Master Thesis

Czech
Technical
University
in Prague

F3

Faculty of Electrical Engineering
Department of Cybernetics

Object Scene Flow in Video Sequences

Bc. Michal Neoral

Supervisor: Mgr. Jan Šochman, Ph.D.
Field of study: Open Informatics
Subfield: Computer Vision and Image Processing
May 2017

Czech Technical University in Prague
Faculty of Electrical Engineering

Department of Cybernetics

DIPLOMA THESIS ASSIGNMENT

Student: Bc. Michal Neoral

Study programme: Open Informatics

Specialisation: Computer Vision and Image Processing

Title of Diploma Thesis: Object Scene Flow in Video Sequences

Guidelines:

1. The thesis builds on the state-of-the-art Object Scene Flow (OSF) approach [1]. Familiarize yourself with this paper, code and results.
2. Familiarize yourself also with the other state-of-the-art scene flow estimation methods.
3. Improve the OSF method. Consider for instance speeding up the method or using temporal consistency of the results in a video sequence.
4. Evaluate the proposed improvements on publicly available dataset like KITTI [3,1] and HCI [4].

Bibliography/Sources:

[1] Menze, M. & Geiger, A. Object Scene Flow for Autonomous Vehicles. Conference on Computer Vision and Pattern Recognition (CVPR), 2015

[2] Vogel, C.; Schindler, K. & Roth, S. 3D Scene Flow Estimation with a Piecewise Rigid Scene Model. International Journal of Computer Vision, Springer, 2015

[3] Geiger, A.; Lenz, P. & Urtasun, R. Are we ready for autonomous driving? The KITTI vision benchmark suite Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on, 2012

[4] Kondermann, D.; The HCI Benchmark Suite: Stereo and Flow Ground Truth With Uncertainties for Urban Autonomous Driving The IEEE Conference on CVPR Workshops, 2016

Diploma Thesis Supervisor: Mgr. Jan Šochman, Ph.D.

Valid until: the end of the winter semester of academic year 2017/2018

L.S.

prof. Dr. Ing. Jan Kybic
Head of Department

prof. Ing. Pavel Ripka, CSc.
Dean

Prague, May 25, 2016

Acknowledgements

I would like to express my gratitude to my thesis adviser Mgr. Jan Šochman, PhD. for his valuable guidance and advice through the process of researching and writing this thesis. My thanks also go to my family and my girlfriend for providing me with support and encouragement throughout my years of study.

Thank you.

The research was supported by Toyota Motor Europe HS and by CTU student grant SGS17/185/OHK3/3T/13

Declaration

I declare that the presented work was developed independently and that I have listed all sources of information used within it in accordance with the methodical instructions for observing the ethical principles in the preparation of university theses.

Prague, 25th May 2017

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

V Praze dne 25. května 2017

....
podpis autora práce

Abstract

This thesis proposes several modifications of the Object Scene Flow (OSF) algorithm [MG15], algorithm for simultaneous estimation of 3D geometry and 3D motion called *scene flow*.

We focus on the addition of temporal consistency to the algorithm's output. Our proposed modification applies the temporal consistency to the OSF algorithm in the form of previously estimated independently moving objects propagated to the currently estimated frame using constant 3D motion assumption. The OSF does not use the scene flow estimated in previous frame nor any other estimated information.

We are also interested in the details of individual parts algorithm's time consumption. We propose a modification to speed-up algorithm more than three times.

We evaluate the progress on the KITTI'15 multi-frame dataset. We show that propagating the labels and the corresponding motion information using the estimated flow reduces the false negative rate (missed cars). However, this naïve propagation also increases the false positive rate significantly. We reduce the false positive rate with further proposed modifications, which result in the same false positive rates as the original OSF, but reduce false negatives by 35%. The proposed modifications also reduce an error of estimated scene flow on the KITTI'15 optical flow from 10.23% to 9.23% ranks 2nd in scene flow estimation category over whole image area, respectively 1st in scene flow estimation category over non-occluded areas only.

Keywords: scene flow, optical flow, disparity, autonomous driving, 3D scene geometry, rigid motion, motion segmentation

Supervisor: Mgr. Jan Šochman, Ph.D.

Abstrakt

Tato práce představuje několik modifikací algoritmu Object Scene Flow (OSF) [MG15]. Algoritmus je určen pro současné odhadování 3D geometrie scény a 3D pohybu ve scéně nazývaného *scene flow*.

Naše navrhnutá modifikace přidává temporální konzistenci do OSF algoritmu ve formě propagace nezávisle se pohybujících objektů odhadnutých v předchozích snímcích k vylepšení scene flow v právě počítaném snímku. Původní algoritmus OSF nevyužívá scene flow ani žádné jiné informace z předchozích snímků.

Detailně se zabýváme i časovými nároky individuálních částí algoritmu. Navrhujeme úpravu, která v průměru více než trojnásobně zrychluje algoritmus.

Naše veškeré modifikace vyhodnocujeme na testovací sadě KITTI'15. Ukazujeme, že propagace segmentace a korespondující informace o pohybu nezávisle se pohybujících objektů přispívá ke snížení míry nedetekovaných vozidel. Avšak tato propagace znatelně zvyšuje i počet falešně pozitivních detekcí, která je ovšem redukována dalšími prezentovanými úpravami. S vybranými modifikacemi je celková míra nedetekovaných vozidel snížena o 35%. Navrhnuté úpravy také snižují celkovou chybu odhadnutého scene flow, v benchmarku KITTI'15, z 10.23% na 9.23%. Modifikovaný algoritmus dosahuje prvního místa pro scene flow kategorii – vyhodnocení přes body viditelné v obou kamerách, respektive druhého místa pro scene flow kategorii – všechny obrazové body.

Klíčová slova: scene flow, optický tok, disparita, autonomní řízení, 3D geometrie scény, rigidní pohyb, segmentace dle pohybu

Překlad názvu: Použití Object Scene Flow ve video sekvencích

Contents

1 Introduction 1
1.1 Motivation 1
1.2 Contributions 2
1.3 List of Publications 3
1.4 Thesis Outline 3

2 Problem Formulation 5
2.1 Correspondence Problem 5
2.2 Stereo Vision 5
2.3 Optical Flow 6
2.4 Scene Flow 6
2.5 Challenges 7
2.5.1 Occlusion 7
2.5.2 Large Displacements 7
2.5.3 Illumination Condition 8
2.5.4 Textureless Surfaces and Repetitive Patterns 9
2.5.5 Aperture problem 10
2.5.6 Focus of Expansion Problem 10
2.6 Projection Pipeline 11

3 Related Work 13
3.1 Scene Flow Algorithms 13
3.2 Variational Methods 13
3.3 Piece-wise Rigid Methods 14
3.4 Deep Learning and Recent Development 14
3.5 On Temporal Consistency 15

4 Datasets 17
4.1 KITTI 17
4.2 HCI 18
4.3 FlyingThings3D, Driving, Monkaa 19
4.4 Discussion 19

5 Object Scene Flow Algorithm 21
5.1 Notation 21
5.2 Structure of the Algorithm 21
5.3 Time Consumption Analysis . . . 23
5.4 Discussion 25

6 Object Scene Flow with Temporal Consistency 27
6.1 Object Motion Labels Propagation 27
6.2 Ego-motion Outlier Redefinition 28
6.3 Robust Motion Hypotheses Generation 29
6.4 Temporally Consistent Superpixels 30
6.5 Optimisation Space Reduction . . 31

7 Experiments 35
7.1 Evaluation protocol 35
7.2 Object Scene Flow Variance Analysis 36
7.3 Evaluation of the Proposed Object Scene Flow Extensions 37
7.3.1 Summary 41
7.4 Comparison with the State of the Art 42

8 Future Work 45

9 Conclusion 47

Bibliography 49

A DVD content 57

B Tables 59

Chapter 1

Introduction

1.1 Motivation

For many computer vision tasks, it is essential to extract the geometry of surrounding area of a camera and independent motion in the scene. Typical examples of such tasks are driving assistance, autonomous driving or various outdoor robotic applications like Visual Odometry or SLAM.

3D motion and 3D geometry of the scene estimated together is called scene flow. Scene flow could then be used as input to the higher level algorithm for obstacle avoidance, motion planning or prediction of the other vehicles and pedestrians motion.

The most of the driving assistance or autonomous systems are using sensors for direct depth measurement – LiDARs. LiDAR has many advantages. The output of LiDAR sensor is directly the 3D point cloud of the surrounding areas. It is independent of the illumination conditions since it emits a light for measuring distance. LiDARs are more expensive than cameras (it costs more than an ordinary car), provide only sparse information (the best models provide 64-pixel high image of scanned area), and estimated depth of the scene is limited to the range up to 100 m. Thus, it is assumed that cameras will needed in the future of autonomous industry. However, scene flow estimation from cameras is an ill-posed problem under general assumptions.

Accurate and efficient estimation of the scene flow is still an unresolved problem. Figure 1.1 shows examples of estimated optical flow by state-of-the-art scene flow and optical flow estimation algorithms on our own sequences. Even the best methods often fail when the conditions differ from the ones of KITTI [MG15] for which the methods were developed. However, from the official KITTI results, it could be seen that stereo methods work better than monocular.

Most of the state-of-the-art methods consider only two consecutive frames. However, in practice, we have available whole video sequences, or we need to process just captured frames in an online manner. Not using frames from previous time steps leads to the inconsistent results and obviously, some important information is neglected.

This thesis is focused on the addition of temporal consistency to the existing algorithm. There are methods using temporal consistency, but they are focused on temporal consistency on the level from pixel correspondences to the temporal consistency of small planar patches. In particular, out proposed

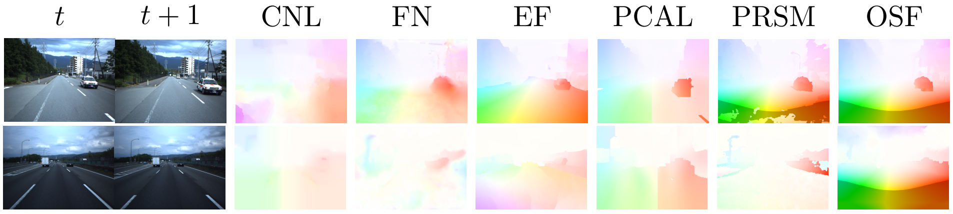


Figure 1.1: State of the art optical flow estimation results on our internal dataset. Compared optical flow methods: C+NL-fast (CNL) [SRB14], FlowNet (FN) [FDI$^+$15], EpicFlow (EF) [RWHS15], PCA-Layers (PCAL) [WB15] and scene flow methods PRSM [VSR15], OSF [MG15]. Images come from our internal dataset.

temporal consistency is on the level of the independently moving objects.

We propose several modifications of the Object Scene Flow (OSF) algorithm [MG15]. The OSF finds segmentation of independently moving objects as part of scene flow estimation. We show that adding temporal consistency leads to a more accurate scene flow estimation as well as more precise detection of independently moving objects.

1.2 Contributions

This thesis presents several contributions:

- The main contribution of the thesis is the addition of temporal consistency of independently moving objects to the OSF algorithm. Temporal consistency usage results to a reduction of estimated scene flow and optical flow error on the foreground and to decrease the missed car rate. It also stabilises scene flow estimation so that the same independently moving objects are detected more often through the sequence of images.

- We analysed the individual parts of the algorithm with attention to non-determinism of the algorithm. We identified a critical component, which was responsible for adding the most variance to the output of the OSF algorithm – the independent moving objects proposals. To stabilise the results we propose another two modifications:

  - A more robust dynamic ego-motion outlier definition. It replaces the original fixed threshold and allows to better distinguish between background and independently moving objects.
  - Using local optimised RANSAC [CMK03] instead of an non-optimised version of the algorithm increases the robustness of the algorithm and also decrease the variance of the results.

- We also report attempt with only partial success as application of temporal consistency on another level – temporally consistent superpixels.

- We provide a detailed analysis of the OSF components time complexity.

- To reduce time complexity of the algorithm, we propose the search space reduction of possible solution modification. Total time was reduced twice in average using the assumption that is no need to optimise estimated 3D geometry and 3D motion over areas without any independent motion hypothesis but ego-motion.
- We evaluated on KITTI'15 testing benchmark[1]. Our modifications reduced the erroneous pixel percentage from 10.63% to 9.65% of estimated scene flow, according to the original OSF. Moreover, **we achieved total 1st position in the scene flow category evaluated over non-occluded areas** and **total 2nd position in the scene flow category evaluated over whole image area**.
- Finally, we experimentally evaluated the ability of the original and extended OSF algorithms to detect independent moving objects. For this evaluation, we use KITTI'15 training dataset and report false positives (FP) and false negatives (FN) rates. The provided KITTI'15 dataset contains only partially annotated set of moving objects. We completed the annotation and added all moving objects in the scene. We show that proposed modifications reduce the number of missed vehicles by **35%**.

1.3 List of Publications

Parts of this thesis were published in:
Michal Neoral and Jan Šochman. Object scene flow with temporal consistency. In *22nd Computer Vision Winter Workshop(CVWW)*. Pattern Recognition and Image Processing Group, TU Wien & PRIP Club, Vienna, Austria, February 2017. ISBN: 978-3-200-04969-7.

1.4 Thesis Outline

The remainder of this thesis is organised as follows. Chapter 2 presents problem formulation and challenges during estimation. In Chapter 3, we review state-of-the-art methods and related works. Chapter 4 presents datasets and benchmark for evaluation. Chapter 5 describes the OSF algorithm in details. In Chapter 6, we present the proposed modifications of Object Scene Flow algorithm and gives the reasons for individuals modifications. Chapter 7 evaluates results of the proposed modifications and compares them with state of the art methods. Finally, the conclusions of this thesis are reported in Chapter 9.

[1] results at the time of publishing CVWW paper [NŠ17]

Chapter 2

Problem Formulation

The aim of this chapter is to introduce stereo vision, optical flow and scene flow estimation tasks. Then the basic challenges of aforementioned tasks are described. These problems and challenges are introduced only to the necessary level of detail to enable understanding of this thesis without deep knowledge in computer vision field.

