the proposed model with brightness constancy, gradient constancy, the census transform and MLDP on two different sequences. It can be seen that TV-L1 model based on the classical brightness constraint is not able to produce consistent flow fields. Despite, the model based on MLDP can detect the correct motion, especially with the final-sets suffering from motion blur. As shown in figure 4.4, the flow fields estimated with MLDP are more accurate than the other flow fields based on the other constancy assumptions. Besides, they contain more motion details and preserve flow discontinuities, as well as the contours of objects, are significantly more precise than other data terms mentioned above. "final".

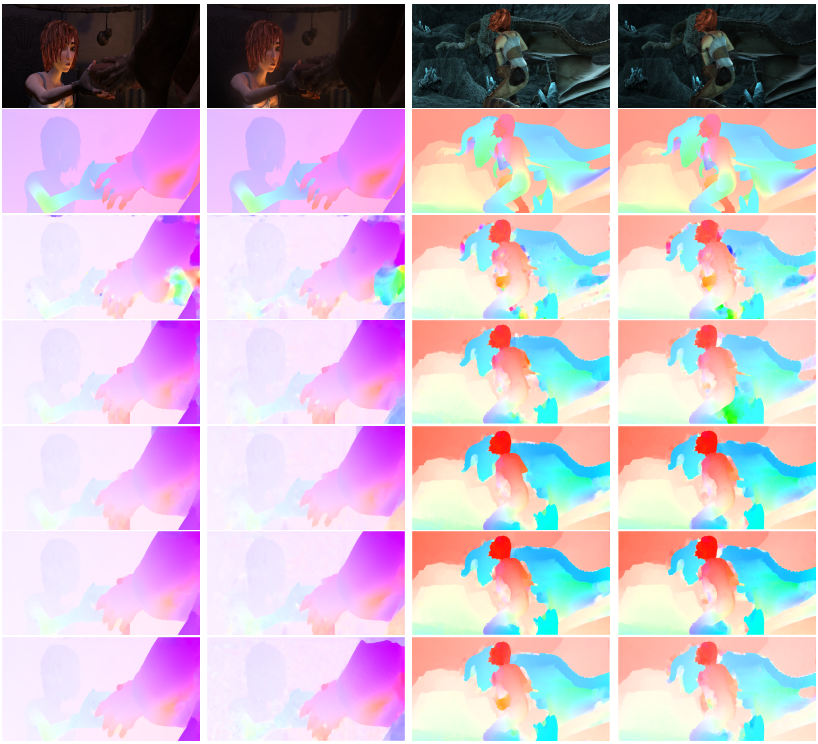


Figure 4.4: Row 1: Sequences form MPI-sintel datasets. Row 2: optical flow ground truths. Row 3: Optical flow fields using the brightness constancy. Row 4: Optical flow fields of using the gradient constancy. Row 5: Optical flow fields using a (3×3) census transform. Row 6: Optical flow fields using a (5x5) census transform. Row 7: Optical flow fields using MLDP. (from [MRM+14] with permission from IEEE)

4.4.4 Middelebury Dataset

As mentioned in chapter 3, The Middlebury [BSL+al] dataset contains eight training images with known ground-truth, in addition to twelve testing images without known ground-truth [BSL+11]. It is used to evaluate the performance of the proposed algorithm based on different constancy assumptions. Table 4.7 shows a qualitative comparison of the results for the average end-point error (AEE) based on brightness constraint (BC), census (3×3 and 5×5), MLDP, and gradient constancy constraint compared to the methods proposed in [VRS13] and [RMG+13]. The proposed algorithm based on MLDP outperforms all different methods and yields the minimum AEE.

Table 4.7: AEE in pixels for the proposed method using the MLDP compared with some of the state-of-the-art methods on the Middelbury training dataset

Method	Dimetrodon	Grove2	Grove3	Hydrangea	Rubberwhale	Urban2	Urban3	Venus
BC	0.21	**0.14**	0.64	0.19	0.11	0.35	0.69	0.29
census (7×7) [VRS13]	0.24	0.21	0.66	0.19	0.14	0.39	0.56	0.36
CSAD (7×7) [VRS13]	0.18	0.23	0.61	0.18	0.12	0.36	0.63	0.32
HOG (3×3) [RMG+13]	0.18	0.23	0.68	0.22	0.17	0.35	0.44	0.33
HOG (5×5) [RMG+13]	**0.14**	0.15	**0.49**	0.18	0.10	0.26	**0.40**	0.26
census (3×3)	0.28	0.16	0.54	0.18	0.11	0.25	0.44	0.25
census (5×5)	0.21	**0.14**	0.52	**0.17**	0.10	0.25	0.43	0.25
MLDP-OF	**0.14**	**0.14**	0.50	**0.17**	**0.09**	**0.24**	**0.40**	**0.24**

Figure 4.5 shows qualitative results of some examples from the Middlebury dataset. The proposed algorithm based on MLDP yields precise flow fields and preserves edges and motion boundaries [MRM+14].

According to the Middlebury benchmark, the proposed method MLDP-OF [MRM+14], had an average rank of 30.3 concerning the average end-point error (AEE). On the contrary, the other much-advanced methods, such as the complementary optical flow [ZBW11, ZBW+09] and the optical flow based on the complete rank transform [DHW13] had 30.9, 39.5 and 44.8, respectively.
Table 4.8 shows a comparison with selected state-of-the-art methods on the Middlebury dataset.

Figure 4.5: (Row 1) Sequences: Grove2, Rubberwhale, Venus, Dimetrodon, and Urbahn2 from Middlebury dataset. (Row 2) Corresponding ground-truths. (Row 3) Corresponding flow fields using MLDP.

Table 4.8: Middlebury on-line evaluation. The AEE in pixels of some of the state-of-the-art methods.

Method	Army	Mequon	Schefflera	Wooden	Grove	Urban	Yosemite	Tedy
NN-field [CJL+13]	0.08	0.17	0.19	0.09	0.41	0.52	0.13	0.43
Layers++ [SSB10]	0.08	0.19	0.20	0.48	0.48	0.47	0.15	0.46
PH-Flow [YL15a]	0.08	0.21	0.23	0.56	0.56	0.30	0.15	0.44
TV-L1-MCT [MM12b]	0.08	0.24	0.32	0.14	0.72	0.54	0.11	0.54
MLDP-OF [MRM+14]	0.09	0.19	0.24	0.16	0.74	0.46	0.12	0.78
EpicFlow [RWHS15b]	0.12	0.25	0.39	0.19	0.89	0.53	0.10	0.67
CRTflow [DHW13]	0.11	0.24	0.50	0.23	0.86	0.60	0.12	0.19
LDOF [BM11]	0.12	0.32	0.43	0.45	1.01	1.10	0.12	0.94

4.4.5 Evaluation of Color Texture

In this experiment, we tested the effect of integrating color information. Therefore, we did not use a weighted median filter. Table 4.9 and table 4.10 show the effect of integrating color information on the AEE and the outliers (AEE_out) with the HSV and the CIE-Lab color spaces.

Table 4.9: The AEE using gray, HSV and CIE-Lab color space.

Algorithm	Gray	HSV	CIA-Lab
HOG	2.53 px	2.48 px	2.52 px
MLDP	2.67 px	2.64 px	2.67 px
Census	2.95 px	2.84 px	2.95 px

Table 4.10: The percentage of outliers using gray, HSV and CIE-Lab color space.

Algorithm	Gray	HSV	CIE-Lab
HOG	10.31 %	10.31 %	10.28 %
MLDP	11.02 %	10.93%	11.01 %
Census	12.09 %	11.90 %	12.09 %

The HSV color space produced better optical flow than the CIE-Lab color space. Although there is only a little improvement in the average results, some individual sequences have a more significant difference.

4.4.6 Evaluation of the Execution Time

In this section, we evaluate the execution time of the proposed algorithm on different platforms, using different pyramid factors. Table 4.11 shows the specifications of the computers that have been used to evaluate the proposed algorithm.

Table 4.11: The specifications of the test platform that has been used to test the proposed algorithm.

CPU	Intel Core i7 5820k @ 12 x 4.1 GHz
RAM	16 GB DDR4-2400 @ Quad Channel 2400 MHz
GPU	Nvidia Geforce GTX 970 4GB @ 1177 MHz
OS	Windows 7 64-bit

The execution times for the algorithm can be seen in the table 4.12.

Table 4.12: Run time of the proposed algorithm using difference descriptors

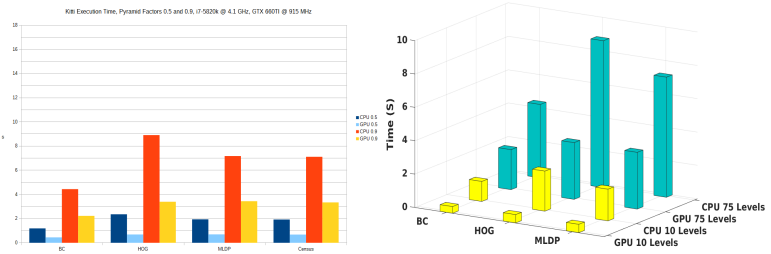
Descriptor	Pyr.Fac.	CPU [s]	GPU [s]
HOG	0.5	2.362	0.647
HOG	0.9	8.913	3.300
MLDP	0.5	1.940	0.658
MLDP	0.9	7.178	3.332
Census	0.5	1.924	0.644
Census	0.9	7.116	3.313

The extension to the HSV color space including the descriptors with Hue and saturation slightly increases the execution time (see table 4.13):

Table 4.13: Run time of the proposed algorithm with color information using difference descriptors

Algorithm	Pyr.Fac.	CPU [s]	GPU [s]
HOG	0.5	2.563	0.736
HOG	0.9	9.476	3.547
MLDP	0.5	2.097	0.739
MLDP	0.9	7.701	3.622
Census	0.5	2.088	0.730
Census	0.9	7.666	3.540

Figure 1.6a shows the execution time of the proposed algorithm on a CPU and a GPU using different pyramid factor. It can be seen from that figure, that BC has the lowest processing time, in the second place come the MLDP and Census, while the HOG descriptor has the highest processing time. The number of levels in the coarse-to-fine scheme affect the processing time. Therefore, we evaluated the processing time of the proposed algorithm with different descriptor with different number of levels (see figure 4.6b).

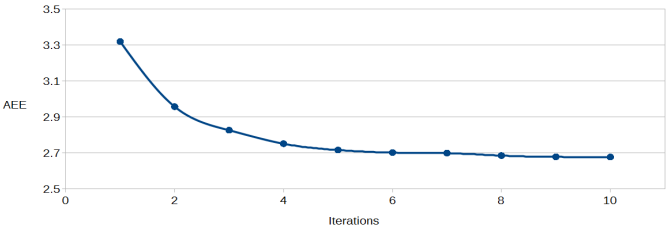


(a) (b)

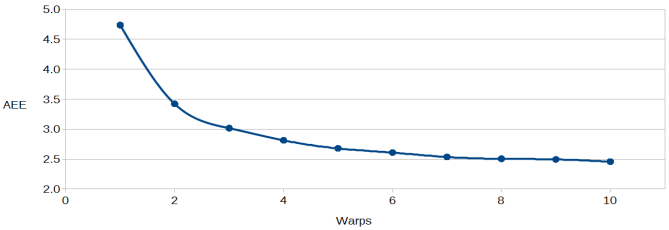
Figure 4.6: Processing time of the proposed approach using deferent descriptors.

The optical flow estimation algorithm used the TV-L1 approach that minimize an objective function. The objective function is a kind of partial deferential equation. We used the fixed point iteration scheme together with the image wrapping algorithm to minimize that function. Figure 4.7a shows the effect of the number of iteration while figure 4.7b shows the effect of the number of warps on the average endpoint error. It can bee seen that the error decreased and reach the steady state after 5 iteration and 5 warps. The runtime is also affects by the resolution of the images. Hence, figures 4.7c and 4.7d show the effect of resolution of images on the processing time as well as the average endpoint AEE. It can be seen that the resolution increase the processing time exponentially, while the error is decreasing.

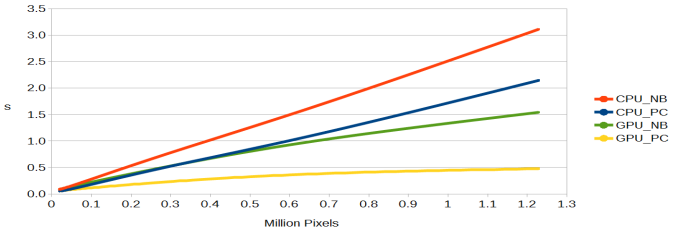
Figure 4.7: Analysis of different parameters on the average end-point error and the processing time.



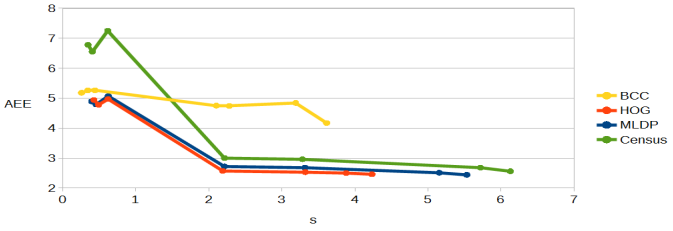
(a)



(b)



(c)



(d)

4.5 Summary

In this chapter, we present the texture constancy assumption for a variational optical flow model. Image texture has been extracted using several texture descriptors (i.e., HOG, LDP, Census transform, ..., and so forth). Besides, we integrated a residual function based on the texture constraint directly in the duality of the TV-L1 optical flow model with a weighted non-local module. We proposed to use a new texture descriptor called MLDP. The MLDP descriptor is robust against noise and illumination variations. Its features are derived from eight edge responses generated with a compass mask. The proposed algorithm has been evaluated with different datasets such as KITTI 2012, MPI-Sintel, and Middlebury benchmarks. Ultimately, the proposed method provided correct flow fields and increased the robustness against illumination changes and large displacements. In the next chapter, we discuss the cases in which the texture constraint fails, especially in case of texture-less regions while texture and brightness constraints fail to find a solution because of the singularity of the system of equations.

5 Proposed Monocular Epipolar Line Constraint

In chapter 4, we developed an optimization scheme for optical flow estimation based on an objective function, which minimizes a residual function between two local descriptors. This new formulation allows the integration of different types of texture descriptors (i.e., HOG, MLDP, Census, ..., and so forth) with different types of residual functions without having to change the optimization scheme. In chapter 5, we extend this concept and show how other constraints can be integrated. Therefore, we investigate the usage of the monocular epipolar geometry constraint for the calculation of optical flow. To our knowledge, this is the first effort to impose the epipolar geometry constraint directly in the differential optical flow estimation.
This chapter is organized as follows: an introduction is presented in section 5.1. The integration of the epipolar constraint in the proposed TV-L1 for texture descriptor formulation is discussed in section 5.2. Section 5.3 describe the optical flow model, while section 5.4 introduces an iterative method for the update of the fundamental matrix. The evaluations of the proposed method are conducted in section 5.5. Finally, section 5.6 concludes this chapter.

5.1 Introduction

For homogeneous and therefore less textured regions, the data term Eq. (4.18) can be singular, i.e., all equations are dependent. Nevertheless, if the optical flow between two images is caused by camera motion for the most parts and the objects are mostly stationary, the epipolar constraint can be applied to add one more constraint to estimate the optical flow even for low textured images [MMM15]. To calculate the epipolar geometry between two frames, the

fundamental matrix which describes the camera motion between two frames can be estimated. Consequently, for each pixel in one frame, the location of the correspondent point in the next consecutive frame is constrained over an epipolar line. Typically, fundamental matrices are estimated after calculating point correspondences and applying the 8-point or the 7-point methods [HZ03]. The epipolar geometry of a stereo camera has been widely used for the estimation of optical flow [TSS17]. On the contrary, only few methods have applied the monocular epipolar geometry for the optical flow estimation. In [YMU13b], the correspondent points are found by searching over epipolar lines and using a semi-global block matching technique. The method has three main shortcomings. First, the estimation of the fundamental matrix is not always accurate especially for small baselines, and relatively large rotations. Second, the camera calibration information is used to speed up the epipolar search based on a semi-global block matching method. Third, the applied approximation of the rotation matrix is valid only for small rotations. In [VBW08], the joint estimation of the fundamental matrix and the optical flow estimation are formulated as a total variation with L2 norm (TV-L2) problem. Such a formulation is severely ill-conditioned and diverges in most of the cases if no initial guesses of the fundamental matrices are available. The reason is that the fundamental matrix estimation is sensitive even to a minor amount of measurement noise. Additionally, the formulation of the method is not presented for the multi-resolution pyramid analysis. Moreover, in [YMU13a] the authors estimated the optical flow along the epipolar lines of the ego-motion by adapting slanted-plane stereo models to the problem of monocular epipolar flow estimation. Furthermore, the authors represented the problem as one of inference in a hybrid Markov Random Field (MRF) and applied a flow-aware segmentation algorithm based on superpixels. In turn, [MMM15] used the epipolar constraint for the estimation of sparse optical flows and did not apply a smoothness constraint. In [RVCK16], an approach to dense depth estimation from a single monocular camera was proposed. The authors proposed an algorithm that segments the optical flow field into a set of motion models, each with its epipolar geometry.

The presented work in this chapter does not use any segmentation such as [YMU13a] and [RVCK16]. We apply the epipolar constraint directly in the context of a global-local optimization process. Moreover, we give appropriate

weight to the epipolar constraint profit from both brightness/texture information and epipolar constraint to tackle with the inaccuracy of fundamental matrices. Additionally, we formulate the usage of the epipolar line constraint for the calculation of optical flow in a pyramid context to deal with large optical flows. The effect of different parameters is investigated and optimal parameters to reach the best performance is exhibited. We study the effect of the update of the fundamental matrix, and we show under which conditions the update can improve optical flow estimations. The performance of the proposed formulation is examined by applying different types of descriptors and different methods to estimate fundamental matrices. Ultimately, the challenging KITTI dataset evaluates the method.