2.1 Correspondence Problem

The purpose of finding correspondences is to decide which part of one image belongs to which parts of another image. The images could be taken at the same time with different cameras (stereo) or by the same camera at another time step (optical flow), or the configuration could be completely arbitrary. When capturing two or more images of the same real world scene, the search for correspondence is a search for a set of points in the scene which are displayed in one image and identify them in the second another.

The correspondence problem is not trivial (more about it in Section 2.5), and all of the following tasks could be seen as correspondence problem, where dense correspondence solution is preferred. If the correspondence is determined accurately, reconstruction of 3D scene geometry or 3D motion in the scene is given by triangulation.

2.2 Stereo Vision

The stereo vision is a computer vision task which uses 2D images and known relative calibration between cameras as an input to reconstruct the 3D geometry of the scene from individual cameras viewpoints. 3D information is estimated only from images without usage of specialised range measurement devices. The most common methods use rectified images to build a dense correspondence map between images called disparity map. The 3D information is then reconstructed from disparity and camera calibration using geometrical triangulation. Estimation of 3D geometry is not limited only to stereo pairs, arbitrary set-up of cameras could be used as well. Figure 2.1 shows the stereo estimation principle.

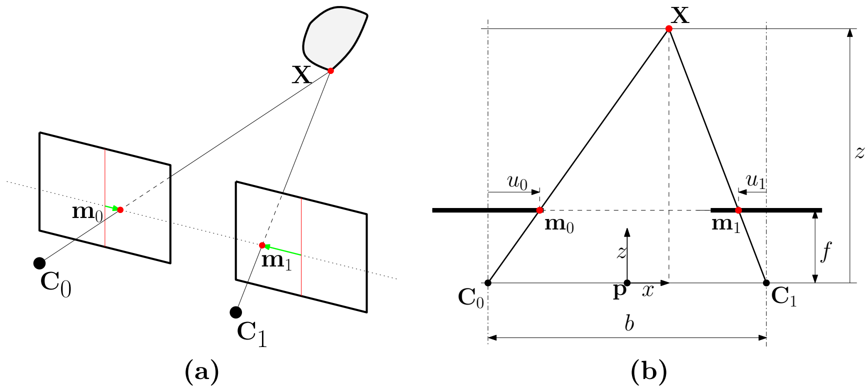


Figure 2.1: Stereo geometry principle for rectified images, where x, y and z are world coordinates, u and v are image coordinates, b is baseline between cameras, f is focal length, $\mathbf{X} = [x, y, z]^{\mathrm{T}}$ is point in 3D space, $\mathbf{m}_0 = [u_0, v_0]^{\mathrm{T}}$ and $\mathbf{m}_1 = [u_1, v_1]^{\mathrm{T}}$ are projected points to respective image planes, $\mathbf{C}_0$ and $\mathbf{C}_1$ are camera centres and $\mathbf{p}$ is centre of coordinates.

2.3 Optical Flow

The optical flow is 2D motion field which reflects the changes between two consecutive frames due to motion in the scene. The most commonly accepted definition of optical flow is as the apparent motion of brightness patterns in an image sequence. Figure 2.2 a) shows optical flow estimation principle. This definition refers only to estimate the correspondence between two images represented by 2D vector field. Thus, optical flow does not represent 3D motion in the scene, it represents only the projection of that motion in the image plane.

2.4 Scene Flow

Scene flow is a 3D motion field – 3D vector for each visible 3D point between two consecutive frames. Optical flow introduced above is a projection of scene flow to the image plane. Also, disparity map could be reconstructed from scene flow. Thus, the estimation of scene flow could be seen as the simultaneous optical flow and disparity estimation. A very similar task to the scene flow estimation is structure-from-motion (SfM), which also reconstruct the 3D geometry of a scene from different time steps or different camera positions. However, SfM relies on the assumption that reconstructed scene is static without any independently moving objects. The scene flow estimation could be transformed to the problem that identifies the correspondences among four images (two consecutive stereoscopic frames). Figure 2.2 b) show scene flow principle. The 3D geometry and 3D motion are then reconstructed from these correspondences. The scene flow estimation is an ill-posed problem and inherits the most typical challenges from stereo and optical flow. However, scene flow estimation uses more information therefore it is assumed that it

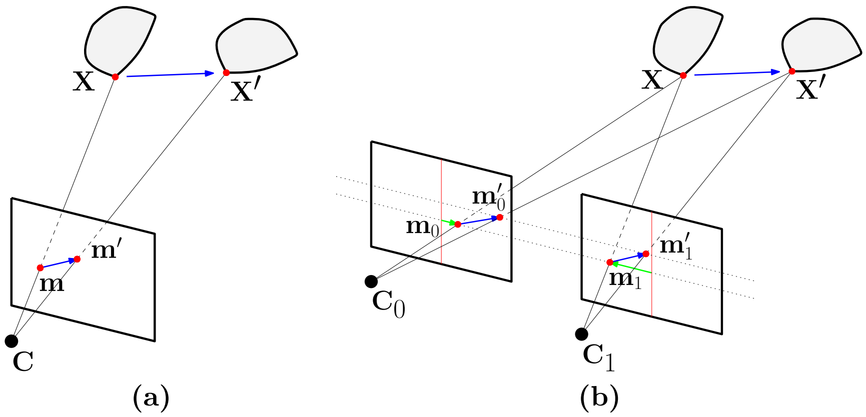


Figure 2.2: Optical flow and scene flow principle, where x, y and z are world coordinates, u and v are image coordinates, $\mathbf{X} = [x, y, z]^{\mathrm{T}}$ is point in 3D space, $\mathbf{m}_0 = [u_0, v_0]^{\mathrm{T}}$ and $\mathbf{m}_1 = [u_1, v_1]^{\mathrm{T}}$ are projected points to respective image planes and $\mathbf{C}_0$ and $\mathbf{C}_1$ are camera centres. Notation without comma means in current time step and notation with comma in the next time step. Image **(a)** shows optical flow principle – estimation of correspondences between two consecutive monocular images and image **(b)** shows scene flow principle – estimation of correspondences between consecutive two stereoscopic frames.

should bring better results. Scene flow, optical flow and disparity estimation could be solved as dense or sparse tasks.

2.5 Challenges

Scene flow, optical flow and disparity estimation could be seen as a correspondence problem. If correspondences between images were found accurate, we could reconstruct 3D geometry and 3D motion of scene directly from correspondences and calibration using triangulation (see Sec. 2.6). Correspondence estimation is, however, ill-posed task under the general assumptions. There are many problems in real lighting conditions or noise within the images.

2.5.1 Occlusion

Occlusion is a phenomenon that occurs when the scene is captured from two (or more) viewpoints. Occlusive pixels are such a group of pixels that is visible from one viewpoint but not from another viewpoint. Even small occlusions interfere with the assumption of illumination data consistency and can lead to poorly estimated correspondences in the image and loss of information about the hidden areas. Figure 2.3 shows the occlusion problem.

2.5.2 Large Displacements

Displacement is a situation, where pixels or a group of pixels that changed their position between the two images due to movement in the scene. Algorithms for estimating optical flow and scene flow most often assume presence of only

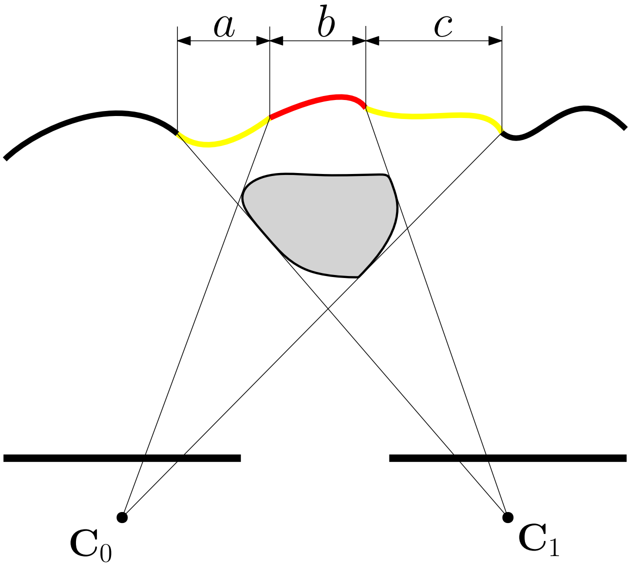


Figure 2.3: Occlusion phenomenon. The yellow areas a and c are visible only from one camera but not from another; the red area is not visible in cameras at all.

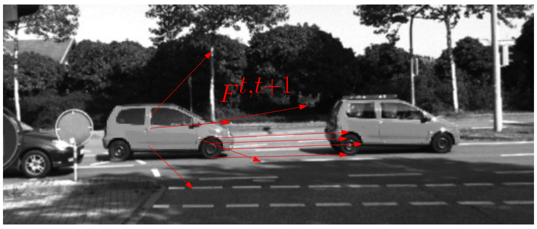


Figure 2.4: Large displacements problem. Fast motions or low frame rate could lead to bad estimation of optical flow $F^{t,t+1}$ between frames at time t and $t+1$. Images are from [MG15].

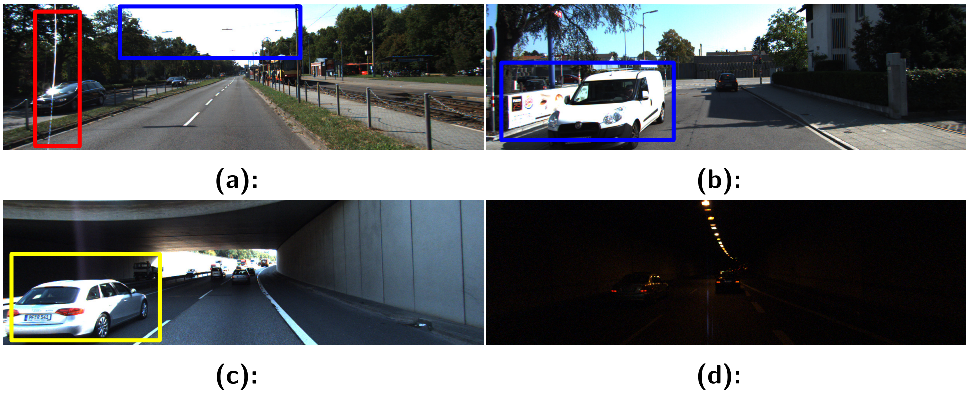
small motion in the consecutive image. Significant shifts in the image can lead to a flow estimation in the local minimum of possible solution and cause the algorithm to fail. Large displacements occur in images with fast motion or with low frame rate. Figure 2.4 shows displacement problem.

2.5.3 Illumination Condition

Matching is not an easy task even with simple lighting conditions. It is, therefore, no surprise that under general lighting conditions and non-Lambertian surfaces, this task is even more challenging.

One of the fundamental problems is oversaturation. This phenomenon occurs due to the small dynamic range of the camera used to capture the images and at the same time a high range of brightness in the scene (shadows and dark objects vs. bright objects in direct sunlight).

In scenes with individually moving objects, there is also a phenomenon of “strong brightness differences” caused by changing the position of an object



(a):

(b):

(c):

(d):

Figure 2.5: Various illumination conditions. Images **(a)** and **(b)** show oversaturated areas in the images within blue bounding boxes and reflection in the red bounding box. Image **(c)** shows strong brightness changes when the car goes to a bridge shadow. Image **(d)** shows driving during the night with low light. Images are from [MG15].

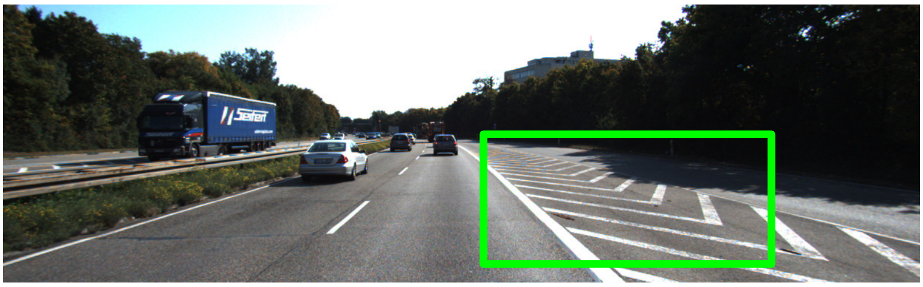


Figure 2.6: Repetitive patterns problems. Image shows repetitive pattern within the green bounding box.

from places with less light to a place with more light and vice versa.

Other problems in general scenes are reflective surfaces and specular highlights. Reflection problems are not caused only by human-made objects such as glass or reflective paint on vehicles or buildings, but the reflective surface could also be created from a puddle of water or a wet road.

A severe problem is also driving at night or under reduced visibility. Not only the dark surfaces but also the brightness changes produced by car lights or public lighting increase the difficulty of the correspondence task. The examples of different illumination conditions are depicted in Figure 2.5.

2.5.4 Textureless Surfaces and Repetitive Patterns

The textureless surface is a problem for matching, as the task becomes ill-posed, especially for pixel-based methods. It occurs, that pixel areas with insufficient texture result in false-positive correspondences. This problem is similar to oversaturation. Textureless surfaces problem cause an error in the stereo, optical flow and scene flow algorithms outputs.

Repetitive patterns are related to aperture problem. If the pattern in

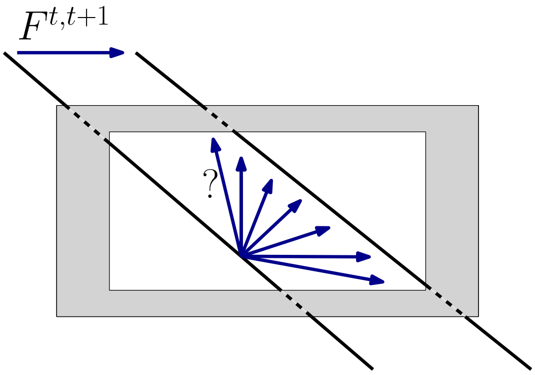


Figure 2.7: Aperture problem. The scheme shows a situation where is not possible distinguish the motion of the line between two timesteps. The necessary information is out of captured image area.

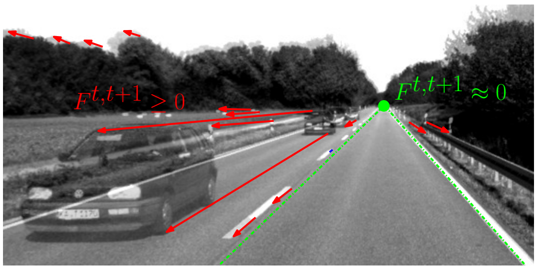


Figure 2.8: Focus of expansion problem. Image shows that size of optical flow $F^{t,t+1}$ between frames t and $t+1$ inducted by ego-motion depends on the position in the image area. Optical flow leads to zero close to the point of expansion (green point) and reduces the accuracy of estimated motion. The image is used from the KITTI'15 [MG15] dataset.

the image is repeated several times, it may happen that the algorithms end up at the local minima, which can lead to an algorithm failure in this area. Figure 2.6 shows an instance of a repetitive pattern problem.

2.5.5 Aperture problem

The aperture problem is a problem of motion uncertainty that is observed through the aperture, and there are no visible motion boundaries in the image. and it is demonstrated in Figure 2.7. The problem is mostly related to textureless regions or repetitive patterns regions. The estimated motion direction of such object is subject to considerable uncertainty. The algorithm cannot decide in which direction the object moves with such texture, when at least the boundary of the object is not visible in the image. This problem affects pixel-wise matching as well as block matching.

2.5.6 Focus of Expansion Problem

The focus of expansion (FoE) is a point in the image where optical flow vectors caused by camera motion intersect. If such a point is visible in the

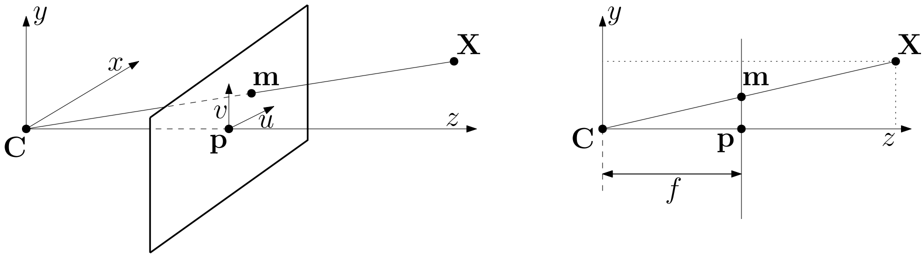


Figure 2.9: Pinhole camera model, where x, y and z are world coordinates, u and v are image coordinates, f is focal length, $\mathbf{X} = [x, y, z]^{\mathrm{T}}$ is point in 3D space, $\mathbf{m} = [u, v]^{\mathrm{T}}$ is projected point to image plane, $\mathbf{C}$ is camera centre and $\mathbf{p} = [u_0, v_0]$ is centre of coordinates in the image plane.

image, optical flow leads to zero in the areas near to FoE and scene flow estimation is an ill-posed problem in these areas. Figure 2.8 shows focus of expansion.

2.6 Projection Pipeline

The camera is a device which maps 3D information from scene to 2D image plane. Point in 3D space is labelled as $\mathbf{X} = [x, y, z]^{\mathrm{T}}$ and projected point to image plane as $\mathbf{m} = [u, v]^{\mathrm{T}}$. Projection of points according to standard camera model (pinhole camera) is depicted in Figure 2.9. Projection function is following:

$$\underline{\mathbf{m}} = \mathbf{P}\underline{\mathbf{X}}, \tag{2.1}$$

where $\mathbf{P} \in \mathbb{R}^{3\times4}$ is projective camera matrix and underlined symbols mean homogeneous coordinates, i.e. $\underline{\mathbf{X}} = \left[\omega\mathbf{X}^{\mathrm{T}}, \omega\right]^{\mathrm{T}}$, where $\omega \neq 0$ is the weight of the point. Homogeneous coordinates allow us to use complex transformations of points. Projection camera matrix

$$\mathbf{P} = \mathbf{K}\mathbf{R}\left[\mathbf{I} \ \ -\mathbf{C}\right] \tag{2.2}$$

is composed of inner calibration camera matrix $\mathbf{K} \in \mathbb{R}^{3\times3}$, rotation matrix $\mathbf{R} \in \mathbb{R}^{3\times3}$, $\mathbf{I} \in \mathbb{R}^{3\times3}$ is the identity matrix and $\mathbf{C} \in \mathbb{R}^{3\times1}$ is camera centre w.r.t. world coordinate system. Inner calibration camera matrix

$$\mathbf{K} = \begin{bmatrix} f & 0 & u_0 \\ 0 & f & v_0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.3}$$

contains three parameters: focal length f and $[u_0, v_0]^{\mathrm{T}}$, which represent the principal point of the image.

In a case that we have images from more than one camera, we can compute 3D information from the images. In other words, we can reconstruct 3D point $\mathbf{X}$ from the pair of 2D points $\mathbf{m}_0$ and $\mathbf{m}_1$. For the following calculations, we count on the situation that cameras are fully calibrated, and images are rectified (image plane is common for both images), which corresponds to proposed algorithm input images. Process of camera calibration exceeds a scope of this thesis and could be found in Hartley and Zisserman [HZ03].

Figure 2.2 **(b)** shows scheme of stereo reconstruction and its derivation from similarity of triangles. If the stereo correspondences are solved, derived reconstruction is following

$$z = \frac{b\,f}{d} \ , \quad x = z\frac{u_0 + u_1}{2f} \ , \quad y = z\frac{v_0}{f} \ , \tag{2.4}$$

where b is the baseline between camera centres and $d = u_0 - u_1$ is the disparity.

Chapter 3
Related Work

This chapter first reviews different approaches for scene flow estimation. There are discussed methods from variational approaches to methods with stronger motion assumption on objects level segmentation. Briefly, is discussed the recent development using deep learning. Then, follows the overview of the algorithms using temporal consistency during estimation of optical flow or scene flow.

3.1 Scene Flow Algorithms

Vedula et al. introduced the concept of scene flow [VBR$^+$99] as a three-dimensional vector field describing the motion of each three-dimensional point, visible in the camera, or each visible surface in a scene. Scene flow is also understood as a combination of dense stereo reconstruction and optical flow estimation, which are both challenging problems themselves. The algorithms for scene flow has to be able to solve the same problems as algorithms for optical flow and disparity estimation. That means occlusions, large displacements or radiometric challenges (see Sec. 2). But a simultaneous estimation would help to solve some ambiguities, as soon as more information is available (more cameras). However, there are also more parameters to estimate, since we estimate 3D geometry and 3D motion for each visible pixel.

3.2 Variational Methods

Various approaches have been proposed for optical flow estimation since Horn and Schunck [HS81]. Similarly to the optical flow methods, most of the state of the art approaches for scene flow estimation are also often based on variational methods. Vedula et al. [VBR$^+$99, VRCK05] presented two-step approach: in the first step the optical flows for each camera pair are estimated and in the second step a scene flow is fitted to the computed optical flow. However, during the second step, image intensities values are not used, and scene flow estimation fails over areas, where illumination changes are not induced by displacement but image perturbation (see Sec 2.5.3).

Huguet and Devernay [HD07] proposed a method for first simultaneous estimation of depth and optical flow in a single optimisation. Their joined approach handles with large displacements and occlusions for both, stereo and optical flow, nevertheless algorithm fails under the general illumination condition (66.90% scene flow error in the KITTI'15 benchmark [MG15]). Wedel et al. [WRV+08] parametrise scene flow in the image plane. Motivated by time consumption reduction, they decoupled scene flow estimation to optical flow and disparity estimation. Optical flows are estimated for each view using fixed pre-computed stereo estimation for each time step.

Valgaerts et al. [VBZ+10] generalised scene flow estimation task for uncalibrated stereo sequences (the estimation is up to scale) and they are using epipolar constraints instead. Unlike Huguet et al. [HD07] it does not use smoothness term for displacements. All of these methods [HD07, WRV+08, VBZ+10] are using only 2D parametrisation between individual images to describe scene flow. Basha et al. [BMK13] shows that using 3D parametrisation for scene flow brings better results to the algorithm output. However, stronger piece-wise rigid assumption leads to better results than variational approaches.

3.3 Piece-wise Rigid Methods

Recently, many successful methods [VSR13, YMU14, VSR15, MG15, LBA+16] started using small planar patches to represent a description of the scene instead of direct pixel-wise representations [HD07, VBR+99]. Segmentation of the scene into rigid planar regions increases robustness and decreases the number of parameters which must be estimated [VSR13].

Unger et al. [UWPB12] proposed segmentation based optical flow estimation with occlusion handling where affine transformation parametrises each segment in the image plane. Yamaguchi et al. [YMU13] extend his slanted-plane stereo algorithm [YHMU12] using continuous MRF to epipolar flow estimation. The motion of superpixel along epipolar lines describes flow. Later, the algorithm was improved using joint stereo and flow estimation of epipolar flow [YMU14]. However, this approach still assumed the static scene without independent motion. Vogel et al. [VSR11, VSR13, VRS14, VSR15] further reinterpreted scene flow as joint task of shape and motion of each superpixel and proposed discrete CRF containing regularisation of motion, geometry and occlusions superpixels. Lv et al. [LBA+16] uses the same representation of a scene, however they focus on the time complexity of the algorithm.

The Object Scene Flow (OSF) [MG15] further introduces an idea that scene flow is composed from only a small number of independent motions. This assumption leads to a strong regularisation for scene flow computation and leads into a more accurate scene flow. Each independent motion is further restricted spatially, allowing independently moving object segmentation.

3.4 Deep Learning and Recent Development

Since seminal work of Krizhevsky et al. [KSH12] for image classification, the deep neural network started to be applied to a wide range of computer vision tasks.

Dosovitskiy et al. [FDI+15] referred the importance of large dataset with high-quality ground truth for learning optical flow. He proposed synthetic dataset of 2D flying chairs for deep learning and end-to-end method of learning optical flow with convolutional neural networks. His method demonstrates that is possible to learn optical flow using CNN. However, his approach does not achieve state-of-the-art results in popular benchmarks [MG15, BWSB12]. Recently, Ilg et al.[IMS+16] present an improved version of Dosovitskiy's approach [FDI+15], which achieved the best position on the KITTI'15 optical flow benchmark. Moreover, the algorithm belongs to the fastest on the KITTI'15.

Mayer et al. [MIH+16] proposed the first method for learning scene flow by convolution neural network by simultaneously learning disparity and optical flow on their proposed datasets (see Sec. 4.3).

More recently, there started appeared approaches using less number of cameras for estimation, than it is specified from the minimal configuration of the individual problems. Zhou et al. [ZBSL17] extended Godard et al. [GMAB16] for monocular depth estimation and learned depth and ego-motion at the same time. Their approach can estimated scene flow of static scenes from monocular camera sequence. Vijayanarasimhan et al. [VRS+17] proposed a learning scene flow in dynamic scenes by simultaneous estimation of depth, ego-motion and motion of independently moving objects from the monocular camera. However, for now, these methods achieve significantly worse results than methods using a standard configuration for scene flow estimation.

Currently, the best methods for stereo estimation [SW16, GK16, KMD+17] and optical flow estimation [IMS+16][1] are using deep learning, while the best positions for and scene flow estimation still belongs to "non-learning" methods [VRS14, VSR15, MG15, LBA+16, NŠ17].

3.5 On Temporal Consistency

Since Murray and Buxton [MB87], various approaches using temporal consistency have been proposed for optical and scene flow. Some of them rely on smoothness assumption of trajectory over multiple frames. A spatio-temporal smoothness term for the optical flow was proposed in [MB87]. However, the algorithm does not work well for large displacements. Irani [Ira02] estimates optical flow over long trajectories using multi-frame subspace constraints. Volz et al. [VBVZ11] proposed adaptive trajectory regularisation over five consecutive frames. Motion fields of all frames are parametrised with respect to the central reference frame. All of the above-listed methods use temporal consistency on 2D pixel-level for optical flow estimation.

Devernay et al. [DMG06] show that tracking of 3D points and surfels (small planar square regions) bring better results than tracking only 2D points. They proposed extension of [LK+81] using multiple cameras for temporally consistent scene flow estimation. Rabe et al. [RMWF10] used

[1] with minimal optical flow configuration i.e. single camera

extended Kalman filter [K$^+$60] for tracking, but instead of tracking matched features they tracked dense scene flow computed by [WRV$^+$08]. Although the algorithm is real-time, its use is rather limited, since it is not able to handle fast motions. Basha et al. [BMK13] parametrise model according to 3D scene flow constraints from 3D point clouds from several time steps and simultaneously estimate depth and scene flow. They also show that 3D parametrisation leads to better results than 2D parametrisation of scene in methods of [DMG06, WBV$^+$11]

Using robustly linked frames, Hung et al. [HXJ13] proposed optical flow and stereo estimation from long-temporal motion trajectories but algorithm needs the whole sequence for the computation, therefore it is inappropriate for online scene flow estimation.

Recently, Vogel et al. [VRS14, VSR15] achieved temporal coherence using sliding temporal windows for their both viewpoints and multi-frames consistent model. They also proposed temporally consistent piecewise-planar segmentation of the scene with an assumption of constant 3D motion. Their approach belongs to the state-of-the-art methods ranked on the KITTI'15 benchmark. However, the method does not produce independent motion segmentation like OSF, which is not only desirable as a function output but as a strong regularisation for the scene flow estimation as well.

In the context of methods using temporal consistency listed above, our proposed independent motion propagation (Section 6.1) is a temporal consistency of the highest level. Instead of enforcing individual pixels, small patches or superpixel consistency over several frames, our proposed approach uses propagation of whole objects' segmentations and their estimated motions.

Chapter 4

Datasets

Recent research in autonomous driving requires datasets and benchmarks with realistic data and dense ground truth. As the acquisition of ground truth scene flow for real world data is complicated, only a few datasets were published for its evaluation. Benchmarks for scene flow should cover all possible failures of algorithms. This chapter presents some benchmarks for scene flow evaluation.