5.2 Epipolar Constraint

Assume two matching points, $q = (x, y)$ and $p = (x', y')$ in two consecutive frames captured at two different camera positions, the following equation holds:

$$\mathbf{q}^T \mathrm{F} \mathbf{p} = 0 \tag{5.1}$$

where $\mathbf{p} = [x\ y\ 1]^T$ and $\mathbf{q} = [x'\ y'\ 1]^T$, F is a 3×3 fundamental matrix, which is singular and can be determined using the 8-point or the 7-point method [HZ03]. The epipolar line constraint is formulated as follows: if the dominant motion is induced only by camera motion, the location of a point in image I_2 corresponding to a point in image I_1 is constrained by a line equation. In this regard, Eq. (5.1) can be rewritten as follows:

$$ax'_i + by'_i + c = 0 \tag{5.2}$$

where $[a\ b\ c]^T = \frac{1}{\eta} F\mathbf{q}$. η is a normalization factor such that $a^2 + b^2 = 1$. By substituting $x'_i = x_i + u_i$ and $y'_i = y_i + v_i$ in Eq. (5.2), an equation in terms of u and v is obtained:

$$au + bv = -ax - by - c \tag{5.3}$$

The above formulation is valid for the case that the optical flow is calculated on the original image resolution. For the calculation of large displacement optical flow, the approach proposed in chapter 3 can be applied. Therefore, the image is sub-sampled iteratively several times, and then the flow is calculated from the coarsest to the finest level. Given the fundamental matrix for the original image, the epipolar line equations at each level have to be determined. In this regard we formulate the co-planarity constraint as follows:

$$[\alpha x' \ \alpha y' \ \alpha]\mathrm{F}[\alpha x \ \alpha y \ \alpha]^T = 0 \tag{5.4}$$

where $\alpha = s^l$. We can verify that for two correspondence points on level l such as $[x_l, y_l]^T$, $[x'_l, y'_l]^T$, they can be associated to a point in the original image as follows:

$$\begin{aligned} [x_l, y_l]^T &= \alpha[x, y]^T \\ [x'_l, y'_l]^T &= \alpha[x', y']^T \end{aligned} \tag{5.5}$$

Therefore we have:

$$[x'_l \, y'_l \, \alpha] \, \mathrm{F} \, [x_l \, y_l \, \alpha]^T = 0 \tag{5.6}$$

Consequently, we obtain a new line equation for each point at level l as follows:

$$a_l x'_l + b_l y'_l + c_l = 0 \tag{5.7}$$

As a result, the line equation containing u_l and v_l will be as follows:

$$a_l u_l + b_l v_l + a_l x_l + b_l y_l + c_l \tag{5.8}$$

Let assume that

$$d = a_l \cdot x_l + b_l \cdot y_l + c_l, \tag{5.9}$$

we can write Eq. (5.8) as follows:

$$a_l u_l + b_l v_l + d \tag{5.10}$$

5.3 Optical Flow Model

Generally, local optical flow methods such as [LK81] assume a smooth optical flow field in a small neighborhood of a pixel and minimize the sum of quadratic deviations. Such a neighborhood consists of $N = n \times n$ pixels, and the brightness constraint is evaluated using all pixels [MMM15]. The brightness constraint is usually not fulfilled for all pixels as the assumption of equal flow vectors within the window is violated. More equations concerning u and v can be obtained after assuming smooth optical flow in a neighborhood. The singularities occur in homogeneous regions or at the edges. Typically, the system of equations is formed using $(n \times n)$ equations, and it is usually solved using the least squared minimization technique as mentioned in the previous chapters. Consequently, assigning a different weight for the equations yields different solutions. Given a set of neighbor pixels such as $\{(x_i, y_i)|i = 0...n\}$, the system of equations is formulated as:

$$\begin{bmatrix} w_1 & 0 & ... & 0 \\ 0 & w_2 & & 0 \\ \vdots & ... & \ddots & 0 \\ 0 & & & w_N \end{bmatrix} \begin{bmatrix} I_x(x_1, y_1) & I_y(x_1, y_1) \\ I_x(x_2, y_2) & I_y(x_2, y_2) \\ \vdots & \\ I_x(x_N, y_N) & I_y(x_N, y_N) \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = \tag{5.11}$$

$$\begin{bmatrix} w_1 & 0 & ... & 0 \\ 0 & w_2 & & 0 \\ \vdots & ... & \ddots & 0 \\ 0 & & & w_N \end{bmatrix} \begin{bmatrix} I_t(x_1, y_1) \\ I_t(x_2, y_2) \\ \vdots \\ I_t(x_N, y_N) \end{bmatrix} \tag{5.12}$$

where (x_0, y_0) is the center point for which the optical flow is currently calculated and $(w_1, w_2, ..., w_N)$ are the weights of each equation.

The resulting flow vector is sub-pixel accurate. However, due to the use of the first order Taylor approximation, this algorithm is valid only for small displacement optical flow. To estimate the optical flow in cases of larger displacements, this approach has to embedded into a pyramid approach, by finding the optical flow at low-frequency structures at low-resolution scales of an image and refining the flow on the higher resolution levels as discussed in chapter 3.

We come up with a regularized cost function including the epipolar constraint as follows:

$$\min_{u,v} \; E(u,v) = \sum_{\Omega} \left(\lambda E_{data} + \gamma E_{epip} + \eta E_{smooth} + E_{dual} \right), \tag{5.13}$$

where

$$E_{data} = \rho(x,y,u,v)^2 \tag{5.14}$$

$$E_{epip} = \parallel a_l u + b_l v + d \parallel \tag{5.15}$$

$$E_{smooth} = (\parallel \nabla u \parallel + \parallel \nabla v \parallel) \tag{5.16}$$

$$E_{dual} = \frac{1}{2\theta}(u - \hat{u})^2 \tag{5.17}$$

where λ, η and γ are weights of the data-term, the smoothness term, and the epipolar constraint. In section 4.2.2, we showed how to optimize an objective function based on the residual between two texture descriptors. Here, we optimize the cost function in Eq. (5.13) including epipolar constraint using a combination of local and global costs (see chapter 4 for details) as follows:

$$\min_{u,v} \; E_d(u,v) = \lambda \rho(x,y,u,v)^2 + \gamma(a_l u + b_l v + d)^2 + \frac{1}{2\theta}(u - \hat{u})^2, \tag{5.18}$$

$$\min_{\hat{u},\hat{v}} \; E_s(\hat{u},\hat{v}) = \eta \left(\parallel \nabla \hat{u} \parallel + \parallel \nabla \hat{v} \parallel \right) + \frac{1}{2\theta}(u - \hat{u})^2, \tag{5.19}$$

The residual function $\rho(x,y,u,v)$ in Eq. (5.18) is the same as in section (4.2.2). Using the dual TV-L1 algorithm, Eq. (5.18) can be optimized by solving for (u,v) by doing:

$$\begin{aligned} &\frac{\partial}{\partial u}\left(\lambda \tilde{\rho}(x,y,u,v)^2 + \gamma(a_l u + b_l v + d)^2 + \frac{1}{2\theta}(u - \hat{u})^2 \right) = 0, \\ &\frac{\partial}{\partial v}\left(\lambda \tilde{\rho}(x,y,u,v)^2 + \gamma(a_l u + b_l v + d)^2 + \frac{1}{2\theta}(v - \hat{v})^2 \right) = 0. \end{aligned} \tag{5.20}$$

The equations can be written in vector forms as:

$$\begin{aligned} 2\lambda \left(S_t + \nabla S'(x,y,\hat{u},\hat{v})^T \left([u,v]^T - [\hat{u},\hat{v}]^T \right) \right) \nabla S'(x,y,\hat{u},\hat{v}) + \\ 2\gamma(a_l u + b_l v + d) A_l + \frac{1}{\theta}([u,v]^T - [\hat{u},\hat{v}]^T) = 0, \end{aligned} \tag{5.21}$$

where:

$$A_l = \begin{bmatrix} a_l & 0 \\ 0 & b_l \end{bmatrix} \tag{5.22}$$

Eq. (5.21) is linear with respect to u and v. It is solved as a linear system $A\boldsymbol{w} = B$, where $\boldsymbol{w} = [u, v]^T$. Similar to section 4.2.2, Eq. (5.21) is written in matrix form. Subsequently, the A and B matrices of the linear system described in Eq. (5.21) are written as:

$$A = \begin{bmatrix} \frac{1}{2\lambda\theta} + \sum S_x'^2 + 2 \cdot \gamma \cdot (a_l)^2 & \sum S_x' S_y' + 2 \cdot \gamma \cdot a_l \cdot b_l \\ \sum S_x' S_y' + 2 \cdot \gamma \cdot a_l \cdot b_l & \frac{1}{2\lambda\theta} + \sum S_y'^2 + 2 \cdot \gamma \cdot (b_l)^2 \end{bmatrix}. \tag{5.23}$$

and

$$B = \begin{bmatrix} \left(\frac{1}{2\lambda\theta} + \sum S_x'^2\right) \hat{u} + \sum S_x' \sum S_y' \hat{v} - \sum S_x' \sum S_t + 2 \cdot \gamma \cdot a_l (a_l \cdot x + b_l \cdot y + d) \\ \left(\frac{1}{2\lambda\theta} + \sum S_y'^2\right) \hat{v} + \sum S_x' \sum S_y' \hat{u} - \sum S_y' \sum S_t + 2 \cdot \gamma \cdot b_l (a_l \cdot x + b_l \cdot y + d) \end{bmatrix}. \tag{5.24}$$

We are using the same smoothness term as explained in chapter 4 (see section 4.2.2).

5.4 Enhancement of Fundamental Matrix

Theoretically, it sounds plausible that fundamental matrix can also be enhanced based on the obtained accurate optical flows at each level. To this end, we discuss briefly, how the fundamental matrix can be iteratively improved at each iteration. In the experimental result section, we evaluate how it affects the results practically. In [LDFP93], different methods to enhance iteratively fundamental matrices are discussed. One of the best methods is based on the epipolar distances between each point and its correspondent epipolar line. Given N matched point as (x_i, y_i) and (x_i', y_i'), $i = 1, ..., N$, the following error function should be minimized:

$$\xi = \sum_{i=1}^{N} \frac{(a_i x_i' + b_i y_i' + c_i)^2}{a_i^2 + b_i^2} + \frac{(a_i' x_i + b_i' y_i + c_i')^2}{a_i'^2 + b_i'^2} \tag{5.25}$$

where $[a_i \ b_i \ c_i]^T = F[x_i, y_i, s^l]^T$ and $[a'_i \ b'_i \ c'_i]^T = F^T[x'_i, y'_i, s^l]^T$. Hence the fundamental matrix is a singular matrix, it can be formulated as follows:

$$\mathrm{F} = \begin{bmatrix} f_1 & f_2 & f_3 \\ f_4 & f_5 & f_6 \\ \alpha f_1 + f_4 & \alpha f_2 + f_5 & \alpha f_3 + f_6 \end{bmatrix} \tag{5.26}$$

As a result, the above matrix is written in a vector form as follows:

$$\mathbf{f} = [f_1, \ f_2, \ f_3, \ f_4, \ f_5, \ f_6, \ \alpha]^T \tag{5.27}$$

Eq. (5.25) can be written as multi objective error functions to form a non linear equation system containing N equations. The equation system is solved iteratively in a regularized form using e.g. the Marquardt-Levenberg method.

5.5 Experimental Results

Some concerns for optical flow estimation are the selection of features, finding corresponding points and the estimation of the fundamental matrix. Hence, we used and compared two different feature matching methods: SIFT [Low04] and the pyramid Lucas-Kanade optical flow [LK81]. For the estimation of the fundamental matrix, we evaluate the usage of the 7-point and the 8-point methods by applying the RANSAC algorithm. The 8-point method uses nine matched points while the 7-point method uses eight points.

5.5.1 Epipolar Line Constraint for Sparse Optical Flow

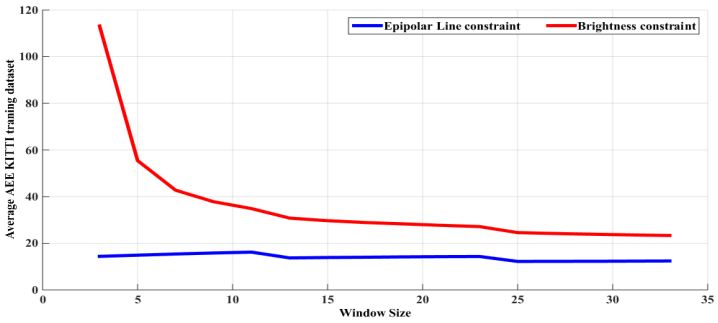
In this experiment, we evaluate the effect of using the epipolar constraint on the data-term based on the training images from the KITTI 2012 dataset. Therefore, we calculate the average end-point error (AEE) and the average angular error (AAE) of the estimated optical flow using two different data-terms. The first data-term [LK81] uses only the brightness constraint in a local window of size $(N \times N)$, while the second data-term integrates the epipolar and the brightness constraints in a local window of size $(N \times N)$. In this experiment, we did not

apply any smoothness regularization. Table 5.1 shows the average errors using various window sizes of the local Gaussian mask with different data terms and different methods for estimating the fundamental matrix. The result demonstrates significant improvement for the epipolar line constraint with lower average errors for the dataset. The best performance is achieved using the fundamental matrix based on the LK and the 7-point method. The reason lies in the fact that SIFT works based on blob features which are not precise enough to yield a good fundamental matrix. The 7-point method also performs better than the 8-point method because it requires fewer feature points, that make it more robust against outliers. In figure 5.1, the effect of increasing the size of the surrounding window is shown. It can be seen, that the brightness constraint has an inferior performance at small window sizes, whereas using the epipolar constraint yielded good results even for the small windows.

Table 5.1: AEE and AAE of the estimated optical flow for the 194 training sequences from KITTI dataset.

Method		AEE			AAE		
		(3 × 3)	(7 × 7)	(15 × 15)	(3 × 3)	(7 × 7)	(15 × 15)
Using Epipolar Line Constraint	LK 7 pts	**17.74 px**	**10.01 px**	**7.83 px**	**9.93°**	**7.93°**	**7.29°**
	LK 8 pts	19.11 px	11.55 px	9.33 px	13.42°	11.24°	10.47°
	SIFT 7 pts	17.87 px	10.25 px	8.19 px	10.35°	8.12°	7.92°
	SIFT 8 pts	18.65 px	11.15 px	9.01 px	12.43°	10.51°	9.84°
Without Epipolar Line Constraint		127.85 px	29.69 px	14.56 px	36.88°	17.23°	11.07°

Figure 5.2 shows comparisons of the AEE and the AAE for each image of the training sequences. It can be seen from figure 5.2 significant improvements of the accuracy for the epipolar line with brightness constraints in most of the sequences. Figure 5.3a shows that the integration of the epipolar line constraint with the brightness constraint fixes the large deviation between the estimated optical flow and the ground truth using only the brightness constraint, see figure 5.3b. However, in scenarios such as sequence 150 (see figure 5.4), the average errors based on the epipolar constraint are more substantial than those for the brightness constraint. In fact, the reason for this error is that, the camera in this scenario performed side translations and relatively high rotations which



(a)

Figure 5.1: The effect of window size on the average AEE for KITTI training dataset.

strongly leads to a wrongly estimated fundamental matrix, even in case of a small computational error [MMM15].

5.5.2 Epipolar Line Constraint for Dense Optical Flow

Based on the KITTI training dataset and the overall formulation presented in section 5.3, we wanted to investigate how the epipolar constraint works if it is applied along with different data terms and fundamental matrix estimation methods. Therefore, in this regard, we used different data terms, feature matching techniques, and fundamental matrix estimation methods, as follows:

- Data term: BC, MLDP, HOG, Census.
- Feature tracking methods: SIFT [Low04], Lucas-Kanade sparse optical flow [LK81].
- Fundamental matrix calculation methods: 8-point, 7-point.

It is interesting to observe how the combination of the three categories mentioned above affects the performance of the optical flow estimation. Moreover, an optimal weight for the epipolar constraint should be found. In figure 5.5 an average of

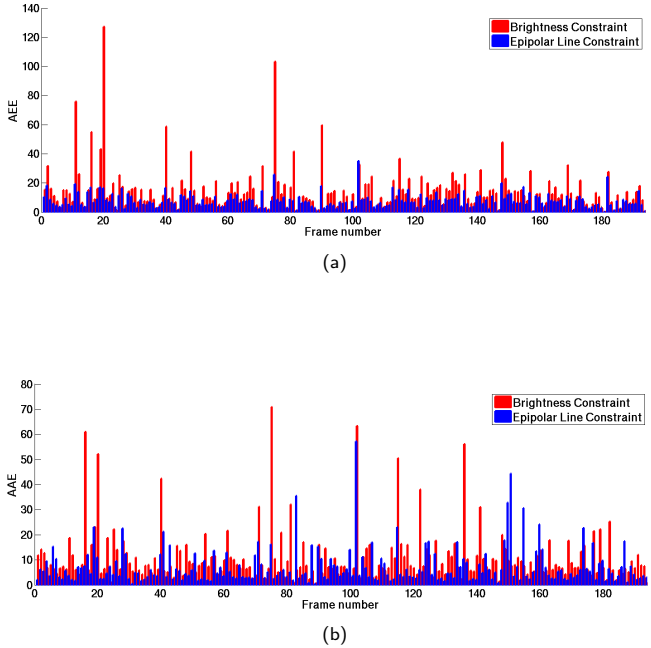
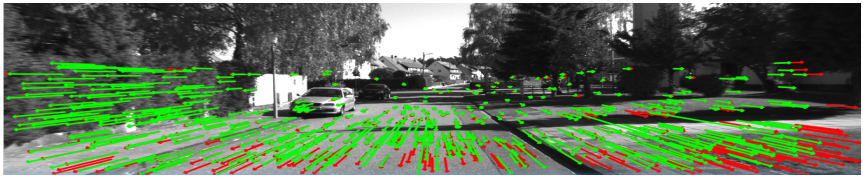


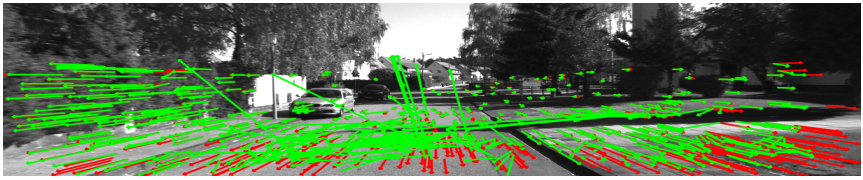
Figure 5.2: The average error for each sequence of the 194 sequences of the KITTI training dataset. (a) AEE. (b) AAE.

of the end-point errors AEE of more than three pixels which we called it as %Outliers can be seen. Figure 5.5a depicts the results based on the 7-point method for fundamental matrix estimation while figure 5.5b is based on the 8-point method. It can be seen, increasing the epipolar weight γ from zero reduces the errors regardless of the type of data term, feature tracker or fundamental matrix estimation method.

Looking at both figures, unexpectedly, we can notice that the LK feature tracker outperforms SIFT noticeably in all cases, despite the common belief about the high accuracy of SIFT. Practically, the corner features in outdoor scenes are



(a)

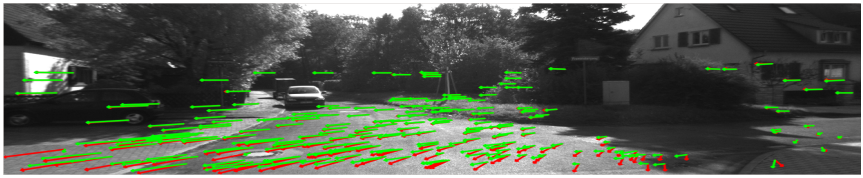


(b)

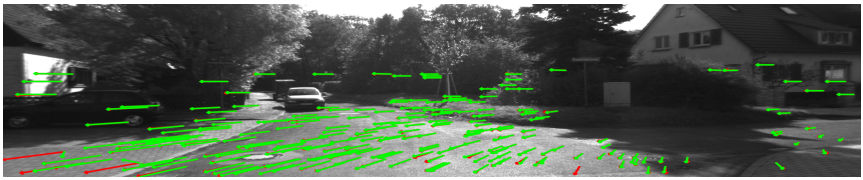
(c)

Figure 5.3: Comparison between ground truth (red) and estimated optical flow (green) for sequence 24 of the KITTI training dataset for some feature points. (a) Using epipolar constraint based on 7-points method. (b) Using brightness constrain. (c) Estimated epipolar lines.

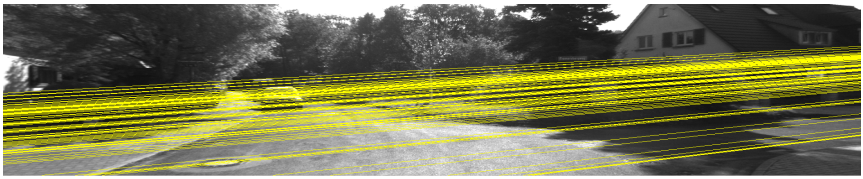
extracted and localized much better using LK than blob features using SIFT. Additionally, unlike SIFT, the LK does not rely on the repetition of features. In SIFT, blob features from both images are extracted and then based on their descriptors the matched points are found. This problem increases the ratio of mismatches especially in the case of occluded points and affects the quality of the fundamental matrix significantly. In turn, the LK tracks corner features from one frame using the sparse optical flow technique. As a result, repeatability is not an issue for the LK tracker. Concerning the weight of the epipolar constraint, we can see that the minimum errors are obtained for the weight $\gamma = 1.5$. Apparently, for smaller γ, the data term plays a more critical role in the determination of



(a)



(b)

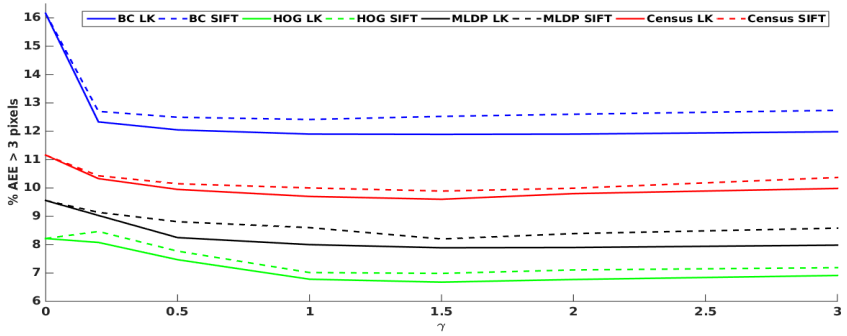


(c)

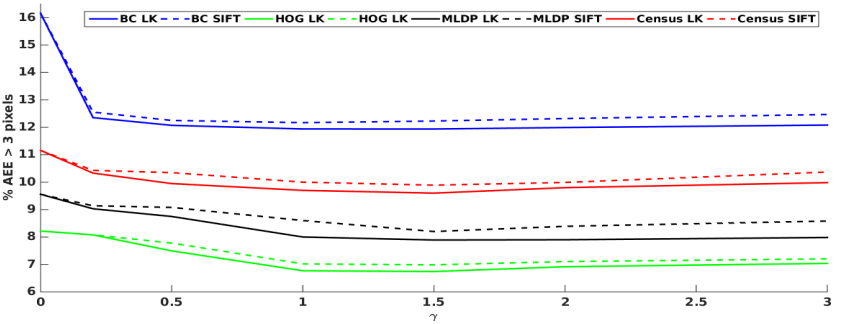
Figure 5.4: Comparison between ground truth (red) and estimated optical flow (green) for sequence 150 of the KITTI training dataset for some feature points. (a) Using epipolar constraint based on 7-points method. (b) Using brightness constrain. (c) Estimated epipolar lines.

flow which can result in significant errors at homogeneous regions. Moreover, the inaccuracy of the estimated fundamental matrix degrades the results for increasing values of γ.

The average of the end-point error (AEE), the average of the angular errors (AAE) and the percentage of outliers are presented in table 5.2. It can be observed that the LK tracker combined with the 7-point method for the estimation of fundamental matrix gives the best results for all variations of data term. The best combination is HOG descriptor, LK tracker, and 7-point method. The the data term using HOG outperformed other data terms. The reason that 7-point



(a)



(b)

Figure 5.5: The average error for each sequence of the 194 sequences of the KITTI training dataset. (a) Percentage of outliers using 7-points algorithm. (b) Percentage of outliers using 8-points algorithm.

method outperforms the 8-point method stems from two facts. First, the 7-point method needs one point less than the 8-point method which makes it more robust against outliers. Second, rank deficiency of the fundamental matrix is directly taken into account during the fundamental matrix calculation process. Whereas, in the 8-point method the rank deficiency is enforced later indirectly.

Table 5.2: The effect of usage the epipolar line constraint. The average end-point error and average angular error estimated using 7-point and 8 points fundamental matrix with Lucas Kanade and SIFT.

		LK			SIFT		
		AEE (px)	AAE (°)	Outliers (%)	AEE (px)	AAE (°)	Outliers (%)
BC	7-pts	02.45 px	02.72°	11.80 %	02.53 px	02.76°	12.23 %
	8-pts	02.50 px	02.73°	11.94 %	02.62 px	02.90°	12.53 %
HOG	7-pts	**01.52 px**	**01.72°**	**06.68%**	01.65 px	01.84°	06.99 %
	8-pts	01.56 px	01.74°	06.74 %	01.65 px	01.84°	06.99 %
MLDP	7-pts	01.69 px	01.73°	06.68 %	01.71 px	01.88°	08.01 %
	8-pts	01.74 px	01.76°	07.89 %	01.75 px	01.93°	08.20 %
Census	7-pts	01.95 px	01.87°	09.80 %	02.06 px	02.03°	09.85 %
	8-pts	02.06 px	01.93°	09.89 %	02.20 px	02.17°	09.93 %

5.5.3 Fundamental Matrix Re-estimation

We conduct the enhancement of the fundamental matrix starting at different levels of corse-to-fine pyramids. In this regard, we used the combination consisting of the 7-point method, the LK feature trackers and the HOG descriptor, which provided the minimum errors. Figure 5.6 presents the errors concerning the starting level of enhancement. It can be seen that the update of the fundamental matrix results in a good improvement, only if algorithm is applied in fine levels with high-resolution. Nevertheless, if the matrix is updated at the coarsest level, the errors noticeably increased. One critical issue is an incorrect initial guess of the fundamental matrix at coarse levels causes a divergent, and therefore, the error increases accumulatively. The reason is the estimation of the fundamental is usually sensitive to measurement noise more than two pixels for the 7-point method which results in large errors in the estimation of the fundamental matrix.

5.5.4 Challenging Sequences

In this experiment, we measure the effect of applying the epipolar constraint with a weight of $\gamma = 1.5$. We chose challenging sequences from KITTI training dataset which have fewer texture regions, repeated patterns, strongly varying lighting conditions, and many non-Lambertian surfaces. Tables 5.3 and 5.4 show a comparison between the estimated optical flow based on the HOG descriptor and

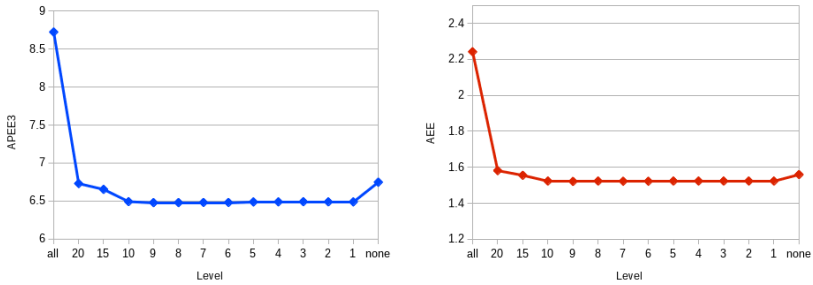


Figure 5.6: The percentage of average endpoint error more than 3 pixels and the average endpoint error based on the updating fundamental matrix starting at different levels.

the BCA. The epipolar line constraint succeeds to get a better optical flow and the AEE and the percentage of outliers are significantly reduced. Some images of these sequences are shown in figures 5.7 and 5.8.

Table 5.3: The effect of the usage of the epipolar line constraint on the AEE in pixels (px) applied on challenging sequences of KITTI dataset. $\gamma = 0$ means no epipolar was used, while $\gamma = 1.5$ is of the epipolar constraint in the data term.

	γ	10	19	39	70	71	84	114	178
BC	0.0	18.50	08.03	16.80	15.65	15.26	09.27	15.58	10.48
	1.5	04.62	04.98	12.34	03.64	00.60	00.73	02.72	02.76
HOG	0.0	01.75	01.14	02.84	01.38	00.50	01.12	00.98	01.53
	1.5	01.21	01.07	01.45	01.27	00.48	00.92	00.96	01.32

5.5.5 KITTI Evaluation

The proposed method was evaluated using the KITTI 2012 flow dataset. For testing the effect of the epipolar line constraint on the accuracy of the optical flow estimation, we did not use the weighted median filter explained in chapter 3 and 4. Meanwhile, we tested the proposed method using HOG descriptor and without using the epipolar line constraint and we got the results shown in tables

Table 5.4: The effect of the usage of the epipolar line constraint on the outliers (%) applied on challenging sequences of KITTI dataset.

	γ	25	54	74	101	113	135	163	181
BC	0.0	38.87	15.87	84.31	57.78	12.37	55.45	14.02	54.95
	1.5	37.81	15.86	74.94	48.74	09.67	48.05	11.80	55.70
HOG	0.0	30.57	23.33	50.77	19.42	22.97	14.58	19.31	41.71
	1.5	13.01	14.79	42.40	11.41	16.20	07.80	12.18	31.49

5.6. It can be seen from table 5.5 that the percentage of outliers is significantly decreased when we used the epipolar constraint. Table 5.6 shows comparisons with recently published methods.

Table 5.5: KITTI 2012 evaluation (percentages of outliers) of the epipolar texture constraint using the HOG with epipolar constraint $\gamma = 1.5$ and without the use of the epipolar constraint $\gamma = 0.0$.

	$\gamma = 1.5$	$\gamma = 0.0$
2 px	09.51 %	13.41 %
3 px	06.95 %	10.87 %
4 px	05.79 %	09.58 %
5 px	05.06 %	08.71 %

Table 5.6: The percentages of outliers and the AEE of some of the state-of-the art methods on the KITTI 2012 dataset.

Method	Outliers (%)	AEE (px)
GC-BM-Bino [KL12]	18.83 %	5.0 px
TF+OFM [KT15b]	10.22 %	2.0 px
MLDP-OF [MRM+14]	8.67 %	2.4 px
TVL1-HOG [RMG+13]	7.91 %	2.0 px
EpicFlow [RWHS15b]	7.88 %	1.5 px
MEC-Flow (Proposed method)	**6.95** %	1.8 px

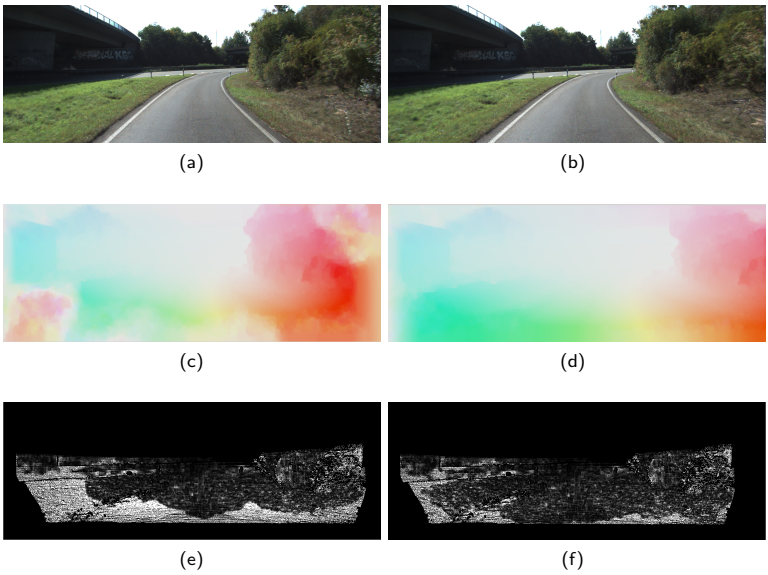


Figure 5.7: Optical flow estimation. (a) and (b) are frame 000025_10 and frame 000025_11 of sequence 25 of KITTI training dataset. (c) and (e) estimated optical flow and AEE error map without epipolar constraint, while (d) and (f) are with epipolar line constraint.

5.6 Conclusion

In this chapter, we explained how to integrate other constraints in the proposed variational optical flow model which is proposed in chapter 4. Therefore, we derived the necessary formulation of the epipolar line constraint for an uncalibrated camera. We proposed a new constraint based on the epipolar geometry along with different data-terms, different feature tracking, and fundamental matrix estimation methods. The optimal combination of different sub-algorithms and the optimal weight were found based on the KITTI 2012 training dataset. Furthermore, we evaluated the results for the test sequences on the KITTI 2012 portal. Moreover, we investigated the iterative enhancement of the fundamental matrix. The

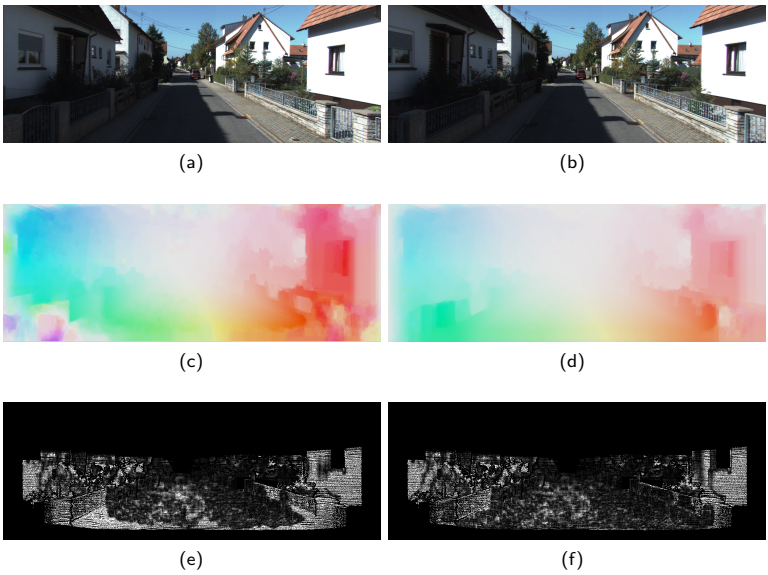


Figure 5.8: Optical flow estimation. (a) and (b) are frame 000054_10 and frame 000054_11 of sequence 25 of KITTI training dataset. (c) and (e) estimated optical flow and AEE error map without epipolar constraint, while (d) and (f) are with epipolar line constraint.

integration of the epipolar line constraint in the differential optical flow framework decrease the average end-point error to 36.06% and it reach 50% in many cases. Next chapter, we introduce a new method to analyze the motion by detecting and tracking multi moving objects.

6 Proposed Real-time Multi-Objects Tracking

6.1 Introduction

Motion analysis is a crucial problem and a challenging task for a mobile robot to perform tasks in a dynamic environment. On the one hand, the sensor data comprises two independent blended motions which are the camera ego-motion and the motions of moving objects [JS10]. Usually, the ego-motion of the camera is compensated, and the remaining motions are considered as the motion due to moving objects. On the other hand, sensor data is contaminated by various types of noise such as inadequate lighting conditions, camera distortion, and changing the shape of objects. An optimal camera ego-motion compensation task is a hard task, and it is usually scarcely realizable. Therefore, it is considered as another source of noise as it supplements another type of uncertainty to the system [JS10].

This chapter is organized as follows: Section 6.2 presents the main concepts and background of the multi-object tracking problem. The state-of-the-art and the related work are also discussed. In section 6.3, we acquaint with the proposed algorithm and introduce the multi object tracking approach. Section 6.4 shades the light on the motion detection algorithm, while the tracking of moving objects is introduced in section 6.5. A solution for occlusion handling is given in section 6.6. Furthermore, section 6.7 describes the camera motion stabilization algorithm. In section 6.8 the experimental results and the evaluation are reviewed. Finally, some concluding remarks are highlighted in 6.9.

6.2 Related Work

Recently, assortments of techniques have been proposed to track moving objects from stationary cameras based on the background modeling or the background subtraction. Zhu et al. [DZ12] proposed a real-time approach for short-term tracking of multiple objects. This approach uses a motion detection algorithm to segment moving objects from a stationary camera. The authors propose to use an adaptive background update algorithm that handles illumination changes and slow camera motion. However, this algorithm fails for complex scenes and fast camera motions. Furthermore, a variety of research efforts was dedicated to tracking a single object by keeping a model of low-level features and searching for its new location in each frame (see [SGG09]).

Enzweiler et al. [EG11] proposed a multilevel mixture-of-experts approach with the objective to improve the classification of pedestrian. This approach combines information from multiple features and cues. In turn, Talukder et al. [TM04] used a stereo camera and combined dense disparity with a dense optical flow to estimate the background motion and to detect moving objects. Moreover, Henriques et al. [HCB11] introduced a graph structure that encodes multiple-match events as standard one-to-one matches. As a result, it allows the computation of the solution in real time. Furthermore, Wojek et al. [WWR+13a] proposed a probabilistic 3D scene model. This model integrates a geometric 3D reasoning together with multi-class object detection, an object tracking, and a scene labeling.

In case of moving camera, most of the real-time object tracking approaches use sparse features or assume flat scene structures [ST94], [ML11]. On the one hand, sparse optical flow process few feature points, such as corners or key-points. However, sparse features are not sufficient to infer valuable information of an object, such as shape. On the other hand, dense optical flow diffuses all pixels through a regularization [BM11], [SBK10] and reveals information about objects. Although dense optical flow is a reliable source of information for motion segmentation, object detection, and flow-based tracking, its high computational cost affects the real-time performance, especially for high-resolution cameras. One solution is to combine sparse feature and dense optical flow to track moving

objects. Therefore, Peter et al. [ST06] proposed an algorithm that integrates feature points tracking with dense optical flow fields to a set of spatially dense and temporally smooth trajectories. Furthermore, Rubinstein et al. [RLF12] proposed an algorithm to estimate long-range motion trajectories and used them as an initial estimate for a global solution.

Several approaches for estimating dense optical flow concerned with estimating accurate and robust flow fields under various conditions such as [MRM+14] and [RMG+13]. However, such approaches neglected the processing time, and they are not suitable for real-time applications. Nevertheless, some works have been introduced to decrease the processing time, but most of them are based on the internal structure of the algorithm itself and are not applicable to other techniques. That techniques tried to use the parallel processing nature of the algorithms and used GPUs or other hardware to do the computation in parallel.

In this chapter, we introduce a real-time approach that works with a stationary as well as a moving camera to detect and track moving objects in a dynamic scene. The proposed algorithm uses sparse and dense optical flow as input to detect moving objects. It uses sparse optical flow to identify moving regions and dense optical flow to detect and track moving objects. Afterwards, we applied a planar parallax motion constraint which assumes that independently moving objects undergo pure translation. The Kalman filter is used to track the orientation of the moving objects based on the dense optical flow.

6.3 The Proposed Approach

For an optical flow-based application to work real-time, it should calculate the optical flow and do other processing such as segmentation, interpretation or other high-level analysis within a few milliseconds. Unfortunately, most of the optical flow algorithms require more time to estimate the dense flow only, even if the GPU hardware was used. Thus, a need for a new technique for calculating the optical flow is a must.

The proposed solution is to calculate the dense optical flow only if it is necessary. The solution is to detect and divide the motion into moving and static segments and calculate the optical flow only for each moving segment in parallel. For instance, for stationary camera applications, it is more critical to calculate the optical flow for the regions with moving objects than for the whole scene. Typically, for a moving camera in a dynamic scene, if the ego-motion is known using an inertial sensor or other visual techniques, the optical flow algorithm should not be applied to the background. Using this strategy the results show a significant reduction of the processing time for the overall system.

Thus, in this chapter, we propose a new algorithm for tracking multiple moving objects from a moving camera based on optical flow with real-time performance. The proposed algorithm uses a mixed-sparse-dense flow field to gain the benefits of the reliability, accuracy and real-time performance. We suppose rigid objects and assume a single segment can represent an object in a scene. The proposed algorithm contains two main stages. The first stage is to initialize and detect moving areas hypotheses. Therefore, to achieve this task, the camera motion is stabilized based on the homography matrix to reduce the effect of the arbitrary motions introduced by camera pans and jitters. Afterwards, we segment the motion to the foreground (moving objects) and the background by dividing the image into isolated moving regions using a blob detection technique. In the second stage, a dense optical flow algorithm is applied in parallel only to each moving region from the two frames using multi-threading processing. Correspondingly, we segmented moving objects in each region based on the parallax constraint and the iterative region growing algorithm. Moreover, a tracker using a Kalman filter for each object is calculated based on the motion orientation of its pixels. Besides, information on objects such as motion models, center points, and shapes are updated at each frame to track objects.