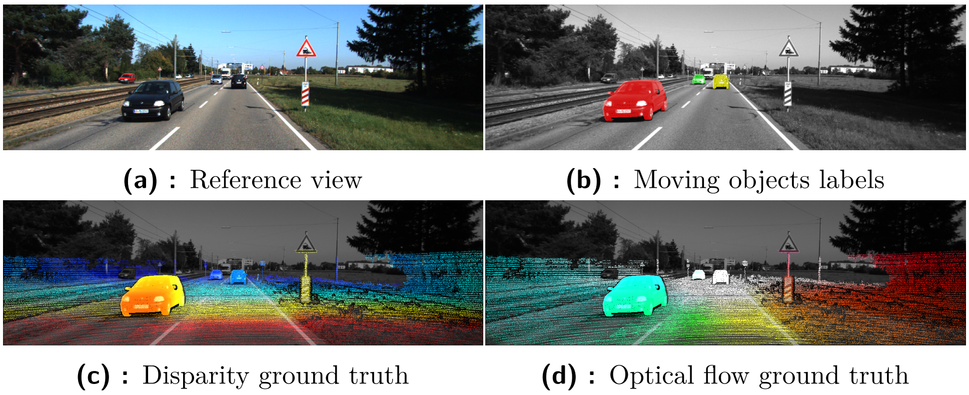
4.1 KITTI

KITTI is very heterogeneous dataset and contains several benchmarks like stereo, optical flow, odometry, object detection and tracking benchmarks as KITTI'12 [GLSU12] or road benchmark [FKG13]. Recently, KITTI'15 dataset [MG15] was published for scene flow, optical flow and stereo containing independently moving objects.

KITTI is a very popular dataset with more than 60 submissions in optical flow and stereo category and with more than 15 in scene flow category. Dataset was captured from a moving platform using four cameras with global shutter (two grayscale and two colours). For producing high-density ground truth, they used Velodyne laser scanner and state of the art localisation system [GLU13]. Dataset is focused on autonomous driving application and highly difficult conditions. It contains scenes with oversaturation, flares, transparent surfaces, strong illumination changes and reflections. Also, it contains displacements longer than 250 pixels and disparities over 150 pixels. Limitations of this dataset are sparsity of ground truth and the range of ground truth (limitations of used range finder).

The biggest difference between KITTI'12 and KITTI'15 dataset is that KITTI'12 contains only static scenes, while KITTI'15 contains 400 scenes with ground truth over moving moving cars and vans, which was computed using fitting CAD models to the scanned depth data. But CAD models were not chosen for every object in the scenes. The other moving objects are not presented (included some cars and vans far from the camera or with partial overlap). As another part of the dataset are provided only bounding boxes for all objects in the sequences, but this is focused on the identification and orientation. Thus, no are distinguishing between moving and static objects. The example of the input image and ground truth data is shown in Figure 4.1.

We are interested in the scene flow estimation in sequences containing



(a) : Reference view

(b) : Moving objects labels

(c) : Disparity ground truth

(d) : Optical flow ground truth

Figure 4.1: Sample images from KITTI'15 [MHG15] dataset with the ground truth over independently moving objects.

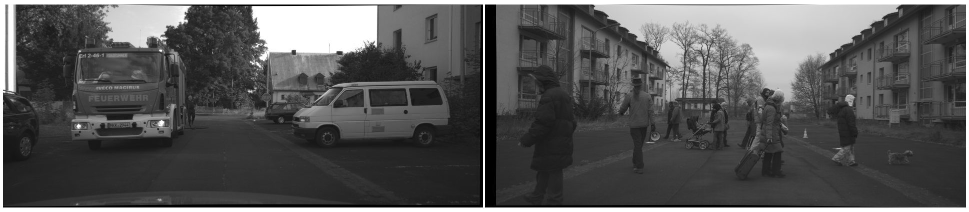


Figure 4.2: Sample input images from HCI benchmark suite [KNH$^+$16]

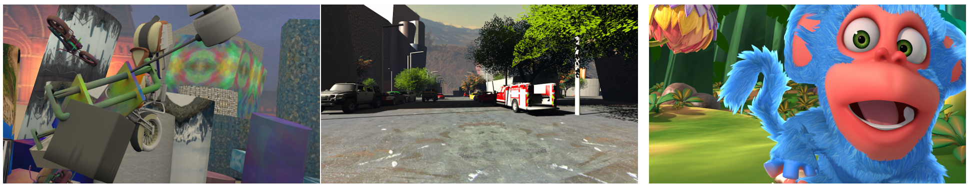
independently moving objects. Thus, we are using KITTI'15 dataset for testing of our modifications influence.

4.2 HCI

HCI optical flow and disparity benchmark [KNH$^+$16] also focuses on the application in the autonomous driving and contains sequences from moving platform. The dataset contains 55 sequences captured in high resolution, high dynamic range and high frame rate with 19 to 100 grayscale images per sequence. Figure 4.2 shows an example of input images. HCI is even more diverse compared to KITTI. Scene flow estimation is made hard by a variety of radiometric challenges, bad weather sequences, bad light conditions during night, different years seasons, fog, snow, raindrops on the windshield, reflecting puddles on the road. On the other hand, it has poor variety at a location, since the dataset was captured in the controlled environment of a single street.

There are several cons to using this dataset for our purposes. First cons is that the dataset contains only grayscale images, while KITTI'15 contains both RGB and grayscale. The most important problem for evaluation of out algorithm is the missing evaluation of scene flow over independently moving objects. HCI excludes pixels over independent moving objects from evaluation. From these reasons, we are not using this dataset for evaluation.

Note, that the benchmark contains only two submissions for optical flow



(a) : FlyingThings3D **(b) :** Driving **(c) :** Monkaa

Figure 4.3: Sample images from scene flow datasets FlyingThings3D, Driving, Monkaa [MIH$^+$16]

and three submissions for disparity estimation. All of them were added by authors of the HCI benchmark suite from publicly availed source codes. Thus, the evaluation does not provide a high level of comparison with other approaches.

4.3 FlyingThings3D, Driving, Monkaa

FlyingThings3D, Driving, Monkaa scene flow datasets [MIH$^+$16] were introduced to be used for training neural networks. All three datasets are computer generated and contain over 35000 frames with dense stereo and optical flow ground truth in total. Figure 4.3 shows examples of input images for all three datasets. FlyingThings3D contains sequences with an arbitrary number of objects moving along randomised 3D trajectories, Monkaa contains sequences from animated movie and Driving is dataset focused on autonomous driving. However, we did not use this dataset for evaluation. Dataset is rendered with low realism, contains only one sequence for autonomous driving, and we cannot easily compare with other algorithms since there is no public benchmark.

4.4 Discussion

There are few other datasets which are used for both optical flow and disparity evaluation. The Middlebury dataset [BSL$^+$11] was captured in a laboratory environment with very high precise accuracy but contains only twelve short sequences for training and twelve sequences for evaluation. The MPI Sintel [BWSB12] dataset is derived from an animated short film. It contains dense optical flow and scene flow ground truth for sequences with various illumination conditions, locations and motions.

As none of these datasets is focused on the autonomous driving, we are using KITTI'15 [MG15] for all important comparisons.

Chapter 5

Object Scene Flow Algorithm

The main goal of this chapter is to apprise the reader of the Object Scene Flow algorithm (OSF), introduced by Menze and Gaiger [MG15]. The OSF is the algorithm for 3D scene flow estimation using strong assumptions about individually moving objects in the observed scene.

The OSF decomposes each dynamic scene into a small number (hundreds) of 3D planar patches using slanted-plane model [BT99, YHMU12]. The algorithm assumes that each patch belongs to one of a few independently moving objects, each with its own rigid motion (six degrees of freedom).

In brief, each of the planar patch is parametrised by four variables: Three of them for the plane parameters and one for a label index. Each label corresponds to an object motion. Further, it is assumed that the set of independently moving objects is small (up to ten). Scene flow estimation is solved as a labelling problem, where each of the planar patches is assigned to one of the rigid body motions using a discrete-continuous CRF. The CRFs objective is defined as a weighted sum of unary and pairwise terms computed from disparity, superpixels, sparse optical flow and motion candidates.

5.1 Notation

The OSF algorithm decomposes a dynamic scene into a set of 3D planar patches $\mathbf{s}_i = (\mathbf{n}_i, l_i)$, where $\mathbf{n}_i$ is a normal of the plane, l_i is a label of 3D motion $l_i \in \{1, \ldots, |O|\}$ and O is a set of a few independent motions. Each 3D motion $\mathbf{o}_k \in O$ is parametrised by rotation $\mathbf{R}_k \in \mathrm{SO}(3)$ and translation $\mathbf{t}_k \in \mathbb{R}^3$. Each plane normal $\mathbf{n}_i$ is computed from a superpixel $i \in S$, where S is a set of superpixels in the reference frame, which is obtained by fitting a plane to the depth values estimated from the disparity. The plane parameters provide the mapping between 3D points $\mathbf{X}_i = [x_i, y_i, z_i]^{\mathrm{T}}$ and its corresponding 2D points $\mathbf{m}_i = [u_i, v_i]^{\mathrm{T}}$. The frame t is considered to be the reference frame, $t + 1$ is the next frame and $t - 1$ is the previous frame, etc.

5.2 Structure of the Algorithm

The structure of the OSF algorithm is shown in Figure 5.1. Following text introduce the OSF algorithm in details.

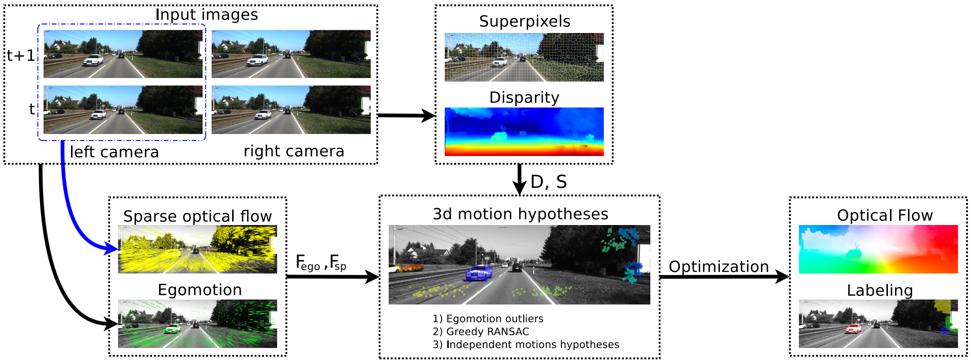


Figure 5.1: Overview of the Object Scene Flow algorithm [MG15]. Superpixel segmentation S, disparity D, sparse-flow F_{sp} and ego-motion F_{ego} of the camera are estimated from input stereo images. Then the independent motion candidates are estimated. Labels of motion candidates to proper segments are assigned during optimisation. Finally, the output in the form of scene flow and label map are computed from motion candidates and segments parameters using MP-PBP [PZBS14] and TRW-S [Kol06].

Input. The input to the algorithm are two consecutive stereo frames $(\mathbf{I}_t^l, \mathbf{I}_t^r)$, $(\mathbf{I}_{t+1}^l, \mathbf{I}_{t+1}^r)$. The OSF model needs the input images to be rectified. The left image at time t, $\mathbf{I}_t^l$, is used as the reference image.

Superpixel segmentation and initial disparity estimation. The superpixels and the initial disparity is computed by SPS-St [YMU14] and SGM [Hir05] respectively. All reference view pixels are segmented into superpixels and each superpixel is assumed to correspond to a planar 3D patch in the scene. The planar patch plane is computed by fitting a plane to the corresponding disparity measurements.

Ego-motion and sparse optical flow. Then, the camera position and orientation are estimated. Ego-motion is computed using Geiger et al. [GZS11] visual odometry algorithm and it assumes that dominant motion of the scene is induced by the motion of the camera (a car with recording platform).

The algorithm relies on sparse features detected by corner and blob detector and estimates the ego-motion by minimising the reprojection error using Gauss-Newton optimisation.

Next, sparse optical flow is computed from set of correspondences between $\mathbf{I}_t^l$ and $\mathbf{I}_{t+1}^l$ images. The correspondences are computed by the same feature and correspondence detector [GZS11] as described above but without estimation of single motion model. Instead, the set of sparse correspondences over whole images are returned (even for independently moving objects). Sparse optical flow is used as a clue to independent motion hypotheses as described bellow.

Motion hypotheses. The rigid body motion hypotheses of independently moving objects are computed next. The ego-motion outliers are found, and they are used as an input to a sequential RANSAC which greedily produces hypotheses. The number RANSAC sequences ran over whole image area and the number of inner RANSAC iterations are set to constant (up to hundred). Non-maxima suppression is applied for motion models as the last step of hypotheses estimation.

Optimisation. Finally, the CRF is formulated in order to assign planar patches with the motion hypotheses as mentioned above.

The CRF function is formulated according to a data-term $\mathbf{E}_\mathrm{D}$ and smoothness (regularisation) term $\mathbf{E}_\mathrm{S}$ [MP76]:

$$\mathbf{E}\left(\mathbf{s},\mathbf{o}\right) = \mathbf{E}_\mathrm{D}\left(\mathbf{s},\mathbf{o}\right) + \lambda\mathbf{E}_\mathrm{S}\left(\mathbf{s}_\mathrm{i},\mathbf{s}_\mathrm{j}\right), \tag{5.1}$$

where λ is weight of smoothness term. Data term evaluates the assumption of the constancy in appearance between corresponding pixels over all four images. Data term is computed for each superpixel and each object from the set of possible motion hypotheses as a summation of matching costs of all pixel inside the superpixel. The OSF uses dense and sparse matching costs. Dense matching cost is defined as Hamming distance of appropriate census descriptors. Sparse matching cost is defined using l_1 norm between warped images using sparse feature correspondences 5.2.

The OSF smoothness term relies on the assumption that adjacent superpixels assigned with the same independent object (and its motion) have smooth transitions between depth and orientation. Smoothness term penalises the undesired relation between adjacent superpixels.

The OSF uses max-product particle belief propagation [PZBS14] and tree-reweighted message passing [Kol06] for the optimisation. Details can be found in the original paper [MG15]. The estimated dense scene flow is computed from the planar patches parameters.

5.3 Time Consumption Analysis

The OSF algorithm is very time-consuming, it takes 50 minutes [1] per frame. In this section, we investigate algorithm time consumption of individual parts to find out a possible opportunity of algorithm speed-up.