In addition, for each object, the color histogram is calculated to be used, if an object is occluded or if the object stops moving. In turn, short-term memory has been used to store the extracted information about all moving objects. Figure 6.1 shows the overall algorithm.

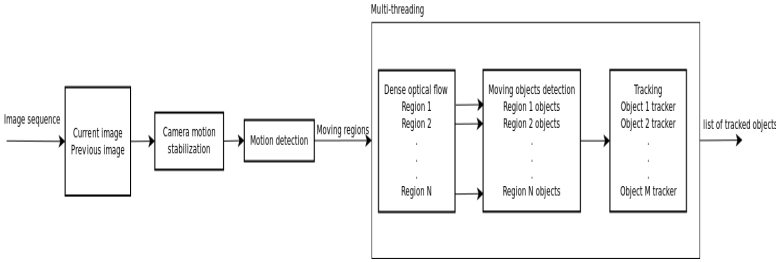


Figure 6.1: The proposed architecture for multi-objects tracking.

6.4 Motion Detection

The task of the motion detection module is to segment moving regions in an image and initialize objects to be tracked. In the proposed algorithm, we use two different methods. The first method uses the background subtraction (BS) after calculating the absolute difference between the frames at times t and $t-1$. Afterward, we apply a blob detection algorithm to the resulting image to segment moving areas. Later, we create a rectangle around each blob, and we store that region in a list of moving regions. This method offers not only a rapid detection of new objects entering the scene, but also it needs only little processing time. Conversely, in the case of a moving camera, it is sensitive to the error of the camera motion stabilization.

The second motion detection method is based on the sparse optical flow between the current frame t and the previous frame $t-1$. The algorithm starts with calculating feature points in the frame at $t-1$ using a method called good features to track [ST94]. Afterward, the algorithm [LK81] is applied on these features to calculate sparse flow. In addition, a merging procedure based on the magnitude of the optical flow vectors in each segment is used locally to merge the moving neighborhood pixels. A predefined window is centered at each pixel,

and the overlapping windows are merged to obtain a larger window. The merging procedure can be formulated as:

$$\Omega_i = \bigcup_{j=1}^{P} W_j, \tag{6.1}$$

where Ω_i denotes a moving region, W_j denotes a centered window, and P denotes the number of overlapping windows. Experimental results show that the sparse optical flow method is more robust than background subtraction in different challenging environments such as severe weather conditions and sudden brightness changes. Furthermore, it is more robust to noise and camera motion errors. However, one drawback of the sparse optical flow method is the detection of new objects entering the scene occurs only after the detection of some feature points on that objects. Figure 6.2 shows the results after applying the sparse optical flow method to detect moving regions.

6.5 Motion Estimation and Multi-Object Tracking

This module extracts accurate information about objects. Thus, for each moving region, a bounding box is located in the two images, and a dense optical flow algorithm is applied. The calculation of the dense optical flow is done in parallel for all moving regions by using multi-thread processing. We used the optical flow model Eq.(4.8) by minimizing the following objective function $E(u, v)$,

$$\min_{u,v} \; E(u,v) = \sum_{i=0}^{N} \sum_{\hat{\Omega}_i} \left(\lambda \, \rho(x, y, u, v)^2 + \parallel \nabla u \parallel + \parallel \nabla v \parallel \right) \tag{6.2}$$

where N is a number of moving regions. As discussed in chapter 4, ρ is a residual function between two texture descriptors. In turn, u and v are the horizontal and vertical optical flow components, and λ is the weight of the data term. $\parallel \nabla u \parallel + \parallel \nabla v \parallel$ is the smoothness constraint which yields the smoothness term which also called the total variation. The objective function in Eq.(6.2)

is solved iteratively after dividing it into two parts using a quadratic coupling term [MBM14]. The data-term is written as:

$$\min_{u,v} \ \mathcal{E}_d(\mathbf{w}) = \sum_{i=0}^{N} \sum_{\hat{\Omega}_i} \left(\lambda \, \rho(x, y, \mathbf{w})^2 + \frac{1}{2\theta} (\mathbf{w} - \hat{\mathbf{w}})^2 \right) \tag{6.3}$$

where $\mathbf{w} = [u, v]^T$ and $\hat{\mathbf{w}} = [\hat{u}, \hat{v}]^T$ is the auxiliary optical flow vector and θ is a threshold. Similarly, the regularization term is written as:

$$\min_{\hat{u},\hat{v}} \ \mathcal{E}_s(\hat{\mathbf{w}}) = \sum_{i=0}^{N} \sum_{\hat{\Omega}_i} \left(\frac{1}{2\theta} (\mathbf{w} - \hat{\mathbf{w}})^2 + \parallel \nabla \hat{u} \parallel + \parallel \nabla \hat{v} \parallel \right) \tag{6.4}$$

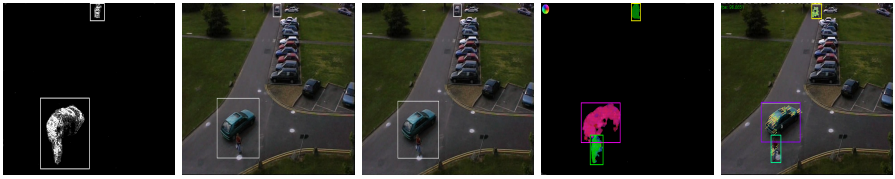
We applied the same procedure explained in chapter 4 to solve for Eq. (6.3) and Eq. (6.4). The resulted optical flow is assumed as a hypothetical region holding one or more moving object. Therefore, we applied a motion segmentation algorithm on the estimated optical flow.

We used the parallax constraint by assuming that all optical flow vectors on an object have similar directions. The proposed algorithm is implemented using an iterative region growing scheme. It works as follows: the starting point is at the center of the bounding box. We assign a label and search for similar pixels in the neighborhood of that point, which move in a similar direction as the current pixel. If one pixel is found, the same label will be assigned to it, and the area of the region is increased in the direction of the optical flow vector at that point. This step is repeated until it reaches the motion border of the object. For the unprocessed pixels, we select a random pixel and start a new segmentation procedure. Besides, the small segments are merged into larger segments, if they lie inside larger segments or if they overlap other segments. Therefore, in the case of non-rigid objects (e. g. people), the moving parts (e. g. hands and legs) are merged to the body segment to create more prominent segments include all parts. However, we have not used these segments to update the objects tracker in the next step. Later, we consider each segment as a moving object. The center (x_c, y_c) of each object is calculated, and a tracker for each object is created. For object tracking, we use the object center point (x_c, y_c) and vector $T = (u_c, v_c)$

which represent the average direction of the optical flow vectors of all pixels inside a segment.

$$T = \frac{\sum_{i=0}^{P} \mathbf{w}_i}{P}, \tag{6.5}$$

In the proposed algorithm, we assume that the object's motion between two consecutive frames are not too large; thus it is likely that the position is only slightly changed in the new frame. Therefore, we use the current position of an object and extend its holding region in all directions based on the estimated magnitude of vector $||T||$. Afterward, we estimate the optical flow using the regions from the two frames. Meanwhile, the tracker T is updated using the new location which is calculated from the estimated optical flow. Figure 6.2 shows an example of the proposed motion estimation and tracking. A Kalman filter is used to predict and update the object tracker to deal with noisy measurements.



(a) (b) (c) (d) (e)

Figure 6.2: Results of multi-objects tracking stages: (a) Resulting moving regions after using the motion detection technique, (b) detected region in the frame $t1$, (c) detected region in the frame $t2$, (d) estimated dense optical flow for each detected moving object and (e) tracking windows for each object.

6.6 Occlusion Handling

For dealing with occlusion of objects, we created an object model based on its motion, location, and appearance using the histogram of the HSV color space for each object at each frame. Let's assume that X is objects list and each object X_i

in the list has the following information: the center point $c_i = (x, y)$, appearance model T_i, motion model $\mathcal{M}_i$, object shape ς_i and state S_i. For all moving objects which are still in the scene, the state S_i is set to $tracked$, while for objects which are occluded, stop moving or leave the scene, the state S_i is set to $drift$.

$$S_i = \begin{cases} \text{tracked} & \text{if the object still moving} \\ \text{drift} & \text{if the object stop moving or leave the scene} \end{cases} \tag{6.6}$$

For all new objects entering the scene, we do object matching with the objects which have state $drift$ tracked only. The matching is based on the maximum cross-correlation of the template. The nearest object concerning the center point will be considered. The object matching is done only for new objects entering the scene and only if there is some drift track still in the memory as the drift tracked is removed after a predefined period.

6.7 Camera Motion Stabilization

In the proposed algorithm, we assume a planar stationary background of the scene, or the background can be approximated by dominant single or multiple planes, or an orthographic camera model. Typically, these assumptions can limit the applicability of the proposed algorithm in the field of robot navigation. However, the goal here is to prove the concept of the detection and the tracking of moving objects and not to precisely compensate the ego-motion. Therefore we developed an efficient stabilization algorithm based on the sparse optical flow and the homography matrix. The proposed algorithm uses only some sparse optical flow points to estimate a perspective camera motion model. However, the proposed algorithm is not limited to a specific type of camera motion stabilization approach.

The homography matrix describes the perspective transformation between a source and a destination planes. Therefore, we used the homography matrix to estimate the perspective transformation and stabilize the camera motion. Therefore, to calculate the homography matrix, a minimum of four feature points are required on the source frame and the corresponding points on the destination

frame. Among these four points, three of them must be non collinear.
For the detection of feature points, methods such as SIFT, SURF, or ORB detector can be used. However, in this work, we choose a simple and an efficient method referred as "good features to track" [ST94] to extract feature points. For estimating the optical flow of the feature points, we used the sparse optical flow algorithm [LK81] implemented in a coarse-to-fine scheme [MMM15]. The feature points in the source frame and the corresponding points in the destination frame are refined, and the outliers are removed. We use a forward-backward validation method to find the component of a flow vector u and v by optimizing the texture constraint twice as follows:

$$\mathbf{S}_1(x, y) - \mathbf{S}_2(x + u, y + v) = 0 \tag{6.7}$$

$$\mathbf{S}_2(x, y) - \mathbf{S}_1(x + u, y + v) = 0 \tag{6.8}$$

We considered only points which have the same absolute value of u, v in both equations, while other points are considered as outliers. Given a list of points p_i in the source frame and correspondences point p_j in the destination frame, the homography matrix is calculated by the following equation:

$$\mathbf{p}_i = H\,\mathbf{p}_j \tag{6.9}$$

$$\begin{bmatrix} x_i \\ y_i \\ w_i \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & 1 \end{bmatrix} \begin{bmatrix} x_j \\ y_j \\ w_j \end{bmatrix} \tag{6.10}$$

The points are represented in homogeneous coordinates. Hence, a 2D point in the image plane (x, y, w) donates as $(x/w, y/w)$ in the Cartesian coordinates. Minimum four points correspondence are sufficient to calculate H, as each point provide two equations.
For optimizing the homography matrix, we used the RANSAC [FB81] to find the best homography model. RANSAC tries many different random subsets consists of four corresponding point pairs. We used the least-square method to minimize

the back projection error r_i to find the homography matrix and calculate the inliers and the outliers as follows:

$$r_i = ||p_i - Hp_j|| \tag{6.11}$$

As a result, the resulted homography matrix has the minimum number of outliers. Conversely, the homography matrix is further refined to reduce the re-projection error after using only the inliers points. Therefore, we calculate the standard deviation of the inlier distance. We used those points which have smaller distance than a threshold to re-estimate the homography matrix based on the Levenberg-Marquardt method. Finally, the destination image is warped toward the source image after applying a perspective warp.
Figure 6.3 shows the result after applying an absolute difference between the source image and the warped destination image. It is evident that the image warping step reduces the effect of the camera motion and the moving objects are highlighted. However, it is well known, that the homography is a limited method for motion compensation and in the case of global camera motion; it is only valid for a planar scene, or an orthographic camera [DRSS12]. In the case of a non-planar scene or a perspective camera, the homography is valid only for rotational camera motion. If the camera undergoes translational motion while observing a non-planar scene, the homography is not valid. Therefore, under these circumstances, the homography is valid only for some areas inside the images. As a result, the areas for which the homography is not valid appear as falsely detected moving objects.

6.8 Experimental Results

The algorithm is implemented using C++ and the OpenCV library. We performed the computation on a PC with an Intel(R) Xeon(R) 2.67 GHz CPU (4 cores) and a GeForce GTX 580 graphics card. The arguments in the following section are based on the author's work [MBM14].

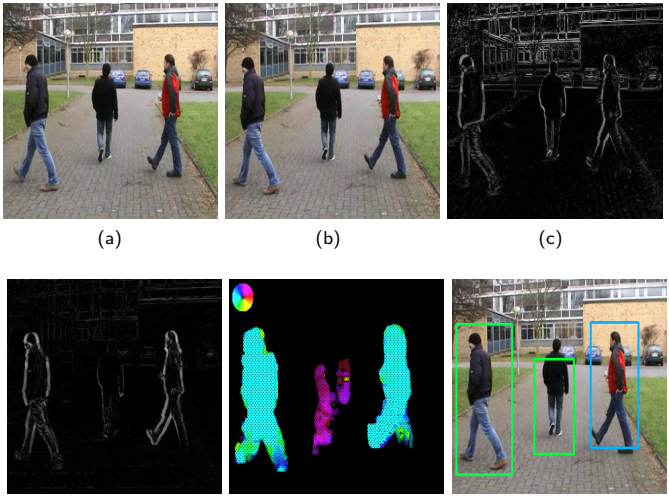


Figure 6.3: Results of the camera motion stabilization and tracking applied on the *Vid_I_person_crossing* sequence. Images 6.3a and 6.3b are two consecutive frames. Images 6.3c shows the absolute difference between image 6.3a and image 6.3b without applying the camera stabilization algorithm. 6.3d shows the absolute difference after applying the camera motion stabilization algorithm. 6.3e shows the estimated optical flow. 6.3e shows the tracking results.

6.8.1 Multi-Objects Tracking Accuracy

For the quantitative evaluation, the standard CLEAR MOT metrics [SBB+06] has been used to assess the performance of the proposed algorithm [MBM14]. Hence, the Multi-Object Tracking Accuracy (MOTA) takes into account false positives, identity switches (when the tracker exchange between objects) and missed targets. It combines them into a single value and does a normalization to the range from 0% to 100%. Accordingly, a match between a tracker output and the ground truth is defined as $> 50\%$ intersection-over-union of their bounding boxes. Consequently, the Multiple Object Tracking Precision (MOTP) merely is

the average distance between ground truth and estimated targets as a measure of localization accuracy. The closely related MODP equates the overlap over all frames. The related Multi-Object Detection Accuracy (MODA) only checks for missed targets and false positives. However, it does not penalize trajectories switching from one target to another.

6.8.2 Datasets

We evaluated the accuracy and the performance of the proposed algorithm based on three challenging datasets: Town Center [BR11], PETS 2001 [BFF09], and the first view of the S2.L1 sequence from the PETS 2009/2010 benchmark [BFF09]. All of the three datasets are publicly available, and they have ground truth. The images of the Town Center dataset have a resolution of 1920×1080 at 25 fps, they are captured using a stationary camera. The PETS 2001 and PETS 2009/2010 datasets consist of images with a resolution of 768×576 at seven frames per second captured by a static camera. We compared the proposed algorithm with some of the state-of-the-art methods [BFF09], [BRL$^+$11], and [ASR12] using the PETS 2009/2010 benchmark [BFF09] using the ground truth from [Milml]. Figure 6.4 shows qualitative results of the proposed algorithm applied on different datasets. It can be seen that the proposed algorithm succeeded to detect and track moving objects. Table 6.1 compares the results of the proposed algorithm

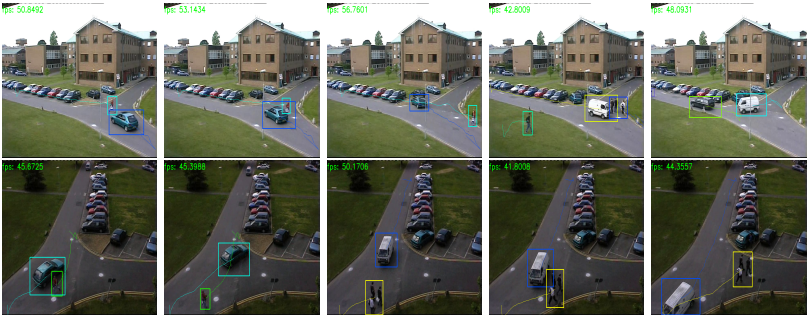


Figure 6.4: Results of the multi-objects tracking based on optical flow. Row(1) PETS 2001 training. Row(2) PETS 2001 test.

with some of the state-of-the-art methods. As can be seen, the proposed algorithm provide competitive results to the state-of-the-art methods. It achieves higher percentages of The MOTA and MODA.

Table 6.1: Comparison of the proposed method to two state-of-the-art method on PETES's09 S2.L1 [BFF09]. The results of [BFF09], [BRL$^+$11], and [ASR12] were extracted from Tabel 2 in [ASR12].

2D performance	MOTA	MOTP	MODA	MODP
Berclaz et al. [BFF09]	82%	56%	85%	57%
Breitenstein et al. [BRL$^+$11]	75%	60%	89%	60%
Andriyenko et al. [ASR12]	89.3%	56.4%	90.8%	57.3%
Proposed approach	84%	55%	85%	56%

6.8.3 Objects Tracking with a Mobile Robot

We tested the algorithm on mobile robots using two scenarios. In the first scenario, we used the GETLab mobile robot (GETbot) which is a typical four-wheeled robot. In this scenario, the GETbot navigated in a rescue robotic arena of the GETLab and was mounted with a camera which has a resolution of (320×240) at 25 fps. Here motion is represented as a signal of life. Therefore, the GETbot searched for surviving victims who are trying to get the attention of the robot by waving their hand (see the first row of Figure 6.5). As can be seen, the algorithm detect the waving hands successfully. In turn, in the second scenario, we used the dataset *Vid_I_person_crossing* sequence, which is publicly available with OpenCV library for a teleoperated robot moving in a dynamic environment in front of several people crossing the way in the front of the robot. To evaluate the algorithm using these two scenarios and due to the lack of the complete ground truth for all moving objects, the objects were marked by creating bounding boxes around moving objects and have been used as a ground truth. The evaluation of the algorithm was done by computing the MOTA, MOTP, MODA, and MODP using the intersection between the ground truth objects and the detected objects by the algorithm. Table 6.2 shows the results of the algorithm. As can be seen, the proposed method has high scores for the MOTA and the MODA in both

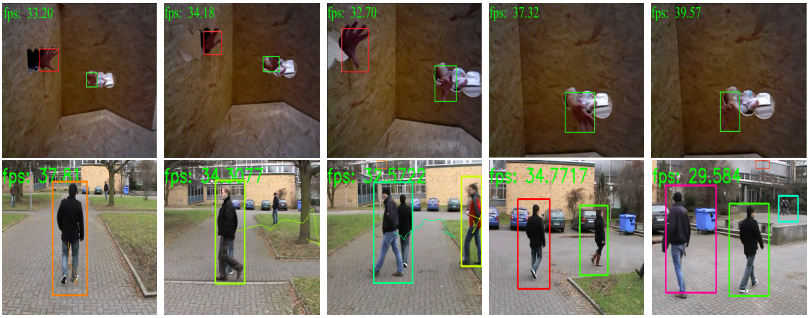


Figure 6.5: Results of the multi-objects tracking based on optical flow. Row(1) GETbot search victims sequence. Row(2) *Vid_Ipersoncrossing* sequence.

scenarios. However, the MOTP and MODP have low score due to the error of the camera motion stabilization.

Table 6.2: Accuracy of the object tracking algorithm applied to the victim detection and the *Vid_Ipersoncrossing* scenarios.

2D performance	MOTA	MOTP	MODA	MODP
Moving robot	83 %	57 %	87 %	50 %
Search victim	89 %	60 %	90 %	60 %

6.8.4 Real-Time Performance

To test the real-time performance, we used several datasets with different resolution and objects. We tested the execution time of estimating the optical flow only by applying the algorithm to the whole image with CPU and GPU [MRM+14]. Furthermore, we calculated the optical flow using the moving regions technique proposed in this work with a single CPU and GPU. Moreover, we tested the algorithm using multi-threads in a CPU and GPU. The results are shown in figure 6.6. The evaluation results show a significant decrease in the processing time when processing moving objects. Although the performance depends on the

number and size of the moving objects, as shown in the moving robot sequence, the overall processing time significantly decreases and is always lower than the processing time for the whole image. For the Town Center sequence, using the moving objects techniques with a single CPU, the processing time is about 11 times faster than the processing time for the whole image, while with the GPU it is three times faster. Using multi-threading processing is 30 times (CPU) and four times (GPU) faster than the processing time for the whole image using CPU and GPU, respectively. Furthermore, the proposed algorithm gives the possibility to process more significant objects on GPU and smaller objects on the CPU, which gives the best performance.

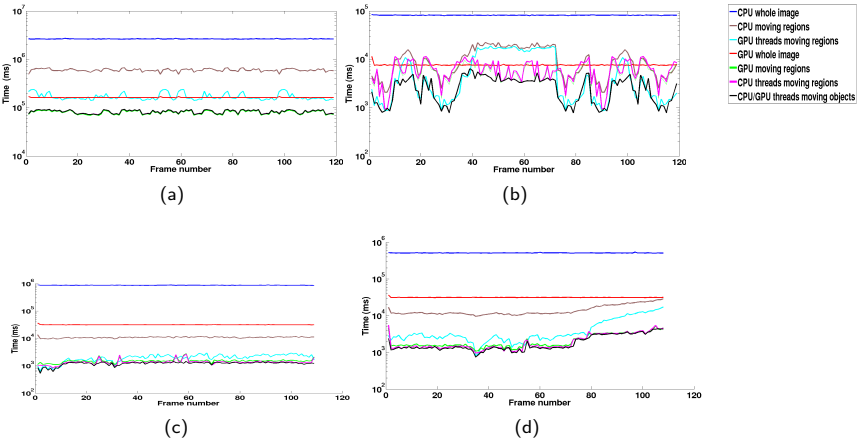


Figure 6.6: The processing time of the dense optical flow [MRM+14] estimation using different techniques applied to the first 120 frames of: (a) Town-Center [BR11]. (b) *Vid_Ipersoncrossing*. (c) PETS 2001 training. (d) PETS 2001 test.

We test the processing time for each step in the proposed algorithm. As shown in table 6.3, the motion estimation and tracking consume the most significant part of the processing time. However, the processing of moving regions is carried out using multi-threading processing. Thus the overall processing time is equal to the processing time of the most significant region in the scene if there are enough

threads for all objects. However, the camera stabilization step is controlled, and it is deactivated in the case of static camera datasets.

Table 6.3: The average processing time per frame in (ms) of different modules of the proposed algorithm using CPU/GPU multi-Threading.

	Camera motion stabilization	Motion detection	Motion estimation & tracking	Motion segmentation	Processing time
TownCenter (1920 × 1080)	-	11.051	94.424	0.174	95.360 (10.49 fps)
PETS 2001 Test (768 × 576)	-	2.439	6.575	0.061	6.922 (144.5 fps)
Moving Robot (320 × 240)	12.149	1.094	4.570	0.012	16.825 (59.4 fps)
PETS 2001 Training (768 × 576)	-	2.446	5.611	0.151	6.068 (164.8 fps)

6.8.5 Outdoor Scenarios

To test the performance of the proposed algorithm in outdoor scenarios, we conducted a test for the detection and tracking of construction workers and equipment. Indeed, detection and tracking of workers and equipment by autonomous vehicles is a crucial prerequisite for any onboard safety system aiming at preventing vehicle-pedestrian collisions. In this work, we test the proposed algorithm for detecting and tracking construction workers and equipment based on optical flow with real-time performance. As a result, all moving objects are detected, and a tracker for each object is created. The experimental results demonstrate that the proposed algorithm works appropriately and that there is a significant reduction in the overall processing time for detecting and tracking multiple moving objects in a scene. Figure 6.7 show the results of applying the camera stabilization algorithm on a sequence of images. The camera motion in these sequence was slowly and therefore, the calculation of the homography matrix was accurate enough to detect the moving objects. Figure 6.8 shows qualitative results of the proposed algorithm applied on in different outdoor scenarios contraction sites in the city of Paderborn in Germany. The author used a hand-held camera to capture videos with a resolution of (640 × 480). It can be seen that the proposed algorithm detects and tracks all moving workers and equipment.

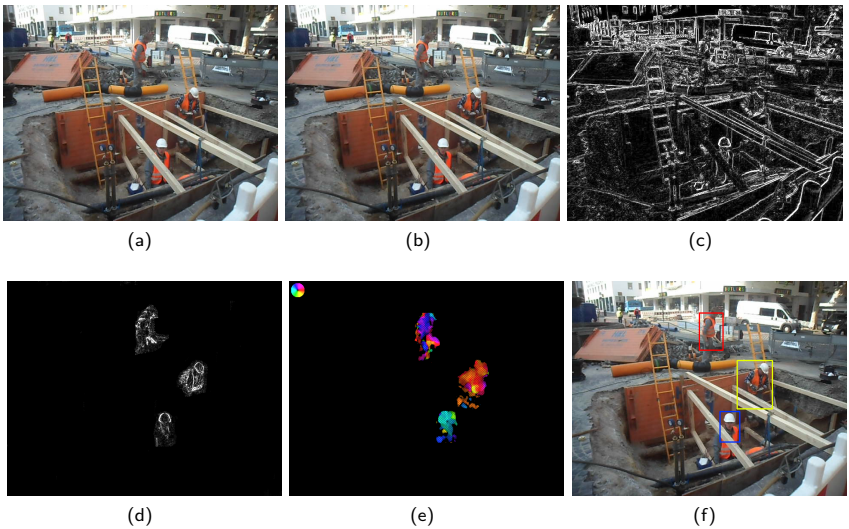


Figure 6.7: Results of the camera motion stabilization. (a) frame at time t. (b) frame at time $t+1$. (c) the absolute difference between frames (a) and (b) without applying the camera motion stabilization algorithm. (d) the absolute difference after applying the camera motion stabilization algorithm. (e) the estimated optical flow. (f) the tracking results.

6.9 Summary

In this chapter, we developed an algorithm using optical flow for doing real-time multi-object tracking. To optimize the processing time, we maintain the estimation of dense optical flow only for image regions where (moving) objects are present and not for the background or static objects. Thus, we used sparse optical flow at spatial feature locations to detect moving objects and subsequently used region growing to form objects hypotheses. Furthermore, the objects are represented by bounding boxes, and the tracking is performed based on dense optical flow computed within the bounding box. Further processing steps include camera motion stabilization and motion segmentation steps. Moreover, we tested the proposed algorithm with different scenarios using various datasets

and real applications. The experimental results demonstrate the efficiency of the proposed algorithm. The experiments showed a significant reduction in the overall processing time for detecting and tracking multiple moving objects in a dynamic scene. We have shown that the proposed algorithm can be used as a base for a high-level analysis for a dynamic scene such as searching victims or navigation in dynamic environments.

Figure 6.8: Results of the multi-objects tracking based on optical flow applied to different scenarios of construction sites.

7 Summary and Outlook

This chapter summarizes the contributions and achievements to dynamic scene analysis using optical flow and indicates the direction for future work.

7.1 Summary

The central topic of this thesis was developing an active vision system that can detect and track all moving objects in a dynamic scene in real-time. For this purpose, the optical flow was used as the primary source of information. In particular, we concentrated on the critical challenges of accurate, robust and fast optical flow estimation. The first milestone was to improve the estimation of large displacement optical flow. Hence, we proposed to update the multi-scale processing scheme through a coarse-to-fine technique to save the small details of objects which are small and fast and therefore affected by the linearization of the data term without needs to high processing power. The proposed algorithm uses points correspondences between feature points at each level, and the estimated optical flow is refined using these points. Afterward, the optical flow estimated at each coarse optical flow level is propagated to the finer level.

Illumination and appearance changes in particular are significant problems, since they contradict the traditional brightness constancy assumption that many methods used. Therefore, the second milestone of this work was to develop an optical flow algorithm that is robust against illumination changes and texture-less regions. Hence, we proposed an robust algorithm based on a texture constraint using local descriptors. To apply the texture constraint in the optimization of the total variational optical flow, it was essential to develop an update of the TV-L1 objective function in order to use a multi-channel descriptor. To represent textures

of an image, we proposed a novel descriptor called modified local directional pattern (MLDP), which encodes the direction of gradients in the form of a binary descriptor. Moreover, we used the histogram of the oriented gradient, which encodes the direction and magnitude of the gradient. Furthermore, we investigate the usage of the monocular epipolar geometry constraint for the calculation of optical flow in the case of texture-less regions.

The third milestone of this work was the development of an algorithm which uses the estimated optical flow to detect and track moving objects in real-time. All moving objects have to be separated from the static ones. Their movement direction and their speed have to be estimated, and a tracker for each object has to be computed for every object. Therefore, we optimized the proposed optical flow algorithm for texture constraint using parallel processing techniques on a CPU and GPU. For segmenting moving objects, we proposed to do camera motion stabilization to compensate the camera ego-motion. Hence, the motion detection can be applied as a post-processing step to detect moving regions. The calculation of the dense optical flow is done to those regions only. Afterward, a 2D motion segmentation based on parallax constraint is applied, and a Kalman filter is used to track each object.

The proposed algorithms works with a static as well as a moving camera, and the results show the successful analysis of dynamic scene. The algorithms work robust to detect, estimate, and track moving objects in indoor and outdoor environments. Several experiments and applications have been conducted to test and evaluate the proposed algorithms extensively. The results have shown that the proposed algorithms in this thesis outperformed the state-of-the-art approaches based on the standard benchmark datasets. Table 7.1 presents comparisons among different approaches for multi-object tracking based on various criteria. We used criteria such as motion of the camera, detection and tracking, real-time performance, and post knowledge about the environment. It can be seen from table 7.1 that the proposed algorithm successfully fulfill most of the criteria. However, it does not have the ability to classify and recognize objects which can be integrated in the future work.

Table 7.1: Comparison among dynamic scene analysis approaches

	Pedestrian Classification [EG11]	Tracking-by-Detection et al., [BRL+09]	Traffic Scenes [WWR+13b]	Video Analysis [OMB14]	Proposed Method [MBM14]
Moving camera	√	√	√	√	√
Moving objects detection & tracking	X	X	√	√	√
Real-time performance	√	X	X	X	√
No traning data	X	X	X	√	√
No Perior Knowlege about the scene	X	X	X	√	√
Active vision system	√	√	X	X	√
Object classification	√	√	√	X	X
Handle robustness	X	√	√	√	√

7.2 Applications

Although robot navigation motivates us to assist autonomous driving, the proposed algorithms were successfully used for various other applications to test the robustness and the reliability. The following list shows topics, tasks, and projects that used the results of the proposed algorithms to achieve their tasks:

- Semantic motion segmentation [Dau19].
- Gesture-based control system for a robot arm [Lu19].
- Detection and tracking of the stork bird in Paderborn.
- Motion detection tasks in the RoboCup Rescue German Open 2013, 2014, 2015, 2017, 2018 and RoboCup Rescue Championship 2016 competitions.
- Moving object detection for a non-stationary camera [Ngu18] and [Bri18].
- Optical flow estimation using image segmentation and texture constraint [Rai17].
- Classification of moving objects [Dod16].
- Semantic annotation of object representations [Lan15] and [Rol16].
- Implementation and evaluation of variational optical flow Based on texture constraints [Vog15].
- Detection and tracking of construction workers and equipment [BMM14].

- Human action recognition using optical flow and Hidden Markov Model (HMM) [Moh14].
- Application of optical flow In automation, grasping of moving objects. [MM12c].

7.3 Outlook

In this thesis, robust algorithms based on local texture descriptors for the estimation of optical flow were proposed. After completion of the work presented here, many new directions have become visible that need to be investigated further for reaching an ultimate model of dynamic scene analysis. The dissertation concludes with indications of such directions. One ad-hoc possibility to achieve robust optical flow estimation is to use image segment instead of local descriptors; however, modifications w.r.t optimization algorithm would be necessary. In the context of multi-objects tracking, extending the applicability of the proposed algorithm to be used in real rescue robots moving in a 3D environment might be an interesting topic (see, e.g., [GKN$^+$18] and [MTM18] for an overview). Moreover, an interesting topic is to develop an multi-object tracking that cope with varying appearance, limited side view, deformed object shapes, inner-class variation, unknown motion, occlusion, and other influences on the objects.

Another interesting topic could be the usage of deep neural networks to estimate the motion and track multi-objects. In this regard, the texture constraint can be used to construct an objective function to be used in unsupervised manner. Most existing deep learning based methods for object detection and tracking assume that the models are trained off-line in advance, which requires the entire training dataset to be available before the training and application. After training a model, its parameters do not change. Therefore, updating the models on-line is a challenging problem [YCREB19, GAS$^+$19, SPLH17]. In turn, traditional on-line learning methods often optimize predictive models over a stream of data instances sequentially [KMM12,XZH19]. Most existing on-line learning algorithms are designed to learn shallow models (e.g., linear or kernel methods) with an on-line convex optimization, which cannot learn complex nonlinear functions in

complicated application scenarios. Hence, another interesting topic is to consider this challenge and to develop new connections between traditional algorithms for object detection/recognition and tracking with recent advances in deep-learning based techniques.

A collision avoidance framework for the robots moving through a dynamic environment can be developed. An RGB camera can be used as the primary sensing modality. Recently, new approaches using a single RGB camera show reliable results for pose estimation of human [CSWS17, MSS+17] as well as objects [ZSI19, WXZ+19]. Robots are required to deal with dynamic scenes containing moving objects with unexpected behaviors. To achieve this goal, analyzing and understanding of the dynamic environment surrounding a robot is essential.

Bibliography

[ACPP05] Antoniol, G.; Ceccarelli, M.; Petrillo, P.; Petrosino, A.: An ICA Approach to Unsupervised Change Detection in Multispectral Images. In: *15th Italian Workshop on Neural Nets, Biological and Artificial Intelligence Environments*, Springer, 2005, pp. 299–311

[ADB14] Ali, Sharib; Daul, Christian; Blondel, Walter: Robust and Accurate Optical Flow Estimation for Weak Texture and Varying Illumination Conditions: Application to Cystoscopy. In: *4th International Conference on Image Processing Theory, Tools and Applications (IPTA)*, IEEE, 2014, pp. 1–6

[ADGB16] Ali, Sharib; Daul, Christian; Galbrun, Ernest; Blondel, Walter: Illumination Invariant Optical Flow Using Neighborhood Descriptors. In: *Computer Vision and Image Understanding (CVIU)*, vol. 145, 2016, no. C, pp. 95–110

[AGCO05] Aujol, Jean-François; Gilboa, Guy; Chan, Tony; Osher, Stanley: Structure-Texture Decomposition by a TV-Gabor Model. In: *Variational, Geometric, and Level Set Methods in Computer Vision.* Springer, 2005, pp. 85–96

[Ana89] Anandan, P: A Computational Framework and an Algorithm for the Measurement of Visual Motion. In: *International Journal of Computer Vision (IJCV)*, vol. 2, 1989, no. 3, pp. 283–310

[ASR12] Andriyenko, Anton; Schindler, Konrad; Roth, Stefan: Discrete-Continuous Optimization for Multi-Target Tracking. In: *IEEE*

Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2012, pp. 1926–1933

[ASW99] Alvarez, Luis; Sánchez, Javier; Weickert, Joachim: A Scale-Space Approach to Nonlocal Optical Flow Calculations. In: *International Conference on Scale-Space Theories in Computer Vision*, Springer, 1999, pp. 235–246

[BBPW04] Brox, Thomas; Bruhn, Andrés; Papenberg, Nils; Weickert, Joachim: High Accuracy Optical Flow Estimation Based on a Theory for Warping. In: *European Conference on Computer Vision (ECCV)*, Springer, 2004, pp. 25–36

[BBV08] Bouwmans, T; Baf, F E.; Vachon, B: Background Modeling using Mixture of Gaussians for Foreground Detection - A Survey. In: *Recent Patents on Computer Science*, vol. 1, 2008, no. 3, pp. 219–237

[BCM05] Buades, Antoni; Coll, Bartomeu; Morel, Jean-Michel: A Non-local Algorithm for Image Denoising. In: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, 2005, pp. 60–65

[BFF09] Berclaz, Jerome; Fleuret, Francois; Fua, Pascal: Multiple Object Tracking using Flow Linear Programming. In: *IEEE International Workshop on Performance Evaluation of Tracking and Surveillance (Winter-PETS)*, IEEE, 2009, pp. 1–8

[BGM04] Baker, Simon; Gross, Ralph; Matthews, Iain: Lucas-Kanade 20 Years On: A Unifying Framework. In: *International Journal of Computer Vision (IJCV)*, vol. 56, 2004, pp. 221–255

[BHLLR14] Bar Hillel, Aharon; Lerner, Ronen; Levi, Dan; Raz, Guy: Recent Progress in Road and Lane Detection: a Survey. In: *Machine Vision and Applications*, vol. 25, 2014, no. 3, pp. 727–745

[BKP10] Bredies, Kristian; Kunisch, Karl; Pock, Thomas: Total Generalized Variation. In: *SIAM Journal of Imaging Sciences*, vol. 3, 2010, no. 3, pp. 492–526

[Bla92] Black, Michael J.: *Robust Incremental Optical Flow*, Diss., Yale University, 1992

[BM11] Brox, Thomas; Malik, Jitendra: Large Displacement Optical Flow: Descriptor Matching in Variational Motion Estimation. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 33, 2011, no. 3, pp. 500–513

[BMM14] Böddeker, Christoph; Mohamed, Mahmoud A.; Mertsching, Bärbel: Detection and Tracking of Construction Workers and Equipment. In: *Bildverarbeitung in der Automation*, 2014, pp. 1–6

[BN96] Bhat, Dinkar N.; Nayar, Shree K.: Ordinal Measures for Visual Correspondence. In: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, 1996, pp. 351–357

[BR11] Benfold, Ben; Reid, Ian: Stable Multi-Target Tracking in Real-Time Surveillance Video. In: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, 2011, pp. 3457–3464

[Bri18] Brinkmann, Lukas: *Methoden zur Erkennung von bewegten Objekten bei gleichzeitiger Kamerabewegung*, University of Paderborn, GET Lab, Master thesis, 2018

[BRL+09] Breitenstein, Michael D.; Reichlin, Fabian; Leibe, Bastian; Koller-Meier, Esther; Gool, Luc J. V.: Robust Tracking-by-detection using a Detector Confidence Particle Filter. In: *IEEE International Conference on Computer Vision, (ICCV)*, IEEE, 2009, pp. 1515–1522

[BRL+11] Breitenstein, Michael D.; Reichlin, Fabian; Leibe, Bastian; Koller-Meier, Esther; Van Gool, Luc: Online Multiperson Tracking-by-Detection from a Single, Uncalibrated Camera. In: *IEEE Transaction on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 33, 2011, no. 9, pp. 1820–1833

[BSGF10] Barnes, Connelly; Shechtman, Eli; Goldman, Dan B.; Finkelstein, Adam: The Generalized PatchMatch Correspondence Algorithm. In: *European Conference on Computer Vision (ECCV)*, Springer, 2010, pp. 29–43

[BSL+11] Baker, Simon; Scharstein, Daniel; Lewis, J. P.; Roth, Stefan; Black, Michael J.; Szeliski, Richard: A Database and Evaluation Methodology for Optical Flow. In: *International Journal of Computer Vision (IJCV)*, vol. 92, 2011, no. 1, pp. 1–31

[BSL+al] Baker, Simon; Scharstein, Daniel; Lewis, J. P.; Roth, Stefan; Black, Michael J.; Szeliski, Richard: Middlebury benchmark, [On-line; accessed 01-April 2019] http://vision.middlebury.edu/flow/eval/

[BTS15] Bailer, Christian; Taetz, Bertram; Stricker, Didier: Flow fields: Dense Correspondence Fields for Highly Accurate Large Displacement Optical Flow Estimation. In: *IEEE International Conference on Computer Vision (ICCV)*, IEEE, 2015, pp. 4015–4023

[BTVG06] Bay, Herbert; Tuytelaars, Tinne; Van Gool, Luc: SURF: Speeded Up Robust Features. In: *European Conference on Computer Vision (ECCV)*, Springer, 2006, pp. 404–417

[BW05] Bruhn, Andrés; Weickert, Joachim: Towards Ultimate Motion Estimation: Combining Highest Accuracy with Real-Time Performance. In: *IEEE International Conference on Computer Vision (ICCV)*, IEEE, 2005, pp. 749–755

[BW13] Brox, Thomas; Weickert, Joachim: Special Session on Robust Optical Flow. In: *German Conference of Pattern Recognation*, Springer International Publishing, 2013

[BWF+03] Bruhn, Andrés; Weickert, Joachim; Feddern, Christian; Kohlberger, Timo; Schnörr, Christoph: Real-Time Optic Flow Computation with Variational Methods. In: *10th International Conference on Computer Analysis of Images and Patterns CAIP*, 2003, pp. 222–229

[BWml] Brox, Thomas; Weickert, Joachim: Special Session on Robust Optical Flow, [On-line; accessed 20-April 2019] https://resources.mpi-inf.mpg.de/conference/dagm/2013/SpecialSession.html.