The most demanding part of the algorithm is the optimisation of the resulting scene flow, which takes about 32 minutes [2]. Figure 5.2b shows the relative time complexity between initialisation and optimisation of the algorithm. The reported durations are average times measured on the KITTI'15 dataset [MG15].

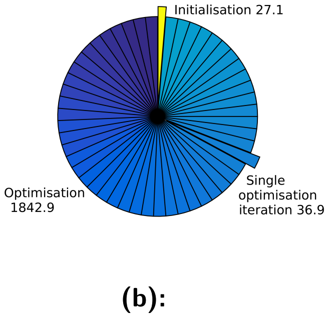
Initialisation part analysis. As seen in Figure 5.2b, initialisation steps take only a fraction of the overall running time. To initialisation part of the algorithm belongs following steps. Computation disparity and superpixel segmentation of images using SPS-St [YMU14] is the first step of the algorithm and takes on the average 6.93 seconds. Note that the most of this time takes

[1] measured time comes from original paper

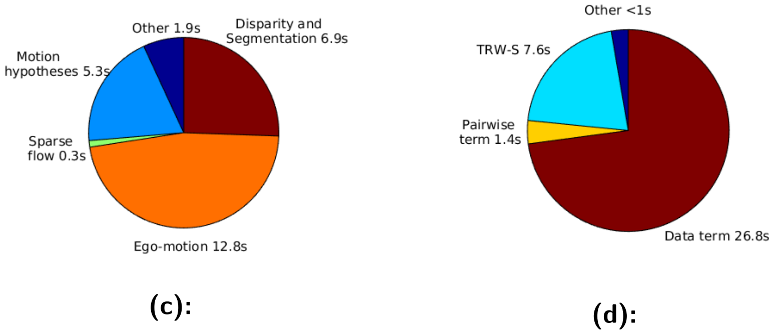
[2] single core Intel i5-2.4GHz, original paper presented 50 minutes

	parts	OSF [s]	OSF-BG [s]
initialisation	disparity and segmentation	6.93	
	ego-motion estimation	12.75	
	sparse optical flow	0.28	
	motion hypotheses	5.27	
	other initialisation	1.88	
optim.	data term comp.	26.88	7.52
	pair-wise term comp.	1.37	0.32
	TRW-S	7.63	1.66
	other optimisation	0.98	0.66
	total (50 optim. iters)	1870.20	535.11

(a):



(b):



(c):

(d):

Figure 5.2: Time consumption analysis of the Object Scene Flow algorithm [MG15]. The top left table shows durations of individual steps of the algorithm, where OSF means a original algorithm and OSF-BG proposed search space reduction. The top right chart shows relative time complexity between initialisation and optimisation. Bottom left chart shows relative time complexity among individual parts of initialisation and bottom right chart shows relative time complexity among individual parts of optimisation (times are shown for one iteration).

up to the initial disparity estimation using SGM [Hir05]. The most time-consuming part of initialisation is finding ego-motion [GZS11], which takes 12.75 seconds.

Searching for hypotheses of moving objects using the information from the previous steps, and the RANSAC scheme takes 5.27 seconds. Figure 5.2c shows relative time complexity of the initialisation parts of the OSF algorithm.

Initialisation of the OSF algorithm is possible to speed-up several ways – e.g. using real-time implementation of SGM algorithm [BHF$^+$10, SLAR14, GR10], application of real-time correspondence and ego-motion estimation [BW16] (even originally applied algorithm [GZS11] is able to run in real-time) of effectiveness parallelisation of sequential RANSAC. However, significantly the most time-consuming part of the OSF algorithm is optimisation. Thus, speed-up of initialisation has been left as future work.

Optimisation part analysis. As mentioned in the previous paragraphs, the slowest part of the algorithm is the optimisation itself. The times listed in the following paragraph relate to one optimisation iteration and their value is averaged over all image pairs in the KITTI'15 dataset. A total number of iteration is set to 50^3.

By far the most expensive part of the optimisation is the data-term computation for all possible superpixel shape combinations and possible hypotheses of individually moving objects generated by the particle filter. It takes 26.88 seconds. Next, there are generated pairwise terms for the adjacent superpixels in 1.37 seconds. Optimisation by TRW-S [Kol06] for computed data terms and pairwise terms takes 7.63 seconds. The relative time complexity of the optimisation parts are showed in the Figure 5.2d.

One of the algorithm speed-up is achieved by the number of iteration reduction. This approach is presented and evaluated in the original OSF paper. Other speed-up options of optimisation are the reduction of possible solution search space, parallelisation on GPU, etc. We propose first of the listed speed-ups in Section 6.5.

5.4 Discussion

The assumptions of locally planar superpixels and several rigidly moving objects which explain motion in the scene have significantly improved the results of algorithms in optical flow and scene flow benchmarks [GLSU12, MG15]. Although the OSF belongs to the state of the art methods, we have identified several shortcomings.

- We run the algorithm on a video sequence and observe that it occurs that the algorithm fails to identify moving objects that were considered moving in previous frames. This effect leads to a poor estimate of the scene flow and also to the loss of information about moving objects. Along with that, the IDs of objects are not preserved thought the sequence. Section 6.1 focus of this problem.

- The quality of the estimated scene flow is strongly dependent on initialisation of the random generator. Total variance of estimated optical flow is more than 3% for the scene flow computed on the same input data, more about variance problem in Chapter 7.2.

- The algorithm is very time-consuming (see Sec. 5.3). One reason is non-optimal algorithm implementation (no parallelisation) and the second reason is very complex optimisation (see Sec. 6.5).

- Other problems are identified as: the wrong estimation of small vehicles near to focus of expansion, more than one hypothesis on large vehicles, failed ego-motion estimation and limitations of greedy hypotheses generation. These problems have been left as future work.

[3] according original paper settings

Chapter 6

Object Scene Flow with Temporal Consistency

In this chapter, we present our extensions of the OSF algorithm. We focus on the temporal consistency of the independently moving objects. Further, we introduce ego-motion outlier redefinition – the method for more precise distinguish between background and moving objects. Next, we propose robust motion hypotheses generation – an application of local optimisation within the algorithm for searching proposals of independently moving objects. Then, modification using temporally consistent superpixels and reduction of optimisation space, which is focused on speed-up of optimisation part of OSF algorithm. All the proposed modifications are evaluated experimentally in Chapter 7.

6.1 Object Motion Labels Propagation

As noted above, the OSF does not use any information from the previous stereo image pairs. We expect that adding temporal consistency will lead to a more accurate scene flow estimation. We also expect that some objects that are missed by standard OSF will be detected thanks to the temporal consistency. Finally, the object labels should become stable throughout the sequence.

Following text uses notation introduced in Chapter 5 and Chapter 2. Assuming constant velocity, we propagate estimated motion parameters $\mathbf{o}_k^{t-1}$ from the frame $t-1$ and use them for estimation of $\mathbf{o}_k^t$ in frame t. However, we do not simply use $\mathbf{o}_k^{t-1}$ as a motion candidate at frame t. Instead, we use disparity and sparse correspondences between the frames t and $t+1$ to find a good candidate motion $\mathbf{o}_k^t$ as it is shown in Figure 6.1.

Let $L_k = \{i; l_i = k\}$ be a set of indexes of all planar patches $\mathbf{s}_i$ assigned with the same motion $\mathbf{o}_k^{t-1}$ and let $\mathbf{X}_{L_k}^t$ be a set of all 3D points associated to all segments from L_k. We get a set of 3D points $\hat{\mathbf{X}}_{L_k}^{t+1}$ using a constant motion assumption.

$$\hat{\mathbf{X}}_{L_k}^{t+1} \simeq \mathbf{R}_k^{t-1}\mathbf{X}_{L_k}^t + \mathbf{t}_k^{t-1} \qquad (6.1)$$

Since the constant motion assumption is only approximately valid, we use 3D points $\mathbf{X}_{L_k}^t$ and $\hat{\mathbf{X}}_{L_k}^{t+1}$ only to estimate the bounding box of expected object location. The actual positions of 3D points are then estimated from disparity and sparse correspondences in the following way. We reproject the

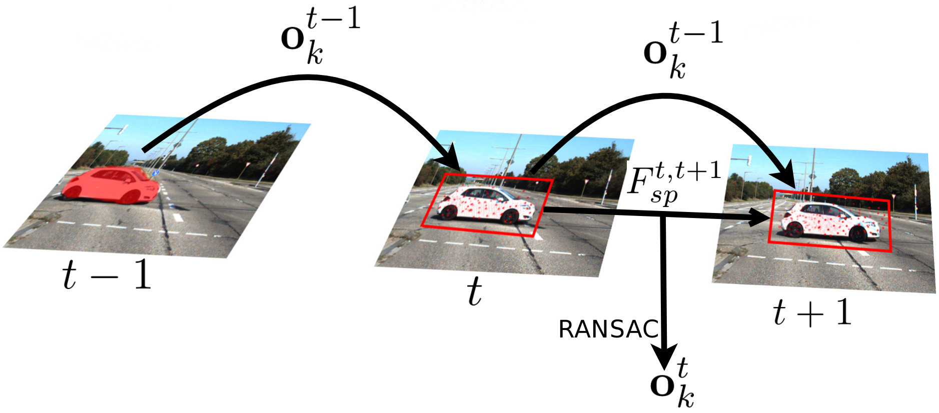


Figure 6.1: Scheme of object motion labels propagation. Dense 3D point cloud $\mathbf{X}_{L_k}^{t-1}$ is computed from segments assigned with motion $\mathbf{o}_k^{t-1}$. 3D points $\mathbf{X}_{L_k}^{t-1}$ are transformed by $\mathbf{o}_k^{t-1}$ to the frame t and $t+1$ as $\mathbf{X}_{L_k}^{t}$ and $\hat{\mathbf{X}}_{L_k}^{t+1}$ respectively. Sparse correspondences $F_{sp}^{t,t+1}$ with larger density for motion estimation are computed in the appropriated areas where $\hat{\mathbf{X}}_{L_k}^{t}$ and $\hat{\mathbf{X}}_{L_k}^{t+1}$ are reprojected.

3D points $\mathbf{X}_{L_k}^{t}$ and $\hat{\mathbf{X}}_{L_k}^{t+1}$ back to the image plane as 2D points $\hat{\mathbf{m}}_{L_k}^{t}$ and $\hat{\mathbf{m}}_{L_k}^{t+1}$, respectively. We compute sparse flow $F_{sp}^{t,t+1}$ correspondences [GZS11] between frames t and $t+1$, with larger density than in the original OSF (five times in our case). These correspondences are computed in the image area bounded with the smallest rectangular bounding box containing all reprojected points $\hat{\mathbf{m}}_{L_k}^{t}$ and $\hat{\mathbf{m}}_{L_k}^{t+1}$, respectively. We enlarge the bounding box by 20 pixels at each side to increase robustness. For all computed correspondences, we estimate their corresponding 3D points $\mathbf{X}_{F_{sp}}^{t}, \mathbf{X}_{F_{sp}}^{t+1}$ using stereo camera calibration and estimated disparity.

To remove obvious outliers, we remove all points $\mathbf{X}_{F_{sp}}^{t+1}$ (with their $\mathbf{X}_{F_{sp}}^{t}$ correspondences) which are further away from the $median(\hat{\mathbf{X}}_{L_k}^{t+1})$ than a threshold θ_{sp}[1]. We also remove all correspondences which have similar motion as the camera ego-motion. Motion hypothesis candidate $\mathbf{o}_k^{t} = (\mathbf{R}_k^{t}, \mathbf{t}_k^{t})$ is estimated on the remaining correspondences by RANSAC. We propagate every object motion $\mathbf{o}_k^{t-1}$ except the ego-motion.

6.2 Ego-motion Outlier Redefinition

The motion hypothesis propagation has a positive effect on the error of the estimated scene flow and decreases the number of missed vehicles. However, the label propagation also increases false positive detection rate (Table 7.1). This is caused mostly by propagating additional false positive detection from previous frames. False positive detection could be seen in Figure 6.2.

As discussed above, the OSF algorithm finds 3D motion hypotheses as

[1] $\theta_{sp} = 3\,\mathrm{m}$; Similar process as used in [MG15]

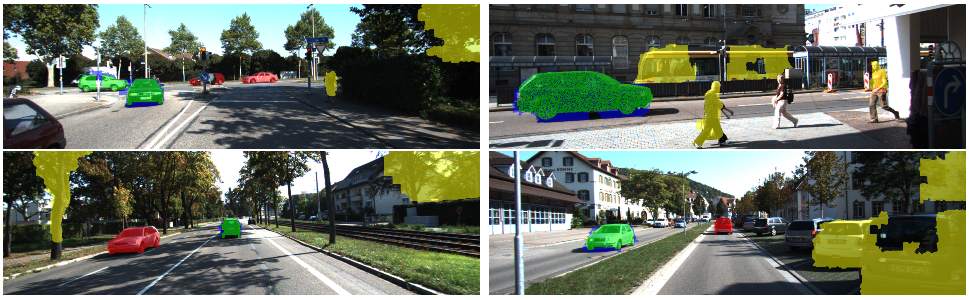


Figure 6.2: Evaluation of independently moving objects labelling from Object Scene Flow algorithm [MG15] on the KITTI'15 dataset. Missed vehicles are coloured in red; correctly detected vehicles are in green, and falsely detected vehicles are coloured in yellow. As false positive detections are considered also moving objects e.g. cyclists, trucks, persons, etc. since foreground ground-truth contains only moving cars.

ego-motion outliers in sparse flow correspondences. A correspondence is considered as ego-motion outlier when its end-point-error $E_{epe}(u, v)$ is greater than a fixed threshold (2 px) for all (u, v) where F_{sp} is defined.

Figure 6.3 shows the ego-motion outliers of the original approach (labelled with red colour). It could be observed that the fixed threshold works well at medium flow magnitudes but worse at the boundary of the images where the optical flow is larger and a small disparity error causes significant EPE. To eliminate this effect, we propose to use a dynamic threshold which depends on the motion magnitude. Correspondence in the image point (u, v) is labelled as ego-motion outlier if

$$(E_{epe}(u, v))^2 \geq \max\left(\left\|F_{ego}^{u,v}\right\|_2, \theta_{min}\right), \tag{6.2}$$

where $\theta_{min} = \sqrt{2}$ is a minimal optical flow threshold to increase robustness.