[BWKS05] Bruhn, Andrés; Weickert, Joachim; Kohlberger, Timo; Schnörr, Christoph: Discontinuity-Preserving Computation of Variational Optic Flow in Real-Time. In: *International Conference on Scale-Space Theories in Computer Vision*, Springer, 2005, pp. 279–290

[BWS05] Bruhn, Andrés; Weickert, Joachim; Schnörr, Christoph: Lucas/Kanade Meets Horn/Schunck: Combining Local and Global Optic Flow Methods. In: *International Journal of Computer Vision (IJCV)*, vol. 61, 2005, no. 3, pp. 211–231

[BWSB12] Butler, Daniel J.; Wulff, Jonas; Stanley, Garrett B.; Black, Michael J.: A Naturalistic Open Source Movie for Optical Flow Evaluation. In: *European Conference Computer Vision (ECCV)*, Springer Lecture Notes in Computer Science, 2012, pp. 611–625

[BWSBts] Butler, Daniel J.; Wulff, Jonas; Stanley, Garrett B.; Black, Michael J.: MPI benchmark, [On-line; accessed 01-April 2019] http://sintel.is.tue.mpg.de/results

[BYJ14a] Bao, Linchao; Yang, Qingxiong; Jin, Hailin: Fast Edge-Preserving PatchMatch for Large Displacement Optical Flow. In: *IEEE*

Transactions on Image Processing (TIP), vol. 23, 2014, no. 12, pp. 4996–5006

[BYJ14b] Bao, Linchao; Yang, Qingxiong; Jin, Hailin: Fast Edge-Preserving PatchMatch for Large Displacement Optical Flow. In: *IEEE Transactions on Image Processing (TIP)*, vol. 23, 2014, no. 12, pp. 4996–5006

[BZDB13] Braux-Zin, Jim; Dupont, Romain; Bartoli, Adrien: A General Dense Image Matching Framework Combining Direct and Feature-based Costs. In: *IEEE International Conference on Computer Vision (ICCV)*, 2013, pp. 185–192

[CBM03] Cristani, M.; Bicego, M.; Murino, V.: Multi-level Background Initialization using Hidden Markov Models. In: *ACM SIGMM International Workshop on Video Surveillance (IWVS)*, ACM, 2003, pp. 11–20

[CCBK06] Criminisi, Antonio; Cross, Geoffrey; Blake, Andrew; Kolmogorov, Vladimir: Bilayer Segmentation of Live Video. In: *Computer Vision and Pattern Recognition (CVPR)*, 2006, pp. 53–60

[CD00] Cutler, R.; Davis, L. S.: Robust Real-time Periodic Motion Detection, Analysis, and Applications. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, 2000, no. 8, pp. 781–796

[CGPP03] Cucchiara, Rita; Grana, Costantino; Piccardi, Massimo; Prati, Andrea: Detecting Moving Objects, Ghosts, and Shadows in Video Streams. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 25, 2003, no. 10, pp. 1337–1342

[Cha04] Chambolle, Antonin: An Algorithm for Total Variation Minimization and Applications. In: *Journal of Mathematical Imaging and Vision*, vol. 20, 2004, no. 1-2, pp. 89–97

[CHHN98] Coombs, David; Herman, Martin; Hong, Tsai-Hong; Nashman, Marilyn: Real-Time Obstacle Avoidance Using Central Flow Divergence and Peripheral Flow. In: *IEEE Transactions on Robotics and Automation*, vol. 14, 1998, pp. 49 – 59

[CJL+13] Chen, Z.; Jin, H.; Lin, Z.; Cohen, S.; Wu, Y.: Large Displacement Optical Flow from Nearest Neighbor Fields. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, June 2013, pp. 2443–2450

[CLSF10] Calonder, Michael; Lepetit, Vincent; Strecha, Christoph; Fua, Pascal: BRIEF: Binary Robust Independent Elementary Features. In: *European Conference on Computer Vision ECCV*, Springer, 2010, pp. 778–792

[CSWS17] Cao, Zhe; Simon, Tomas; Wei, Shih-En; Sheikh, Yaser: Realtime multi-person 2d pose estimation using part affinity fields. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 7291–7299

[Dau19] Daube, Christian: *Semantic Motion Segmentation using Deep Learning*, University of Paderborn, GET Lab, Master thesis (ongoing, delvery date 20/5/2019), 2019

[DFI+15] Dosovitskiy, Alexey; Fischer, Philipp; Ilg, Eddy; Hausser, Philip; Hazirbas, Caner; Golkov, Vladimir; Smagt, Patrick van d.; Cremers, Daniel; Brox, Thomas: Flownet: Learning Optical Flow with Convolutional Networks. In: *IEEE International Conference on Computer Vision (ICCV)*, IEEE, 2015, pp. 2758–2766

[DHW13] Demetz, Oliver; Hafner, David; Weickert, Joachim: The Complete Rank Transform: A Tool for Accurate and Morphologically Invariant Matching of Structures. In: *British Machine Vision Conference (BMVC)*, BMVA Press, 2013, pp. 1–12

[DLH14] Dai, Yuchao; Li, Hongdong; He, Mingyi: A Simple Prior-Free Method for Non-rigid Structure-from-Motion Factorization. In: *International Journal of Computer Vision (IJCV)*, vol. 107, 2014, no. 2, pp. 101–122

[DN11] Drulea, Marius; Nedevschi, Sergiu: Total Variation Regularization of Local-Global Optical Flow. In: *IEEE International Conference on Intelligent Transportation Systems (ITSC)*, IEEE, 2011, pp. 318–323

[DN13] Drulea, Marius; Nedevschi, Sergiu: Motion Estimation Using the Correlation Transform. In: *IEEE Transactions on Image Processing (TIP)*, vol. 22, 2013, no. 8, pp. 3260–3270

[Dod16] Dodic, Branko: *Classification of Moving Objects Using Appearance and Optical Flow*, University of Paderborn, GET Lab, Master thesis, 2016

[DRSS12] Dey, Soumyabrata; Reilly, Vladimir; Saleemi, Imran; Shah, Mubarak: Detection of Independently Moving Objects in Non-Planar Scenes via Multi-Frame Monocular Epipolar Constraint. In: *European Conference on Computer Vision (ECCV)*, Springer, 2012, pp. 860–873

[DSV+14] Demetz, Oliver; Stoll, Michael; Volz, Sebastian; Weickert, Joachim; Bruhn, Andrés: Learning Brightness Transfer Functions for the Joint Recovery of Illumination Changes and Optical Flow. In: *European Conference on Computer Vision (ECCV)*, Springer, 2014, pp. 455–471

[DT05] Dalal, Navneet; Triggs, Bill: Histograms of Oriented Gradients for Human Detection. In: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, 2005, pp. 886–893

[DZ12] Dian Zhu, Ningjian Y. Huadong Sun S. Huadong Sun: A Real-time and Robust Approach for Short-term Multiple Objects Tracking.

In: *Computer Science and Information Processing (CSIP)*, 2012, pp. 453 – 456

[EG11] Enzweiler, Markus; Gavrila, Dariu M.: A Multilevel Mixture-of-Experts Framework for Pedestrian Classification. In: *IEEE Transactions on Image Processing (TIP)*, vol. 20, 2011, no. 10, pp. 2967–2979

[EHD00] Elgammal, Ahmed M.; Harwood, David; Davis, Larry S.: Non-parametric Model for Background Subtraction. In: *European Conference on Computer Vision (ECCV)*, Springer, 2000, pp. 751–767

[EKB98] Eveland, Christopher K.; Konolige, Kurt; Bolles, Robert C.: Background Modeling for Segmentation of Video-Rate Stereo Sequences. In: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, 1998, pp. 266–271

[Far03] Farnebäck, Gunnar: Two-Frame Motion Estimation Based on Polynomial Expansion. In: *13th Scandinavian Conference on Image Analysis*, Springer Lecture Notes in Computer Science, 2003, pp. 363–370

[FB81] Fischler, Martin A.; Bolles, Robert C.: Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. In: *Communications ACM*, vol. 24, 1981, no. 6, pp. 381–395

[FBK15] Fortun, Denis; Bouthemy, Patrick; Kervrann, Charles: Optical Flow Modeling and Computation: A survey. In: *Computer Vision and Image Understanding (CVIU)*, vol. 134, 2015, pp. 1 – 21

[FBK16] Fortun, Denis; Bouthemy, Patrick; Kervrann, Charles: Aggregation of Local Parametric Candidates with Exemplar-based Occlusion Handling for Optical Flow. In: *Computer Vision and Image Understanding (CVIU)*, vol. 145, 2016, pp. 81–94

[FE04] Froba, Bernhard; Ernst, Andreas: Face Detection with the Modified Census Transform. In: *IEEE International Conference on Automatic Face and Gesture Recognition*, IEEE, 2004, pp. 91–96

[FR97] Friedman, Nir; Russell, Stuart: Image Segmentation in Video Sequences: A Probabilistic Approach. In: *13th Conference on Uncertainty in Artificial Intelligence*, Morgan Kaufmann Publishers Inc., 1997, pp. 175–181

[FWH12] Fan, Bin; Wu, Fuchao; Hu, Zhanyi: Rotationally Invariant Descriptors Using Intensity Order Pooling. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 34, 2012, no. 10, pp. 2031–2045

[FZMB02] Foroosh, Hassan; Zerubia, Josiane B.; Member, Senior; Berthod, Marc: Extension of Phase Correlation to Subpixel Registration. In: *IEEE Transaction on Image Processing (TIP)*, vol. 11, 2002, no. 3, pp. 188–200

[GAS+19] Goldt, Sebastian; Advani, Madhu S.; Saxe, Andrew M.; Krzakala, Florent; Zdeborová, Lenka: Generalisation dynamics of online learning in over-parameterised neural networks. In: *arXiv preprint arXiv:1901.09085*, , 2019

[GKN+18] Gaspers, D.; Knorr, C.; Nickchen, T.; Nickchen, D.; Mertsching, B.; Mohamed, M. A.: Real-time Graph-Based 3D Reconstruction of Sparse Feature Environments for Mobile Robot Applications. In: *2018 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, 2018, pp. 1–6

[GLSU13] Geiger, Andreas; Lenz, Philip; Stiller, Christoph; Urtasun, Raquel: Vision Meets Robotics: The KITTI Dataset. In: *International Journal of Robotics Research (IJRR)*, vol. 32, 2013, no. 11, pp. 1231–1237

[GM17] Garrigues, Matthieu; Manzanera, Antoine: Fast Semi Dense Epipolar Flow Estimation. In: *IEEE Winter Conference on Applications of Computer Vision (WACV)*, IEEE, 2017, pp. 427–435

[GO09] Gilboa, Guy; Osher, Stanley: Nonlocal Operators with Applications to Image Processing. In: *Multiscale Modeling and Simulation*, vol. 7, 2009, no. 3, pp. 1005–1028

[Har97] Hartley, Richard I.: In Defense of the Eight-Point Algorithm. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 19, 1997, no. 6, pp. 580–593

[HCB11] Henriques, João F.; Caseiro, Rui; Batista, Jorge: Globally Optimal Solution to Multi-object Tracking with Merged Measurements. In: *IEEE International Conference on Computer Vision (ICCV)*, IEEE, 2011, pp. 2470–2477

[HG81] Horn, Berthold K. P.; G., Schunck B.: Determining Optical Flow. In: *Artifical Intelligence*, vol. 17, 1981, no. 1-3, pp. 185–203

[HK12] Hermann, Simon; Klette, Reinhard: Hierarchical Scan-line Dynamic Programming for Optical Flow using Semi-global Matching. In: *Asian Conference on Computer Vision (ACCV)*, Springer, 2012, pp. 556–567

[HKSLML13] Hyun Kim, Tae; Seok Lee, Hee; Mu Lee, Kyoung: Optical Flow via Locally Adaptive Fusion of Complementary Data Costs. In: *IEEE International Conference on Computer Vision (ICCV)*, IEEE, 2013, pp. 3344 – 3351

[HZ03] Hartley, Richard; Zisserman, Andrew: *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2003

[Jen08] Jens, Klappstein: *Optical-Flow Based Detection of Moving Objects in Traffic Scenes*, Diss., Heidelberg University, 2008

[JHD16] Jason, J Y.; Harley, Adam W.; Derpanis, Konstantinos G.: Back to Basics: Unsupervised Learning of Optical Flow Via Brightness Constancy and Motion Smoothness. In: *European Conference on Computer Vision (ECCV)*, Springer, 2016, pp. 3–10

[JKC10] Jabid, Taskeed; Kabir, Hasanul; Chae, Oksam: Local Directional Pattern (LDP) for Face Recognition. In: *International Conference on Consumer Electronics (ICCE)*, IEEE, 2010, pp. 329–330

[JP13] Je, Changsoo; Park, Hyung-Min: Optimized Hierarchical Block Matching for Fast and Accurate Image Registration. In: *Signal Processing: Image Communication*, vol. 28, 2013, no. 7, pp. 779–791

[JRD12] Jiang, Xiaoyan; Rodner, Erik; Denzler, Joachim: Multi-person Tracking-by-Detection Based on Calibrated Multi-camera Systems. In: *International Conference on Computer Vision and Graphics (ICCVG)*, Springer, 2012, pp. 743–751

[JS10] Jung, Boyoon; Sukhatme, Gaurav S.: Real-Time Motion Tracking from a Mobile Robot. In: *International Journal of Social Robotics*, vol. 2, 2010, no. 1, pp. 63–78

[KCHD04] Kim, Kyungnam; Chalidabhongse, Thanarat H.; Harwood, David; Davis, Larry: Background Modeling and Subtraction by Codebook Construction. In: *International Conference on Image Processing (ICIP)*, IEEE, 2004, pp. 3061–3064

[KCHD05] Kim, Kyungnam; Chalidabhongse, Thanarat H.; Harwood, David; Davis, Larry: Real-Time Foreground–Background Segmentation using Codebook Model. In: *Real-Time Imaging*, vol. 11, 2005, no. 3, pp. 172–185

[KHDL04] K., Kim; H., Chalidabhongse T.; D., Hanuood; L., Davis: Background Modeling and Subtraction by Codebook Construction. In: *International Conference on Image Processing (ICIP)*, IEEE, 2004, pp. 3061–3064

[KL12] Kitt, Bernd; Lategahn, Henning: Trinocular Optical Flow Estimation for Intelligent Vehicle Applications. In: *15th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, IEEE, 2012, pp. 300–306

[KMK05] Kim, Yeon-Ho; Martínez, Aleix M.; Kak, Avi C.: Robust Motion Estimation Under Varying Illumination. In: *Image and Vision Computing*, vol. 23, 2005, no. 4, pp. 365–375

[KMM12] Kalal, Zdenek; Mikolajczyk, Krystian; Matas, Jiri: Tracking-learning-detection. In: *IEEE transactions on pattern analysis and machine intelligence*, vol. 34, 2012, no. 7, pp. 1409–1422

[KRL10] Kitt, Bernd; Ranft, Benjamin; Lategahn, Henning: Block-Matching based Optical Flow Estimation with Reduced Search Space based on Geometric Constraints. In: *13th International IEEE Conference on Intelligent Transportation Systems*, IEEE, 2010, pp. 1104–1109

[KT15a] Kennedy, R.; Taylor, C. J.: Hierarchically-Constrained Optical Flow. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, 2015, pp. 3340–3348

[KT15b] Kennedy, Ryan; Taylor, Camillo J.: Optical Flow with Geometric Occlusion Estimation and Fusion of Multiple Frames. In: *International Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition*, Springer, 2015, pp. 364–377

[KTDVG16] Kroeger, Till; Timofte, Radu; Dai, Dengxin; Van Gool, Luc: Fast Optical Flow using Dense Inverse Search. In: *European Conference on Computer Vision (ECCV)*, Springer, 2016, pp. 471–488

[Lan15] Lange, Timo: *Semantic Annotation of View-based Object Representations Using On-line Search Engines*, University of Paderborn, GET Lab, Master thesis, 2015

[LBCS18] Livyatan, Harel; Berberian, Oded; Cohen, Barak; Stein, Gideon: *Dense structure from motion.* 2018. – US Patent 9,959,595

[LCS11] Leutenegger, Stefan; Chli, Margarita; Siegwart, Roland Y.: BRISK: Binary Robust Invariant Scalable Keypoints. In: *International Conference on Computer Vision ICCV*, IEEE, 2011, pp. 2548–2555

[LDFP93] Luong, Quangtuan; Deriche, Rachid; Faugeras, Olivier; Papadopoulo, Theo: On Determining the Fundamental Matrix: Analysis of Different Methods and Experimental Results / INRIA. 1993 (1894). – Technical Report