Application of this change is shown in Figure 6.3. The false ego-motion outliers disappear at image edges and the true estimated outliers are found on the distant vehicles.

6.3 Robust Motion Hypotheses Generation

Examining further the results, most of the remaining errors are caused by the random nature of the algorithm. Depending on the initialisation, we observed a high output variance. Due to inaccurate matches, this approach of multi-instance model fitting could produce imprecise models. Inaccurate models hypotheses are then discarded during labelling and optimisation step of the OSF algorithm. Particle filtering in the optimisation loop should fix the inaccuracy of the models, nevertheless this works only for small deviations.

Figure 6.4 compares the best and the worst case from 10 randomly initialised runs of the algorithm on the same input data. Possible reason of the poor hypotheses are often a small number of correspondences or inaccurately

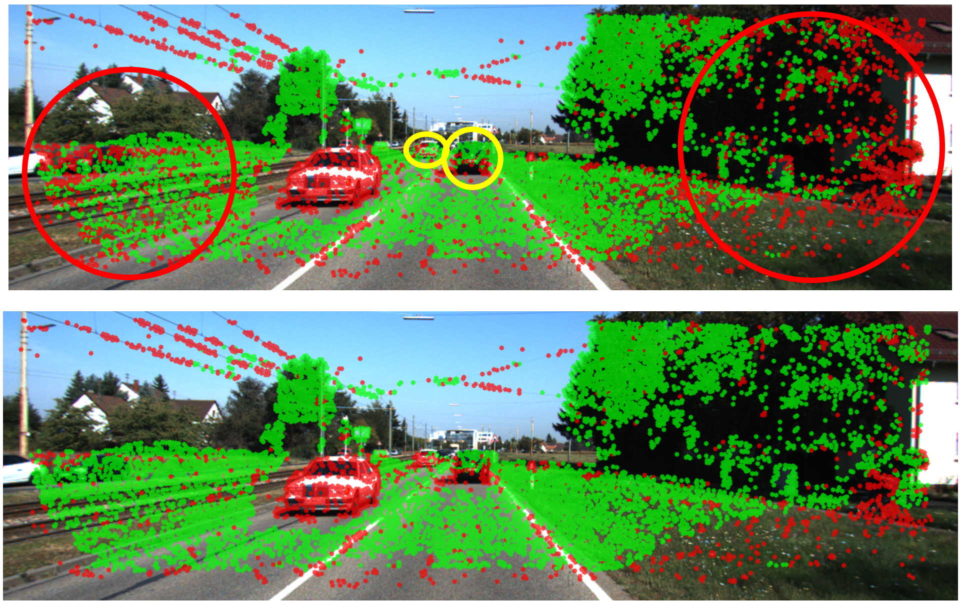


Figure 6.3: Demonstration of ego-motion outlier redefinition. Green colour marks ego-motion inliers and red colour ego-motion outliers. (top) original approach and (bottom) proposed approach. The most significant difference is on the sides of the images, where lots of false-positive ego-motion outliers disappeared. Red ellipses mark areas with significant number of false-positives ego-motion outliers and yellow ellipses mark false-negative ego-motion outliers.

estimated disparity. A small error at disparity leads to a significant error of estimated model.

To alleviate these problems we decided to use LO-RANSAC [CMK03] to generate motion hypotheses. Because of its local optimisation step it tends to produce more precise motion hypotheses. As we will demonstrate it in Section 7, the number of missing cars is the lowest from all tested combinations. The variance of the algorithm is also reduced.

6.4 Temporally Consistent Superpixels

We assume that superpixels formed on the same surfaces in the scene but observed from different viewpoints or time steps should not significantly differ in appearance or shape. We decided to apply temporal consistency on another level and use time-consistent superpixels to initialise the OSF algorithm.

We decided to use Chang et al. [CWF13] algorithm for temporal superpixels (TSP). Their algorithm is based on the probabilistic model and previous seminal work from simple linear iterative clustering (SLIC) [ASS$^+$12]. They explicitly model the optical flow using bilateral Gaussian process. Instead of that, we use optical flow estimated by OSF algorithm.

In a situation that we have not estimated optical flow available (first frame in the sequence), we initialise with disparity estimated by SGM [Hir05]. Figure 6.5 shows the superpixel segmentation of TSP with compare to SPS-St.

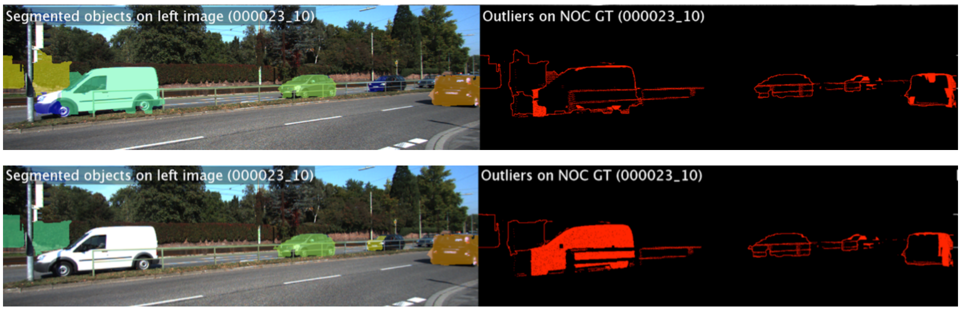

Segmented objects on left image (000023_10)
Outliers on NOC GT (000023_10)
Segmented objects on left image (000023_10)
Outliers on NOC GT (000023_10)

Figure 6.4: Demonstration of OSF variance. (top) shows the best result and (bottom) shows the worst result on a random KITTI'15 sequence with ten randomly initialised computations. Left images show the final labels of independently moving objects (background not shown) and right images represent EPE of the found optical flow (red colour for EPE $\geq$ 3px).

Consequently, we proposed two versions of modification for robust disparity estimation:

- The estimated initial disparity [Hir05] is fitted to all pixels assigned with given superpixel to find a 3D plane describing the shape of the superpixel. Fitting is provided for each superpixel separately by the RANSAC scheme [FB81]. The version is labelled as SGM+TSP in the following text.

- The superpixel shape and appearance estimated by TSP is used only for the default initialisation for the SPS-St[YMU14] algorithm. The rest of the SPS-St algorithm is kept untouched. The version is labelled as SGM+TSP+SPS-St in the following text.

The quantitative and qualitative results of both versions are presented in Section 7.3.

6.5 Optimisation Space Reduction

Speed-up analysis of the OSF (see Sec. 5.3) shows that significant time consumption of the algorithm belongs to optimisation part (98.5% with 50 iteration setting).

As was mentioned in Chapter 5, there are three obvious possibilities to save time consumption of optimisation. The first is the reduction of an optimisation iterations total number. This modification was presented in the original paper. The second possible modification is GPU parallelisation, where theoretical acceleration is up to ten times, as indicated by the [FM08, AMY09]. Due to a complexity of GPU parallelisation algorithm development, it is kept as the future work. The third modification is a searching space of possible solutions reduction, on which we are focused in this section.

We are using an assumption that initially estimated ego-motion and disparity on the background are provide enough accuracy over areas without

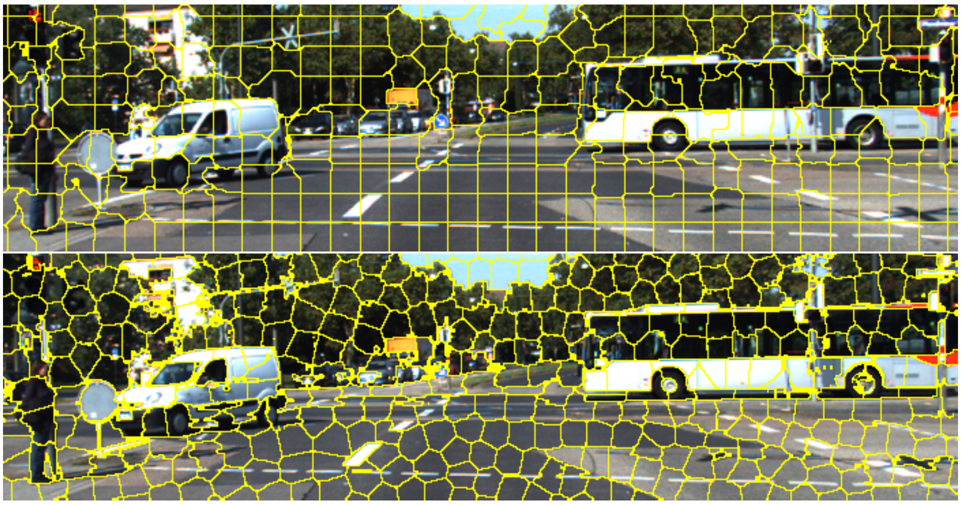


Figure 6.5: Qualitative comparison of superpixel segmentation algorithms on the KITTI'15 dataset [MG15]. The top image is segmentation using SPS-St [YMU14], and the bottom image is segmentation using temporal superpixel representation by [CWF13]. Images show that [CWF13] builds complicated shapes of superpixels but can adjust to the shape of the individual objects.

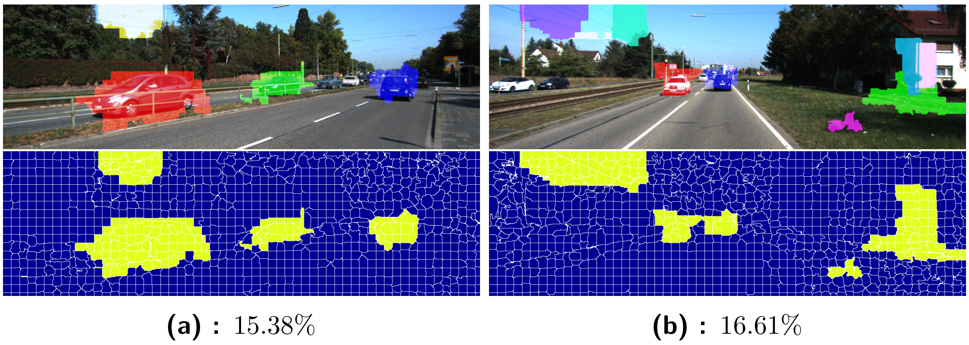
independently moving objects. We build our assumption on the optical flow and disparity benchmark results [GLSU12, MG15] for static scenes, where algorithms for ego-motion scene flow [YMU13, YMU14] reach state-of-the-art results.

Thus, we decided not to optimise the shape and motion for superpixels that have ego-motion as only one possible hypothesis. Instead of that superpixels keep disparity and motion estimated in algorithm initialisation. According to that, the default CRF for OSF is modified as follows:

$$\mathbf{E}\left(\mathbf{s},\mathbf{o}\right)=\sum_{i\in S^{*}}\mathbf{E}_{\mathrm{D}}^{i}\left(\mathbf{s}_{i},\mathbf{o}\right)+\lambda\sum_{i\in S^{*}}\sum_{j\in S^{*}}[\![i\neq j]\!]\,\mathbf{E}_{\mathrm{S}}^{i}\left(\mathbf{s}_{i},\mathbf{s}_{j}\right), \tag{6.3}$$

and $\{S^{*}\subseteq S \mid i\in S^{*}, \|\mathscr{H}_i\|\geq 2\}$, where $\|\mathscr{H}_i\|$ is number of independent motion hypotheses in superpixel i.

Figure 6.6 shows a percentage of the image with superpixels ego-motion hypothesis only. We assume that the theoretical acceleration of the algorithm will be proportional to the area where no optimisation is applied and total image area. We evaluate the results of this modification in Chapter 7, where we focus not only on the algorithm acceleration analysis but also on the impacts that this change affected on the accuracy of estimated scene flow, as well as the impact on false positive and false negative rates of moving objects.


(a) : 15.38%

(b) : 16.61%

Figure 6.6: An example of the superpixels contains independent motion hypotheses on the KITTI'15 dataset [MG15]. Upper images show motion hypotheses after non-maxima suppression in the reference frame. Lower images show areas assigned with hypotheses different from ego-motion and their neighbouring superpixels (yellow) and areas with ego-motion hypothesis only (blue). Under image is a percentage of area with are used in the optimisation excluding ego-motion only areas.

Chapter 7

Experiments

In this chapter, we experimentally evaluate of the proposed extensions. First, we show analysis of variance analysis of the original algorithm. Then we focus on the proposed modifications evaluation. For evaluation, we are using the standard KITTI'15 benchmark [MG15]. The benchmark contains stereo camera sequences with large displacements and nontrivial environment conditions.

We evaluate the precision of the estimated disparity, optical flow and scene flow but we also evaluate the quality of moving objects detection with false positive and false negative rates. Finally, we compare our proposed extensions with state-of-the-art methods for scene flow estimation.

7.1 Evaluation protocol

To evaluate the scene flow, optical flow and disparity we use the standard KITTI'15 metric – a percentage of erroneous pixels. Pixels are considered erroneous when the end-point-error exceeds 3 pixels.