[LK81] Lucas, Bruce D.; Kanade, Takeo: An Iterative Image Registration Technique with an Application to Stereo Vision. In: *7th International Joint Conference on Artificial Intelligence (IJCAI)*, Morgan Kaufmann Publishers Inc., 1981, pp. 674–679

[LMB+15] Li, Yu; Min, Dongbo; Brown, Michael S.; Do, Minh N.; Lu, Jiangbo: SPM-BP: Sped-Up PatchMatch Belief Propagation for Continuous MRFs. In: *IEEE International Conference on Computer Vision, (ICCV)*, IEEE, 2015, pp. 4006–4014

[Low04] Lowe, David G.: Distinctive Image Features from Scale-Invariant Keypoints. In: *International Journal of Computer Vision (IJCV)*, vol. 60, 2004, no. 2, pp. 91–110

[Lu19] Lu, Ke: *Gesture-based Control System for a Robot Arm*, University of Paderborn, GET Lab, Master thesis, 2019

[LYMD13] Lu, Jiangbo; Yang, Hongsheng; Min, Dongbo; Do, Minh N.: Patch Match Filter: Efficient Edge-Aware Filtering Meets Randomized Search for Fast Correspondence Field Estimation. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, 2013, pp. 1854–1861

[LYT11] Liu, Ce; Yuen, Jenny; Torralba, Antonio: SIFT Flow: Dense Correspondence across Scenes and Its Applications. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, , 2011, no. 5, pp. 978–994

[LZL94] Li, Reoxiang; Zeng, Bing; Liou, M. L.: A New Three-step Search Algorithm for Block Motion Estimation. In: *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 4, 1994, no. 4, pp. 438–442

[LZLG12] Liau, Yung S.; Zhang, Qun; Li, Yanan; Ge, Shuzhi S.: Non-metric Navigation for Mobile Robot using Optical Flow. In: *International IEEE/RS Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2012, pp. 4953–4958

[LZS13] Leordeanu, Marius; Zanfir, Andrei; Sminchisescu, Cristian: Locally Affine Sparse-to-Dense Matching for Motion and Occlusion Estimation. In: *IEEE International Conference on Computer Vision (ICCV)*, IEEE, 2013, pp. 1721–1728

[MBM14] Mohamed, Mahmoud A.; Böddeker, Christoph; Mertsching, Bärbel: Real-Time Moving Objects Tracking for Mobile-Robots Using Motion Information. In: *IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, IEEE, 2014, pp. 1–6

[MBW07] Mileva, Yana; Bruhn, Andrés; Weickert, Joachim: Illumination-Robust Variational Optical Flow with Photometric Invariants. In: *DAGM-Symposium*, Springer Lecture Notes in Computer Science, 2007, pp. 152–162

[MCF10] Molnár, József; Chetverikov, Dmitry; Fazekas, Sándor: Illumination-Robust Variational Optical Flow using Cross-Correlation. In: *Computer Vision and Image Understanding (CVIU)*, vol. 114, 2010, no. 10, pp. 1104–1114

[MG15] Menze, Moritz; Geiger, Andreas: Object Scene Flow for Autonomous Vehicles. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, 2015, pp. 3061–3070

[MHG15a] Menze, Moritz; Heipke, Christian; Geiger, Andreas: Discrete Optimization for Optical Flow. In: *German Conference on Pattern Recognition (GCPR)*, Springer, 2015, pp. 16–28

[MHG15b] Menze, Moritz; Heipke, Christian; Geiger, Andreas: Joint 3D Estimation of Vehicles and Scene Flow. In: *ISPRS Workshop on Image Sequence Analysis (ISA)*, 2015

[MHG18] Menze, Moritz; Heipke, Christian; Geiger, Andreas: Object Scene Flow. In: *ISPRS Journal of Photogrammetry and Remote Sensing (JPRS)*, , 2018

[MHGow] Menze, Moritz; Heipke, Christian; Geiger, Andreas: KITTI benchmark, [On-line; accessed 01-April 2019] http://www.cvlibs.net/datasets/kitti/eval_stereo_flow.php?benchmark=flow

[Milml] Milan, Anton: PETS Data Sets and Ground Truth, [On-line; accessed 01-June 2018] http://www.milanton.de/data.html.

[ML11] Mei, Xue; Ling, Haibin: Robust Visual Tracking and Vehicle Classification via Sparse Representation. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 33, 2011, no. 11, pp. 2259–2272

[MM12a] Miksik, Ondrej; Mikolajczyk, Krystian: Evaluation of Local Detectors and Descriptors for Fast Feature Matching. In: *International Conference on Pattern Recognition (ICPR)*, IEEE, 2012, pp. 2681–2684

[MM12b] Mohamed, Mahmoud; Mertsching, Baerbel: TV-L1 Optical Flow Estimation with Image Details Recovering Based on Modified Census Transform. In: *Advances in Visual Computing*, vol. 7431, 2012, pp. 482–491

[MM12c] Mohamed, Mahmoud A.; Mertsching, Bärbel: Application Of Optical Flow In Automation. In: *Bildverarbeitung in der Automation (BVAu)*, 2012, pp. 1–6

[MMM15] Mohamed, Mahmoud A.; Mirabdollah, Hossein; Mertsching, Bärbel: Differential Optical Flow Estimation under Monocular Epipolar Line Constraint. In: *10th International Conference on Computer Vision Systems ICVS*, Springer, 2015, pp. 354–363

[MMM16] Mirabdollah, Hossein; Mohamed, Mahmoud; Mertsching, Bärbel: Distributed Averages of Gradients (DAG): A Fast Alternative for Histogram of Oriented Gradients. In: *20th International RoboCup Symposium*, Springer Lecture Notes on Artificial Intelligence, 2016, pp. 97–108

[MMM17] Mohamed, Mahmoud; Mirabdollah, Hossein; Mertsching, Bärbel: Monocular Epipolar Constraint for Optical Flow Estimation. In: *International Conference on Computer Vision Systems (ICVS)*, Springer, 2017, pp. 62–71

[MN94] Mattavelli, Marco; Nicoulin, Andre N.: Motion Estimation Relaxing the Constancy Brightness Constraint. In: *IEEE International Conference on Image Processing (ICIP)*, IEEE, 1994, pp. 770–774

[Moh14] Mohamed, Mahmoud A.: Human Action Recognition using Optical Flow and Hidden Markov Model / University of Paderborn, Department of Telecommunications. 2014. – Technical Report

[MRM+13] Mohamed, Mahmoud A.; Rashwan, Hatem A.; Mertsching, Bärbel; Garcia, Miguel A.; Puig, Domenec: On Improving the Robustness of Variational Optical Flow against Illumination Changes. In: *International ACM/IEEE Workshop on Analysis and Retrieval of Tracked Events and Motion in Imagery Streams (ARTEMIS)*, ACM, 2013, pp. 1–8

[MRM+14] Mohamed, Mahmoud A.; Rashwan, Hatem A.; Mertsching, Bärbel; Garcia, Miguel A.; Puig, Domenec: Illumination-Robust Optical Flow Using a Local Directional Pattern. In: *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 24, 2014, pp. 1499 – 1508

[MRR+11] Mueller, Thomas; Rabe, Clemens; Rannacher, Jens; Franke, Uwe; Mester, Rudolf: Illumination-Robust Dense Optical Flow Using Census Signatures. In: *DAGM-Symposium*, Springer Lecture Notes in Computer Science, 2011, pp. 236–245

[MSP16] Monzón, Nelson; Salgado, Agustín; Pérez, Javier S.: Regularization Strategies for Discontinuity-Preserving Optical Flow Methods. In: *IEEE Transaction in Image Processing (TIP)*, vol. 25, 2016, no. 4, pp. 1580–1591

[MSS+17] Mehta, Dushyant; Sridhar, Srinath; Sotnychenko, Oleksandr; Rhodin, Helge; Shafiei, Mohammad; Seidel, Hans-Peter; Xu, Weipeng; Casas, Dan; Theobalt, Christian: Vnect: Real-time 3d human pose estimation with a single rgb camera. In: *ACM Transactions on Graphics (TOG)*, vol. 36, 2017, no. 4, pp. 44

[MTM18] Mohamed, M. A.; Tünnermann, J.; Mertsching, B.: Seeing Signs of Danger: Attention-Accelerated Hazmat Label Detection. In: *2018 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, 2018, pp. 1–6

[Nakml] Nakahama, K.: The Aperture Problem in Apparent Motion, [Online; accessed 01-June-2018] http://pages.slc.edu/ebj/sightmind/motion/Nakayama/aperture-problem.html

[Ngu18] Nguyen, Thuy: *Moving Object Detection for a Non-stationary Camera*, University of Paderborn, GET Lab, Master thesis, 2018

[Nis05] Nistér, David: Preemptive RANSAC for Live Structure and Motion Estimation. In: *Machine Vision and Applications*, vol. 16, 2005, no. 5, pp. 321–329

[NM02] Nie, Yao; Ma, Kai-Kuang: Adaptive Rood Pattern Search for Fast Block-Matching Motion Estimation. In: *IEEE Transactions on Image Processing (TIP)*, vol. 11, 2002, no. 12, pp. 1442–1449

[Oli01] Oliensis, John: A Critique of Structure-from-Motion Algorithms. In: *Computer Vision and Image Understanding (CVIU)*, vol. 84, 2001, no. 3, pp. 407–408

[OMB14] Ochs, P.; Malik, J.; Brox, T.: Segmentation of Moving Objects by Long Term Video Analysis. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, 2014, no. 6, pp. 1187 – 1200

[PUZ+07] Pock, Thomas; Urschler, Martin; Zach, Christopher; Beichel, Reinhard; Bischof, Horst: A Duality Based Algorithm for TV-L1 Optical-Flow Image Registration. In: *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, Springer Lecture Notes in Computer Science, 2007, pp. 511–518

[Rai17] Raies, Ali: *Optical Flow Estimation Based on Depth Information and Scene Analysis*, University of Paderborn, GET Lab, Master thesis, 2017

[RBP14a] Ranftl, René; Bredies, Kristian; Pock, Thomas: Non-local Total Generalized Variation for Optical Flow Estimation. In: *European Conference on Computer Vision (ECCV)*, Springer, 2014, pp. 439–454

[RBP14b] Ranftl, René; Bredies, Kristian; Pock, Thomas: Non-local Total Generalized Variation for Optical Flow Estimation. In: *13th European Conference on Computer Vision ECCV*, Springer Lecture Notes in Computer Science, 2014, pp. 439–454

[RCH03] Ren, Ying; Chua, Chin-Seng; Ho, Yeong-Khing: Statistical Background Modeling for Non-stationary Camera. In: *Pattern Recognition Letter.*, vol. 24, 2003, no. 1-3, pp. 183–196

[RGP13] Rashwan, Hatem A.; García, Miguel A.; Puig, Domenec: Variational Optical Flow Estimation Based on Stick Tensor Voting. In: *IEEE Transactions on Image Processing (TIP)*, vol. 22, 2013, no. 7, pp. 2589–2599

[RGPB12] Ranftl, R.; Gehrig, S.; Pock, T.; Bischof, H.: Pushing the Limits of Stereo Using Variational Stereo Estimation. In: *Intelligent Vehicles Symposium (IV)*, IEEE, 2012, pp. 401–407

[RLF12] Rubinstein, Michael; Liu, Ce; Freeman, William T.: Towards Longer Long-Range Motion Trajectories. In: *British Machine Vision Conference*, BMVA Press, 2012, pp. 53.1–53.11

[RMG+13] Rashwan, Hatem A.; Mohamed, Mahmoud A.; García, Miguel A.; Mertsching, Bärbel; Puig, Domenec: Illumination Robust Optical Flow Model Based on Histogram of Oriented Gradients. In: *35th German Conference on Pattern Recognition (GCPR)*, Springer, 2013, pp. 354–363

[RMWF10] Rabe, Clemens; Müller, Thomas; Wedel, Andreas; Franke, Uwe: Dense, Robust, and Accurate Motion Field Estimation from Stereo Image Sequences in Real-Time. In: *European Conference on Computer Vision (ECCV)*, Springer Lecture Notes in Computer Science, 2010, pp. 582–595

[ROF92] Rudin, Leonid I.; Osher, Stanley; Fatemi, Emad: Nonlinear Total Variation Based Noise Removal Algorithms. In: *Physica D: Nonlinear Phenomena*, vol. 60, 1992, pp. 259–268

[Rol16] Roling, David: *Text Mining Methods for Semantic Annotation of View-based Object Representations*, University of Paderborn, GET Lab, Master thesis, 2016

[RPG11] Rashwan, Hatem A.; Puig, Domenec; Garcia, Angel: On Improving the Robustness of Differential Optical Flow. In: *International Conference on Computer Vision Workshop (ICCV Workshop)*, IEEE, 2011, pp. 876–881

[RPG12] Rashwan, Hatem A.; Puig, Domenec; García, Miguel A.: Improving the Robustness of Variational Optical Flow through Tensor Voting. In: *Computer Vision and Image Understanding (CVIU)*, vol. 116, 2012, no. 9, pp. 953–966

[RRCC13] Ramirez Rivera, A; Castillo, Rojas; Chae, Oksam: Local Directional Number Pattern for Face Analysis: Face and Expression Recognition. In: *IEEE Transaction of Image Processing*, vol. 22, 2013, pp. 1740–1752

[RRKB11] Rublee, Ethan; Rabaud, Vincent; Konolige, Kurt; Bradski, Gary: ORB: An Efficient Alternative to SIFT or SURF. In: *International Conference on Computer Vision ICCV*, IEEE, 2011, pp. 2564–2571

[RSPMB16] Rashwan, Hatem A.; Solanas, Agusti; Puig, Domènec; Martínez-Ballesté, Antoni: Understanding Trust in Privacy-Aware Video Surveillance Systems. In: *International Journal of Information Security*, vol. 15, 2016, no. 3, pp. 225–234

[RVCK16] Ranftl, Rene; Vineet, Vibhav; Chen, Qifeng; Koltun, Vladlen: Dense Monocular Depth Estimation in Complex Dynamic Scenes. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 4058–4066

[RWHS15a] Revaud, Jérôme; Weinzaepfel, Philippe; Harchaoui, Zaïd; Schmid, Cordelia: Deep Convolutional Matching. In: *Computer Vision and Pattern Recognition*, vol. abs/1506.07656, 2015, pp. 1164–1172

[RWHS15b] Revaud, Jerome; Weinzaepfel, Philippe; Harchaoui, Zaid; Schmid, Cordelia: EpicFlow: Edge-preserving Interpolation of Correspondences for Optical Flow. In: *Computer Vision and Pattern Recognition (CVPR)*, IEEE, 2015, pp. 1164–1172

[SBB+06] Stiefelhagen, Rainer; Bernardin, Keni; Bowers, Rachel; Garofolo, John S.; Mostefa, Djamel; Soundararajan, Padmanabhan: The CLEAR 2006 Evaluation. In: *International Evaluation Workshop on Classification of Events, Activities and Relationships*, Springer, 2006, pp. 1–44

[SBK10] Sundaram, Narayanan; Brox, Thomas; Keutzer, Kurt: Dense Point Trajectories by GPU-Accelerated Large Displacement Optical Flow. In: *European Conference on Computer Vision (ECCV)*, Springer Lecture Notes in Computer Science, 2010, pp. 438–451

[SCK04] Sen-Ching, S C.; Kamath, Chandrika: Robust Techniques for Background Subtraction in Urban Traffic Video. In: *Visual Communications and Image Processing*, International Society for Optics and Photonics, 2004, pp. 881–893

[SGG09] Stalder, Severin; Grabner, Helmut; Gool, Luc V.: Beyond Semi-Supervised Tracking: Tracking Should Be as Simple as Detection, but not Simpler than Recognition. In: *12th IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*, IEEE, 2009, pp. 1409 – 1416

[SGS16] Senst, Tobias; Geistert, Jonas; Sikora, Thomas: Robust Local Optical Flow: Long-range Motions and Varying Illuminations. In: *IEEE International Conference on Image Processing (ICIP)*, IEEE, 2016, pp. 4478–4482

[Sha12] Shafik, Mohamed Salah El-Neshawy: *3D Motion Analysis for Mobile Robots*, Diss., Paderborn University, 2012

[SKS+12] Sellent, Anita; Kondermann, Daniel; Simon, Stephan; Baker, Simon; Dedeoglu, Goksel; Erdler, Oliver; Parsonage, Phil; Unger, Christoph; Niehsen, Wolfgang: Optical Flow Estimation Versus Motion Estimation / Universitätsbibliothek der Universität Heidelberg. 2012. – Technical Report

[SLSLMB14] Sevilla-Lara, L.; Sun, D.; Learned-Miller, E.; Black, M.: Optical Flow Estimation with Channel Constancy. In: *European Conference on Computer Vision (ECCV)*, Springer, 2014, pp. 423–438

[SM10] SanMiguel, Juan C.; Martínez, José M: On the Evaluation of Background Subtraction Algorithms without Ground-truth. In: *IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, IEEE, 2010, pp. 180–187

[SOCM01] Stone, Harold S.; Orchard, Michael T.; Chang, Ee-Chien; Martucci, Stephen A.: A Fast Direct Fourier-based Algorithm for Subpixel Registration of Images. In: *IEEE Transaction on Geoscience and Remote Sensing*, vol. 39, 2001, no. 10, pp. 2235–2243

[SPLH17] Sahoo, Doyen; Pham, Quang; Lu, Jing; Hoi, Steven C.: Online deep learning: Learning deep neural networks on the fly. In: *arXiv preprint arXiv:1711.03705*, , 2017

[SRB10] Sun, Deqing; Roth, Stefan; Black, Michael J.: Secrets of Optical Flow Estimation and Their Principles. In: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, 2010, pp. 2432–2439

[SRB14a] Sun, Deqing; Roth, Stefan; Black, Michael J.: A Quantitative Analysis of Current Practices in Optical Flow Estimation and the Principles Behind Them. In: *International Journal of Computer Vision (IJCV)*, vol. 106, 2014, no. 2, pp. 115–137

[SRB14b] Sun, Deqing; Roth, Stefan; Black, MichaelJ.: A Quantitative Analysis of Current Practices in Optical Flow Estimation and the Principles Behind Them. In: *International Journal of Computer Vision (IJCV)*, vol. 106, 2014, no. 2, pp. 115–137

[SSB10] Sun, Deqing; Sudderth, Erik B.; Black, Michael J.: Layered Image Motion with Explicit Occlusions, Temporal Consistency, and

Depth Ordering. In: *Advances in Neural Information Processing Systems*, vol. 23, 2010, pp. 2226–2234

[SSB12] Sun, Deqing; Sudderth, Erik B.; Black, Michael J.: Layered Segmentation and Optical Flow Estimation Over Time. In: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, 2012, pp. 1768–1775

[SSB13] Sun, Deqing; Sudderth., Erik B.; Black, Michael J.: A Fully-Connected Layered Model of Foreground and Background Flow. In: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, 2013, pp. 2451–2458

[ST94] Shi., Jianbo; Tomasi, Carlo: Good Features to Track. In: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, 1994, pp. 593–600

[ST06] Sand, Peter; Teller, Seth J.: Particle Video: Long-Range Motion Estimation using Point Trajectories. In: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, 2006, pp. 2195–2202

[ST08] Sand, Peter; Teller, Seth J.: Particle Video: Long-Range Motion Estimation Using Point Trajectories. In: *International Journal of Computer Vision (IJCV)*, vol. 80, 2008, no. 1, pp. 72–91

[Ste04a] Stein, Fridtjof: Efficient Computation of Optical Flow Using the Census Transform. In: *Pattern Recognition: 26th DAGM-Symposium*, Springer Lecture Notes in Computer Science, 2004, pp. 79–86

[Ste04b] Stein, Fridtjof: Efficient Computation of Optical Flow using the Census Transform. In: *Pattern Recognition: DAGM-Symposium*, Springer Lecture Notes in Computer Science, 2004, pp. 79–86

[SW18] Sun, Zefeng; Wang, Hanli: Deeper Spatial Pyramid Network with Refined Up-Sampling for Optical Flow Estimation. In: Hong,

Richang (Hrsg.); Cheng, Wen-Huang (Hrsg.); Yamasaki, Toshihiko (Hrsg.); Wang, Meng (Hrsg.); Ngo, Chong-Wah (Hrsg.): *Advances in Multimedia Information Processing – PCM 2018*, Springer International Publishing, 2018, pp. 492–501

[SWH+18] Sherwood, Christopher R.; Warrick, Jonathan A.; Hill, Andrew D.; Ritchie, Andrew C.; Andrews, Brian D.; Plant, Nathaniel G.: Rapid, Remote Assessment of Hurricane Matthew Impacts Using Four-Dimensional Structure-from-Motion Photogrammetry. In: *Journal of Coastal Research*, , 2018

[TM04] Talukder, Ashit; Matthies, Larry: Real-time Detection of Moving Objects from Moving Vehicle using Dense Stereo and Optical Flow. In: *International IEEE/RS Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2004, pp. 3718 – 3725

[TMJP16] Trupti M., Pandit; Jadhav, P. M.; Phadke, A. C.: Suspicious Object Detection in Surveillance Videos for Security Applications. In: *International Conference on Inventive Computation Technologies (ICICT)*, 2016, pp. 1–5

[Tom92] Tomasi, Carlo: Shape and Motion from Image Streams under Orthography: a Factorization Method. In: *International Journal of Computer Vision (IJCV)*, vol. 9, 1992, no. 2, pp. 137–154

[TSS17] Taniai, Tatsunori; Sinha, Sudipta; Sato, Yoichi: Fast Multiframe Stereo Scene Flow with Motion Segmentation. In: *IEEE Computer Vision and Pattern Recognition (CVPR)*, IEEE, 2017, pp. 6891–6900

[TZD16] Tron, Roberto; Zhou, Xiaowei; Daniilidis, Kostas: A Survey on Rotation Optimization in Structure From Motion. In: *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPR Workshops)*, 2016, pp. 1032–1040

[USAR08] Unaldi, Numan; Sankaran, Praveen; Asari, Vijayan K.; Rahman, Zia ur: Image Enhancement for Improving Face Detection

under Non-uniform Lighting Conditions. In: *International Conference on Image Processing (ICIP)*, IEEE, 2008, pp. 1332–1335

[VBW08] Valgaerts, Levi; Bruhn, Andrés; Weickert, Joachim: A Variational Model for the Joint Recovery of the Fundamental Matrix and the Optical Flow. In: *Pattern Recognition: 30th DAGM-Symposium*, Springer, 2008, pp. 314–324

[Vog15] Vogt, Robin: *Efficient Implementation and Evaluation of Variational Optical Flow Based on Texture Constraints*, University of Paderborn, GET Lab, Master thesis, 2015

[VRS13] Vogel, Christoph; Roth, Stefan; Schindler, Konrad: An Evaluation of Data Costs for Optical Flow. In: *35th German Conference on Pattern Recognition (GCPR)*. Springer Lecture Notes in Computer Science, 2013, pp. 343–353

[VSR13] Vogel, Christoph; Schindler, Konrad; Roth, Stefan: Piecewise Rigid Scene Flow. In: *IEEE International Conference on Computer Vision (ICCV)*, IEEE, 2013, pp. 1377–1384

[WB14] Wulff, J.; Black, M. J.: Efficient Sparse-to-Dense Optical Flow Estimation using a Learned Basis and Layers. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, 2014, pp. 120–130

[Wed09] Wedel, Andreas: *3D Motion Analysis via Energy Minimization*, Diss., Bonn University, 2009

[Wer12] Werlberger, Manuel: *Convex Approaches for High Performance Video Processing*, Diss., Institute for Computer Graphics and Vision Graz University of Technology, 2012

[WFW11] Wang, Zhenhua; Fan, Bin; Wu, Fuchao: Local Intensity Order Pattern for Feature Description. In: *International Conference on Computer Vision ICCV*, IEEE, 2011, pp. 603–610

[WHS13] Weickert, Joachim; Hein, Matthias; Schiele, Bernt: German Conference of Pattern Recognition. In: *Pattern Recognition*, Springer, 2013, pp. 860–873

[Wikki] Wikipedia: The Aperture Problem in Apparent Motion, [On-line; accessed 01-June 2018] http://en.wikipedia.org/wiki/

[WPB10] Werlberger, Manuel; Pock, Thomas; Bischof, Horst: Motion Estimation with Non-Local Total Variation Regularization. In: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, 2010, pp. 2464–2471

[WRHS13a] Weinzaepfel, Philippe; Revaud, Jérôme; Harchaoui, Zaid; Schmid, Cordelia: DeepFlow: Large Displacement Optical Flow with Deep Matching. In: *IEEE International Conference on Computer Vision (ICCV)*, IEEE, 2013, pp. 1385–1392

[WRHS13b] Weinzaepfel, Philippe; Revaud, Jerome; Harchaoui, Zaid; Schmid, Cordelia: DeepFlow: Large Displacement Optical Flow with Deep Matching. In: *IEEE Intentional Conference on Computer Vision (ICCV)*, IEEE, 2013, pp. 1385–1392

[WTK87] Witkin, Andrew; Terzopoulos, Demetri; Kass, Michael: Signal Matching Through Scale Space. In: *International Journal of Computer Vision (IJCV)*, vol. 1, 1987, no. 2, pp. 133–144

[WWR+13a] Wojek, Christian; Walk, Stefan; Roth, Stefan; Schindler, Konrad; Schiele, Bernt: Monocular Visual Scene Understanding: Understanding Multi-Object Traffic Scenes. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 35, 2013, no. 4, pp. 882–897

[WWR+13b] Wojek, Christian; Walk, Stefan; Roth, Stefan; Schindler, Konrad; Schiele, Bernt: Monocular Visual Scene Understanding: Understanding Multi-Object Traffic Scenes. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 35, 2013, no. 4, pp. 882–897

[WXZ+19] Wang, Chen; Xu, Danfei; Zhu, Yuke; Martín-Martín, Roberto; Lu, Cewu; Fei-Fei, Li; Savarese, Silvio: DenseFusion: 6D Object Pose Estimation by Iterative Dense Fusion. In: *arXiv preprint arXiv:1901.04780*, , 2019

[XJM12] Xu, Li; Jia, Jiaya; Matsushita, Yasuyuki: Motion Detail Preserving Optical Flow Estimation. In: *IEEE Transaction on Pattern Analysis and Machine Intelligent (PAMI)*, vol. 34, 2012, no. 9, pp. 1744–1757

[XT13] Xu, R.; Taubman, D.: Robust Dense Block-based Motion Estimation using a 2-bit Transform on a Laplacian Pyramid. In: *IEEE International Conference on Image Processing (ICIP)*, IEEE, 2013, pp. 1938–1942

[XZH19] Xiang, Jun; Zhang, Guoshuai; Hou, Jianhua: Online Multi-Object Tracking Based on Feature Representation and Bayesian Filtering within a Deep Learning Architecture. In: *IEEE Access*, , 2019

[YCREB19] Yang, Tao; Cappelle, Cindy; Ruichek, Yassine; El Bagdouri, Mohammed: Online multi-object tracking combining optical flow and compressive tracking in Markov decision process. In: *Journal of Visual Communication and Image Representation*, vol. 58, 2019, pp. 178–186

[YJS06] Yilmaz, Alper; Javed, Omar; Shah, Mubarak: Object Tracking: A Survey. In: *Acm Computing Surveys (CSUR)*, vol. 38, 2006, no. 4, pp. 1–45

[YL15a] Yang, J.; Li, H.: Dense, Accurate Optical Flow Estimation with Piecewise Parametric Model. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 1019–1027

[YL15b] Yang, Jiaolong; Li, Hongdong: Dense, Accurate Optical Flow Estimation With Piecewise Parametric Model. In: *IEEE Conference*

on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 1019–1027

[YMU13a] Yamaguchi, Koichiro; McAllester, David; Urtasun, Raquel: Robust Monocular Epipolar Flow Estimation. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, 2013, pp. 1862–1869

[YMU13b] Yamaguchi, Koichiro; McAllester, David A.; Urtasun, Raquel: Robust Monocular Epipolar Flow Estimation. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, 2013, pp. 1862–1869

[YP06] Yan, Jingyu; Pollefeys, Marc: A General Framework for Motion Segmentation: Independent, Articulated, Rigid, Non-rigid, Degenerate and Non-degenerate. In: *9th European Conference on Computer Vision (ECCV)*, Springer Lecture Notes in Computer Science, 2006, pp. 94–106

[YS18] Yang, Yanchao; Soatto, Stefano: Conditional Prior Networks for Optical Flow. In: *The European Conference on Computer Vision (ECCV)*, 2018

[ZBW+09] Zimmer, Henning; Bruhn, Andrés; Weickert, Joachim; Valgaerts, Levi; Salgado, Agustín; Rosenhahn, Bodo; Seidel, Hans-Peter: Complementary Optic Flow. In: *International Conference on Energy Minimization Methods in Computer Vision and Pattern Recognition (EMMCVPR)*, Springer, 2009, pp. 207–220

[ZBW11] Zimmer, Henning; Bruhn, Andrés; Weickert, Joachim: Optic Flow in Harmony. In: *International Journal of Computer Vision (IJCV)*, vol. 93, 2011, no. 3, pp. 368–388

[ZLS08] Zappella, Luca; Lladó, Xavier; Salvi, Joaquim: Motion Segmentation: A Review. In: *Conference on Artificial Intelligence Research and Development*, IOS Press, 2008, pp. 398–407

[ZLS17] Zhu, En; Li, Yuanwei; Shi, Yanling: Fast Optical Flow Estimation Without Parallel Architectures. In: *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 27, 2017, no. 11, pp. 2322–2332

[ZM00] Zhu, Shan; Ma, Kai-Kuang: A New Diamond Search Algorithm for Fast Block-matching Motion Estimation. In: *IEEE Transactions on Image Processing (TIP)*, vol. 9, 2000, no. 2, pp. 287–290

[ZSI19] Zakharov, Sergey; Shugurov, Ivan; Ilic, Slobodan: DPOD: Dense 6D Pose Object Detector in RGB images. In: *arXiv preprint arXiv:1902.11020*, , 2019

[ZW94] Zabih, Ramin; Woodfill, John: Non-Parametric Local Transforms for Computing Visual Correspondence. In: *European Conference Computer Vision (ECCV)*, Springer Lecture Notes in Computer Science, 1994, pp. 151–158

List of Publications

The work presented in this thesis has been published in the following blind peer-reviewed international conference proceedings and journals:

[1] **M. Mohamed**, and B. Mertsching. Dense Differential Optical Flow Estimation through Monocular Epipolar Line Constraint, Journal CVIU, under revision.

[2] **M. Mohamed**, J. Tünnermann, and B. Mertsching. Seeing Signs of Danger: Attention-Accelerated Hazmat Label Detection. In: 2018 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR), **Best paper finalist**, Philadelphia, PA, USA (IEEE), 2018.

[3] D. Gaspers, C. Knorr, T. Nickchen, D. Nickchen, B. Mertsching, and **M. Mohamed**. Real-time Graph-Based 3D Reconstruction of Sparse Feature Environments for Mobile Robot Applications. In: 2018 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR), Philadelphia, PA, USA (IEEE), 2018.

[4] **M. Mohamed**, and B. Mertsching. Monocular Epipolar Constraint for Optical Flow Estimation. In: International Conference on Computer Vision Systems (ICVS 2017), Shenzhen, China, 2017.

[5] L. Jianxun, **M. Mohamed**, B. Mertsching, and H. Yuan. Dense Optical Flow Estimation from RGB-D images. In: the 7th International Conference on Instrumentation, Measurement, Computer, Communication and Control (IMCCC), Changchun, China, 2017.

[6] M. H. Mirabdollah, **M. Mohamed**, and B. Mertsching. Distributed Averages of Gradients (DAG): A Fast Alternative for Histogram of Oriented Gradients, in *20th International RoboCup Symposium, Springer Lecture Notes on Artificial Intelligence*, **Best paper finalist**. vol. 9776. 2016.

[7] **M. Mohamed**, H. Mirabdollah, and B. Mertsching. Differential Optical Flow Estimation under Monocular Epipolar Line Constraint, in: *10th International Conference on Computer Vision Systems ICVS*, Lecture Notes in Computer Vision, vol. 9163, pp. 354-363,2015.

[8] **M. Mohamed**, H. Rashwan, B. Mertsching, M. Garcia, and D. Puig. Illumination-Robust Optical Flow Using Local Directional Pattern, in: Journal IEEE Transactions on Circuits and Systems for Video Technology, vol. 24, no. 9, pp. 1499-1508. ISSN 1051-8215, 2015.

[9] **M. Mohamed**, C. Böddeker, and B. Mertsching. Real-Time Moving Objects Tracking for Mobile-Robots Using Motion Information, in: IEEE International Symposium on Safety, Security, and Rescue Robotics, 2014.

[10] C. Böddeker, **M. Mohamed**, and B. Mertsching. Detection and Tracking of Construction Workers and Equipment, in: Bildverarbeitung in der Automation, 2014.

[11] **M. Mohamed**, H. Rashwan, B. Mertsching, M. Garcia, and D. Puig. On Improving the Robustness of Variational Optical Flow against Illumination Changes, in: 4th ACM/IEEE ARTEMIS 2013 International Workshop on Analysis and Retrieval of Tracked Events and Motion in Imagery Streams, pp. 1-8. ISBN 978-1-4503-2393-2, 2013.

[12] H. Rashwan, **M. Mohamed**, M. Garcia, B. Mertsching, and D. Puig. Illumination Robust Optical Flow Model Based on Histogram of Oriented Gradients. in: German Conference on Pattern Recognition, Springer Berlin Heidelberg, Lecture Notes in Computer Science, pp. 354-363. ISBN 978-3-642-40601-0, 2013

[13] **M. Mohamed** and B. Mertsching. TV-L1 Optical Flow Estimation With Image Details Recovering Based on Modified Census Transform, in: Advances in Visual Computing, vol. 7431, 2012, pp. 482-491.

[14] **M. Mohamed** and B. Mertsching. Application of Optical Flow In Automation, in: Bildverarbeitung in der Automation, 2012.

List of Notations

Notation	Explanation
ϵ	constant equal to 0.001 in this thesis, threshold
$\mathcal{E}(u, v)$	Energy Function
$\mathcal{ER}$	Edge Response
E	Essential matrix
$f(x, y)$	$I(x, y) * G(x, y)$ a convolution with a Gaussian kernel
F	Fundamental matrix
H	Homography matrix
$(i^{'}, j^{'}) \in N_{i,j}$	A pixel in region $\cup\{i, j\}$
$I, I(x, y), I(x, y, t)$	Image intensity function
I_t	Temporal gradient
I_x	Spacial gradient in x direction
I_y	Spacial gradient in y direction
J_ρ	A Gaussian $K_\rho(x, y)$ with a stander deviation ρ
$J_\rho(\nabla_3 f)$	$K_\rho * (\nabla_3 f \nabla_3 f^T)$
K	Camera calibration matrix
K_σ	Gaussian Filter with standard deviation σ
K_ρ	Gaussian Filter with standard deviation ρ
M	Camera projection matrix
$N_{i,j}$	$N \times N$ local window
$\mathcal{O}(p)$	Occlusion state of point p
$p(x, y)$	pixel at position (x, y)
$p^{'} = (x + u, y + v)$	correspondence pixel coordinates
$\mathcal{P}$	Parzen-window
$\mathbb{R}^2$	2D Real number set
$\mathbf{S}(x, y)$	Multi-channels image descriptor at point x and y

Notation	Explanation
$T_1, T_2, \bar{T}_1, \bar{T}_1$	Image templates
u, v	Optical flow components in x and y directions
$\hat{v}, \hat{v}$	Dual Optical flow components in x and y directions
$\varnothing u$	Motion field component in x direction
$\varnothing v$	Motion field component in y direction
$\mathbf{w} = [u, v]^T$	Optical flow vector
$\varnothing\mathbf{w} = [\varnothing u, \varnothing v]^T$	Motion field vector
x, y, t	Spatial and time image coordinates
X, Y, Z	3D world coordinate
γ	Regularization parameters for the smoothness term.
$\lambda, \lambda_1, \lambda_2$	Regularization parameters for data-terms.
θ	A threshold for the dual variables.
$\varpi_{p,\acute{p}}$	Weighting function between point p and point $\acute{p}$.
σ	Standard deviation
τ	Time step
χ	Unsigned integer function
ψ	A convex function $\psi(x^2) = \sqrt{x^2 + \varepsilon^2}$
$\nabla_2 f$	$(\partial f/\partial_x, \partial f/\partial_y)^T$
$\nabla_3 f$	$(\partial f/\partial_x, \partial f/\partial_y, \partial f/\partial_t)^T$
Ω	Image domain
α	Wighting value

List of Abbreviations

Abbreviation	Explanation
2D	Two Dimension
3D	Three Dimension
AAE	Average Angular Error
AE	Average error
AEE	Average Endpoint Error
AEE_{out}	Percentage of outliers
BCA	Brightness Constancy Assumption
BCC	Brightness Constancy Constraint
BS	Background Subtraction
CC	Codebook Construction
CLG	Combined Local Global
CT	Census Transform
CPU	Central Processing Unit
CSAD	Centralized Sum of Absolute Differences
DAG	Distributed Average Gradient
EE	Endpoint Error
FPGA	Field Programmable Gate Array
GCA	Gradient Constancy Assumption
GPU	Graphical Processing Unit
GT	Ground Truth
KF	Kalman Filter
HOT	High order terms
HOG	Histogram of Oriented Gradient
HMM	Hidden Markov Model

Abbreviation	Explanation
HS-OF	Horn/Shrunk Optical Flow
HSV	Hue, Saturation, Value
IE	Interpolation Error
KDE	Kernel Density Estimator
LBP	Local Binary Pattern
L_1	First order normalization
L_2	Second order normalization
LDNP	Local Directional Number Pattern
LDP	Local Directional Pattern
LK-OF	Lucas/Kanade Optical Flow
MAD	Minimum Absolute Difference
MCT	Modified Census Transform
MLDP	Modified Local Directional Pattern
MODA	Multi-Object Detection Accuracy
MoG	Mixture of Gaussian
MOTA	Multi-Object Tracking Accuracy
MOTP	Multiple Object Tracking Precision
MSE	Mean Squared Error
MRF	Markov Random Field
NE	Normalized Interpolation Error
NCC	Normalized Cross Correlation
NDAG	Normalized Distributed Average Gradient
PDF	Probability Density Function
PDE	Partial Differential Equation
RANSAC	Random Sample Consensus
ROF	Rudin-Osher-Fatemi
RGB	Red-Blue-Green
SAD	Sum Absolute Difference
SURF	Speeded Up Robust Features
SIFT	Scale Invariant Feature Transform

Abbreviation	Explanation
SVD	Singular Value Decomposition
SNR	Signal-to-noise ratio
Sqrt	Square root
SURF	Speeded Up Robust Features
TCA	Texture Constancy Assumption
TGV	Total Generalized Variation
TV	Total variation

List of Tables

List of Figures