Since the OSF does not compute only scene flow, but returns also segmentation of the scene into independent moving objects, we also report the number of missed moving vehicles – false negatives (FN), and the number of falsely detected vehicles – false positives (FP) as annotated in the data. We label object O_k as true positive if

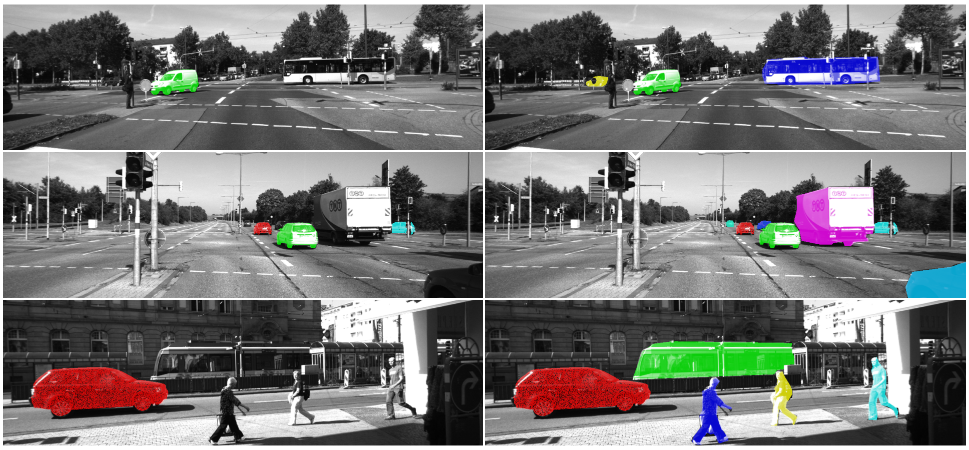
$$\frac{|L^{\mathrm{GT}_m} \cap L^{O_k}|}{|L^{\mathrm{GT}_m} \cup L^{O_k}|} \geq 0.5, \tag{7.1}$$

where L^{GT_m} is the set of pixels of the mth moving vehicle marked in the ground truth and L^{O_k} is a set of pixels labelled as k-th object by the proposed algorithm. This evaluation is complementary to the scene flow quality measure, and we believe that it better reflects the ultimate goal of all these methods - dynamic scene understanding.

Video sequences provided by KITTI'15 dataset contain different types of independently moving objects as pedestrians, cyclists, passenger cars, vans, trucks, trains and buses. However, ground truth labels are available only for some moving cars and vans (see Sec. 4.1). However, we observe that the OSF is able to detect moving objects in the scene no matter object class. Thus, we extend the label annotation of another 148 objects to total number 577. We also distinguish classes among individual objects. Figure 7.1 shows some

	$\overline{\text{FP}}$	σ_{FP}	$\overline{\text{FN}}$	σ_{FN}
OSF [MG15]	236.6	14.7	170.4	3.3
+ label propagation	276.1	11.7	153.5	8.4
+ dynamic outliers	**219.6**	8.3	142.7	3.0
+ LO-RANSAC (3 frames)	244.4	12.4	125.3	2.9
+ LO-RANSAC (5 frames)	235.3	9.3	**121.0**	3.6
+ LO-RANSAC (12 frames)	236.3	17.0	123.2	5.4

Table 7.1: Comparison of detection results of moving vehicles. Tested on the KITTI'15 training multiview dataset. We run listed algorithms algorithm 5 times for each sequence and each extension. $\overline{\text{FP}}$ and $\overline{\text{FN}}$ denote mean of false positive (wrong detection) and false negative (missed detection) respectively. In addition the standard deviations σ_{FP} and σ_{FN} are shown for better comparison. Total number of vehicles is 429.



(a) : Original annotation

(b) : Extended annotation

Figure 7.1: Comparison of original and extended annotation on KITTI'15 dataset for moving objects labels.

instances of extended annotation and specific number of objects among all categories are listed in Table 7.4.

7.2 Object Scene Flow Variance Analysis

As was mentioned above, we noticed that the OSF results vary significantly depending on the random seed initialisation. To investigate this variance, we removed all fixed random generator seeds in all parts of the OSF algorithm and instead initialised all the seeds randomly for each computation. We then run the OSF algorithm 30 times for each sequence. Figure 7.2 shows variance of the OSF results. We noticed that sequences with large variance have difficult radiometric conditions or large displacements.

To determine the cause of high variance more accurately, we performed the following experiment. We ran the OSF algorithm and stored part results after

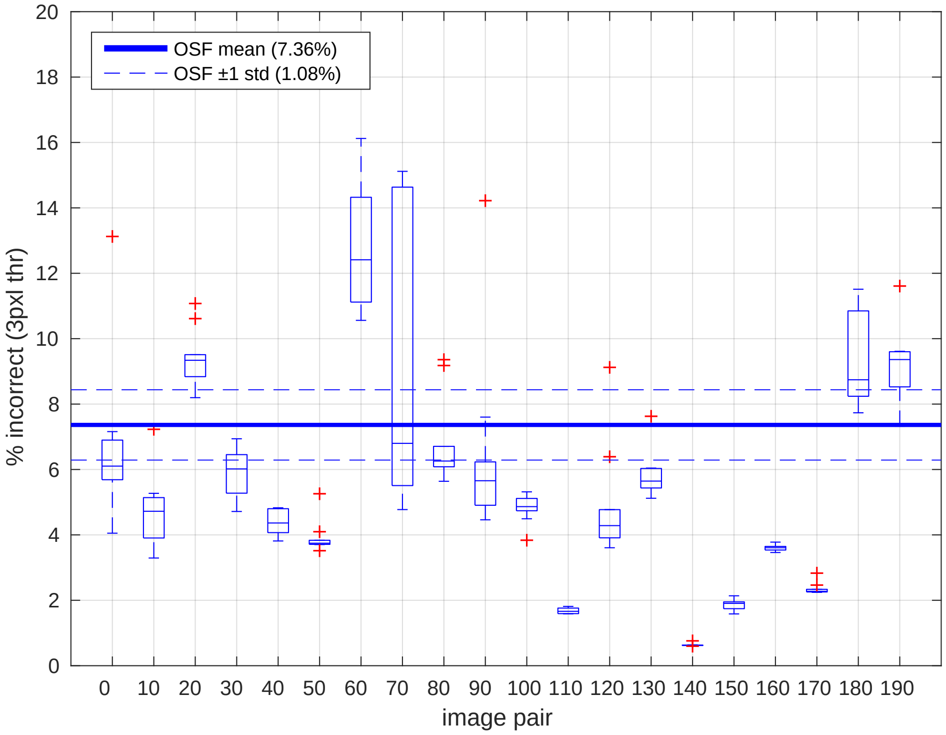


Figure 7.2: Variance of the OSF algorithm (variance bar graphs only for every 10th, mean and variance over all 200 sequences). The central line inside each box indicates the median. The bottom of the box refers the 25th percentile and the top of the box refers to the 75th percentiles, respectively. Outliers are marked with the red symbol 'x'.

every step that is dependent on a random numbers generator. Subsequently, we proceeded from these part results, we restored calculations and continued the algorithm computation (ten times for each part result). We found that the measured data contain a substantial decrease in the variance after estimation of the independent motion hypotheses step. Graph 7.3 shows the experiment for selected examples.

We identified the estimation of motion hypotheses as a critical step of the algorithm. We assume that temporal consistency of independently moving objects should add more stability to the algorithm. Our other two proposed modifications also contribute to the step of motion hypotheses estimation.

7.3 Evaluation of the Proposed Object Scene Flow Extensions

We compare results of our modifications according to various quantitative criteria as erroneous pixels percentage of scene flow, optical flow and disparity for different scene flow estimation variants.

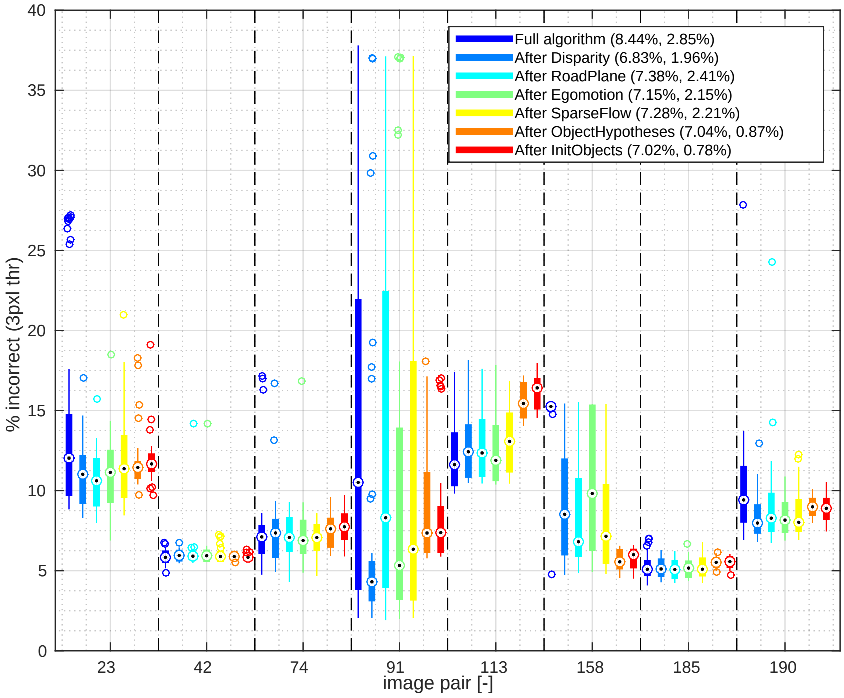


Figure 7.3: Optical flow variance of individual parts of OSF algorithm. The graph shows ten runs of the algorithm on chosen examples. Each stochastic part of the algorithm was precomputed and then was computation initialised with this precomputed values in each stochastic step. Numbers in the legend of the graph represent mean and standard deviance respectively over whole KITTI'15 training dataset. The central point inside each box indicates the median. The bottom of the box refers the 25th percentile and the top of the box refers to the 75th percentiles, respectively. Outliers are marked with the symbol ○.

Object motion label propagation. Object motion label propagation is a modification of OSF algorithm which is focused on an addition of temporal consistency to the algorithm (see Sec. 6.1). The modification influence is shown in Table 7.1. The number of undetected vehicles decreases. On the other hand, the number of false positive detections increases. Besides that, we observe also a slight increase of scene flow error as shown in Table 7.2. As discussed above, this is connected with the propagation of false positives in time. Independently moving objects are stable during longer sequences. Thanks to labels propagation we also preserve objects IDs. Figure 7.6 shows the comparison between original algorithm and proposed modification.

Ego-motion outlier redefinition. Next, we evaluate the ego-motion outlier re-definition (Sec. 6.2). It helps to decrease the number of false positive detections as shown in Table 7.1. Also the number of false detections decreases. The scene flow is still worse than the original OSF but gets slightly better

Alg.\Error	D1-bg	D1-fg	D1-all	D2-bg	D2-fg	D2-all
OSF [MG15]	3.77 %	7.30 %	4.45 %	4.22 %	12.41 %	5.60 %
+ label propagation	3.85 %	6.78 %	4.52 %	4.38 %	10.34 %	5.65 %
+ dynamic outliers	**3.74** %	6.50 %	4.39 %	4.37 %	9.57 %	5.56 %
+ LO-RANSAC (3 frames)	**3.78** %	6.11 %	4.33 %	4.41 %	8.49 %	**5.30** %
+ LO-RANSAC (5 frames)	3.81 %	6.33 %	4.42 %	4.43 %	9.04 %	5.51 %
+ LO-RANSAC (12 frames)	3.81 %	6.28 %	4.42 %	4.44 %	8.76 %	5.47 %
OSF+TC+SGM+TSP	3.56 %	7.40 %	4.96 %	4.20 %	8.31 %	5.68 %
OSF+TC+SGM+TSP+SPS-St	3.53 %	7.34 %	4.90 %	**4.13** %	9.13 %	5.83 %
OSF-BG	3.79 %	5.07 %	4.21 %	4.46 %	9.48 %	5.48 %
OSF+TC-BG	3.80 %	**4.79** %	**4.19** %	4.56 %	**7.63** %	5.38 %

Alg.\Error	Fl-bg	Fl-fg	Fl-all	SF-bg	SF-fg	SF-all
OSF [MG15]	**4.45** %	20.36 %	7.04 %	**5.48** %	22.95 %	8.20 %
+ label propagation	4.68 %	18.90 %	7.44 %	5.73 %	21.41 %	8.51 %
+ dynamic outliers	4.75 %	17.77 %	7.34 %	5.78 %	20.43 %	8.45 %
+ LO-RANSAC (3 frames)	4.79 %	12.74 %	**6.37** %	5.82 %	**15.21** %	**7.52** %
+ LO-RANSAC (5 frames)	4.81 %	14.05 %	6.79 %	5.86 %	16.54 %	7.91 %
+ LO-RANSAC (12 frames)	4.78 %	14.01 %	6.80 %	5.82 %	16.52 %	7.92 %
OSF+TC+SGM+TSP	4.76 %	**11.70** %	6.84 %	5.59 %	22.77 %	8.29 %
OSF+TC+SGM+TSP+SPS-St	4.70 %	12.58 %	7.08 %	5.53 %	24.43 %	8.56 %
OSF-BG (orig, 2 frames)	5.72 %	17.18 %	7.69 %	6.74 %	18.86 %	8.78 %
OSF+TC-BG	6.01 %	13.78 %	7.56 %	7.04 %	15.54 %	8.64 %

Table 7.2: Scene flow evaluation of proposed modifications. Tested on the KITTI'15 training multiview dataset. Columns marks categories of evaluation: scene flow as **SF**, optical flow as **F**, disparity **D1** for the first frame of the test pair and **D2** for the second; and subcategories: evaluation over **bg** background regions, evaluation over **fg** foreground regions and **all** evaluation over all ground-truth. Categories where proposed modifications performs better than the original OSF are highlighted in grey and the best results are in bold.

results compared to the previous experiment (see Table 7.2).

Robust motion hypotheses. The additional application of LO-RANSAC (see Sec. 6.3) in the motion hypotheses estimation leads to a significant decrease of the scene flow error from 8.2% to 7.52% (Tab. 7.2) compared to the original OSF algorithm. Besides, the number of false negatives also decreases. Only the number of the false positives increases slightly (Tab. 7.1). For the case of application of all extensions, we try a different number of frames as input to the temporally consistent OSF.

We run experiments in for 3, 5 and 12 frames. Temporally consistent moving objects are propagated from the previous frame to current frame in an online manner. Using more frames from the past reduces FPs and FNs as shown in Table 7.1, however the scene flow results degrade a bit (Table 7.2). This effect is most likely caused by the different density of ground-truth in the KITTI dataset (foreground is about 4× denser – sparse data from LiDAR and extrapolated data with using of CAD models). Every super-pixel falling on both foreground and background is more likely to be removed from the motion hypothesis when propagated longer. This nibbling of the car borders, however, causes higher foreground scene-flow errors as shown in Table 7.2.

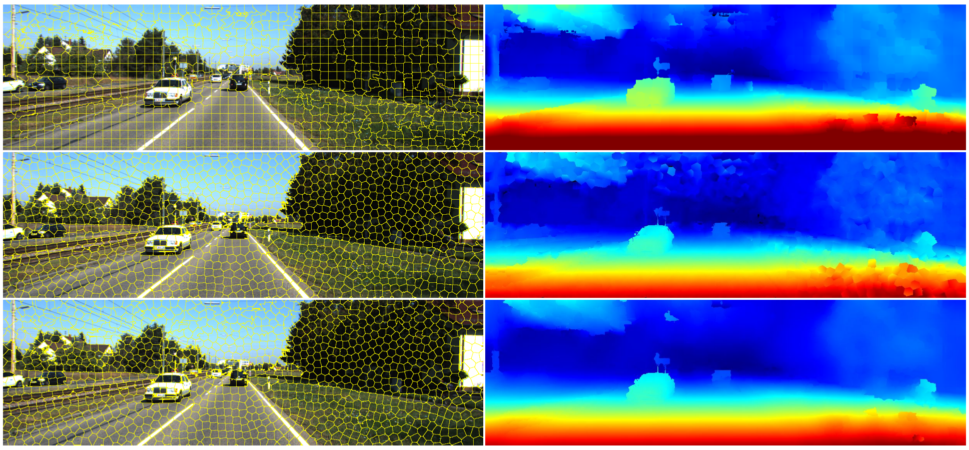


Figure 7.4: Qualitative comparison of superpixel segmentation on KITTI'15 dataset. The top row shows an example of segmentation and initial disparity estimation from SPS-St used in the OSF. The middle row shows an example of TCS and robust estimation of disparity using RANSAC over segment assigned pixels initialised by SGM. The bottom row shows segmentation and disparity estimation produced by the combination of TCS and SPS-St, where TCS segmentation is used as initialisation of SPS-St superpixel shapes.

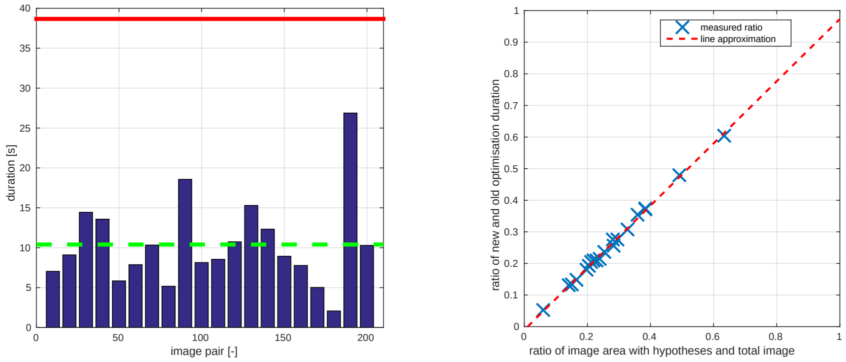
Application of temporally consistent superpixels. The proposed modification of temporal consistency addition on another level is described in Section 6.4. Figure 7.4 shows qualitative comparison between temporal consistent superpixels application (Sec 6.4) and original OSF. Evaluation of disparity, optical flow and scene flow for both proposed scenarios is shown in Table 7.2. Although the qualitative comparison of superpixel shape is better for methods using temporally consistent superpixels. The original usage of SPS-St [Hir05] achieves better results in quantitative comparison for scene flow estimation. Proposed modifications using temporal superpixels achieve the best results for foreground optical flow estimation, but modifications have worse results in the scene flow category than other proposed modifications or their combinations.

From the reason of worse scene flow estimation results, we decided not use this modification with the state-of-the-art comparison[1]. The another modifications using temporally consistent superpixels are kept as the future work.

Reducing time consumption. Measurement of time consumption modification is depicted in Figure 7.5. Average speed-up of algorithm optimisation is 3.82x, which results in a reduction of whole algorithm duration from 32 minutes and 40 seconds to 8 minutes and 52 seconds[2]. Further, we experimentally prove an assumption that speed up linearly depends on the area occupied by the motion hypotheses different from ego-motion. Figure 7.5

[1] KITTI'15 benchmark allows only one submission for one algorithm to reduce the overfitting

[2] single core Intel i5-2.4GHz



(a) : Absolute speed up.

(b) : Relative speed up.

Figure 7.5: Influence of search space reduction on time of the optimisation part of the OSF algorithm. The left chart shows the optimisation speed-up on each 10th image from the KITTI'15 dataset, where the red line is measured mean of original OSF and green line is mean of measured modifications over the whole KITTI'15 training dataset. The right graph shows that the speed-up is approximately linear in the percentage of the area occupied by the motion hypotheses.

shows their approximately linear dependency. Table 7.3 shows an evaluation of false positive and false negative detection rates for this modification. Table 7.2 shows the evaluation of estimated disparity, optical flow and scene flow for the proposed speed-up marked as OSF-BG (for application directly on the original OSF) and OSF+TC-BG (for application on OSF with other extensions applied). Results show that application of search space reduction has a positive effect on false negative moving objects detection (OSF+TC-BG the best result from all tested modifications) with a comparison of methods without space reduction, but at the same time, it increases the number of false positive detection.

The speed-up of the algorithm with accuracy decreasing effect is a problem for the most of the estimation algorithms. For now, we have no correction for an algorithm accuracy with the usage of optimisation space reduction modification. We can choose between slow and more accurate or quicker and less accurate options.

Space reduction also has a positive effect on disparity, optical flow and scene flow estimation over foreground areas. However, with a comparison to scene flow and optical flow is worse across all flow categories than proposed OSF+TC method.

7.3.1 Summary

We also evaluate some of the modifications on the extended dataset of moving objects (see Table 7.4) which was mentioned above. We do not observe big differences for modifications among object categories, except moving cars, where the modifications using temporal consistency are significantly better.

	$\overline{\text{FP}}$	$\overline{\text{FN}}$	$\overline{\text{FP}}^*$	$\overline{\text{FN}}^*$
OSF [MG15]	**236.6**	**170.4**	220.3	**292.7**
OSF+TC	244.4	125.3	**210.7**	233.3
OSF-BG	298.7	150.4	267.1	266.7
OSF+TC-BG	**333.4**	**120.3**	**291.3**	**226.0**

Table 7.3: Comparison of detection results of moving vehicles. Tested on the KITTI'15 training multiview dataset. We run listed algorithms algorithm 5 time for each sequence and each extension. $\overline{\text{FP}}$ and $\overline{\text{FN}}$ denote mean of false positive (wrong detection) and mean of false negative (missed detection) respectively. In addition the standard deviations σ_{FP} and σ_{FN} are shown for better comparison. $\overline{\text{FP}}^*$ and $\overline{\text{FN}}^*$ denote false positive and false negative on spread dataset of moving objects labels. The best case in the category is coloured in green, and the worse case is coloured in red.

	$\overline{\text{FN}}^*$					
	Car	Van	Cyclist	Pedestrian	Truck	Train
OSF [MG15]	**238.7**	**15.0**	2.0	11.7	17.3	**5.0**
OSF+TC	182.7	13.3	2.0	**12.7**	15.3	**4.3**
OSF-BG	214.0	13.7	2.3	**11.0**	**18.0**	4.7
OSF+TC-BG	**177.0**	**12.0**	2.3	12.0	**15.0**	4.7
Total number	482	39	8	18	25	5

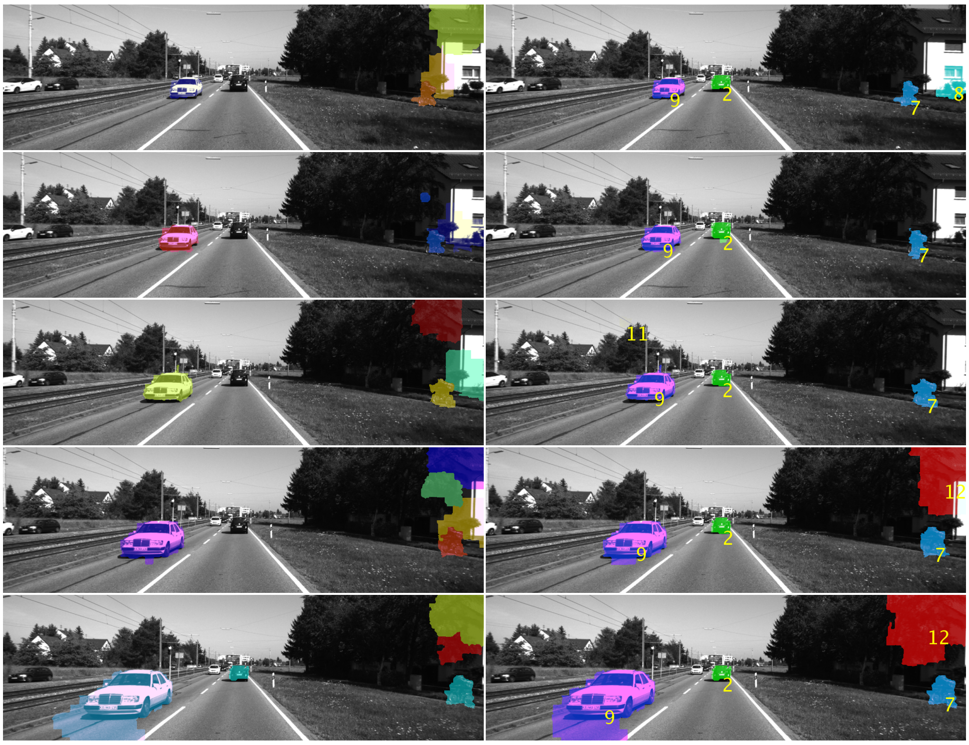
Table 7.4: Comparison of detection results of moving objects. Tested on the KITTI'15 training multiview dataset. We run listed algorithms algorithm 5 times for each sequence and each extension. $\overline{\text{FN}}^*$ denote mean of false negative (missed detection) for individual class of moving objects labels on extended dataset. The best case in the category is coloured in green, and the worse case is coloured in red.

The method is termed OSF+TC in the comparisons.

Based on the results, the propagation through three frames was chosen for further comparison with the state of the art. The level of false positives and false negatives is similar to other variants, but the scene flow errors are significantly lower.

7.4 Comparison with the State of the Art

We compare the best combination of all proposed extensions (temporal consistency using three stereoscopic frames, ego-motion outlier redefinition and robust motion hypotheses generation) with the best-ranked KITTI'15 submissions in the scene flow category. Table 7.5 shows the results for evaluation on all pixels from ground-truth in the image frame. OSF+TC decreases EPE of the original OSF from 10.63% to 9.65% and achieves the second position in the scene flow estimation total. The loss to the first place (PRSM [VSR15]) is less than a quarter percent. Moreover, OSF+TC ranked first for scene flow evaluation on non-occluded pixels as shown in Table 7.6,



(a) : Original **(b) :** Proposed

Figure 7.6: Propagation of moving object label through time. Three moving objects (id=2,7,9) in (b) have stable label over the whole sequence as opposed to the original approach in (a). Car 2 is detected earlier due to the stronger LO-RANSAC model estimation and is then correctly propagated. Object 7 is a man on a bicycle. Also many false positives are reduce due to the ego-motion outlier redefinition.

with a modification to the original algorithm by 1%. Finally, OSF+TC achieves the first position for scene flow and optical flow over foreground regions for both, non-occluded pixel and all pixel evaluation.

A complete comparison with other KITTI'15 competitors is listed in Appendix B. Recently tested but unpublished methods are also listed for fair comparison.

Alg.\Error	**D1-bg**	**D1-fg**	**D1-all**	**D2-bg**	**D2-fg**	**D2-all**
PRSF [VSR13]	4.74 %	13.74 %	6.24 %	11.14 %	20.47 %	12.69 %
CSF [LBA$^+$16]	4.57 %	13.04 %	5.98 %	7.92 %	20.76 %	10.06 %
OSF [MG15]	4.54 %	12.03 %	5.79 %	5.45 %	19.41 %	7.77 %
PRSM [VSR15]	**3.02** %	10.52 %	**4.27** %	**5.13** %	**15.11** %	**6.79** %
OSF+TC (ours)	4.11 %	**9.64** %	5.03 %	5.18 %	15.12 %	6.84 %

Alg.\Error	**Fl-bg**	**Fl-fg**	**Fl-all**	**SF-bg**	**SF-fg**	**SF-all**
PRSF [VSR13]	11.73 %	27.73 %	14.39 %	13.49 %	33.72 %	16.85 %
CSF [LBA$^+$16]	10.40 %	30.33 %	13.71 %	12.21 %	36.97 %	16.33 %
OSF [MG15]	5.62 %	22.17 %	8.37 %	7.01 %	28.76 %	10.63 %
PRSM [VSR15]	**5.33** %	17.02 %	**7.28** %	**6.61** %	23.60 %	**9.44** %
OSF+TC (ours)	5.76 %	**16.61** %	7.57 %	7.08 %	**22.55** %	9.65 %

Table 7.5: Quantitative comparison with the state-of-the-art results (**all pixels**). Columns mark categories of evaluation: scene flow as **SF**, optical flow as **F**, disparity **D1** for the first frame of the test pair and **D2** for the second; and subcategories: evaluation over **bg** background regions, evaluation over **fg** foreground regions and **all** evaluation over all ground-truth. Categories where OSF+TC performs better than the original OSF are highlighted in grey and the best results are in bold.

Alg.\Error	**D1-bg**	**D1-fg**	**D1-all**	**D2-bg**	**D2-fg**	**D2-all**
PRSF [VSR13]	4.41 %	13.09 %	5.84 %	6.35 %	16.12 %	8.10 %
CSF [LBA$^+$16]	4.03 %	11.82 %	5.32 %	6.39 %	16.75 %	8.25 %
OSF [MG15]	4.14 %	11.12 %	5.29 %	4.49 %	16.33 %	6.61 %
PRSM [VSR15]	**2.93** %	10.00 %	**4.10** %	**4.13** %	12.85 %	5.69 %
OSF+TC (ours)	3.79 %	**8.66** %	4.59 %	4.18 %	**12.06** %	**5.59** %

Alg.\Error	**Fl-bg**	**Fl-fg**	**Fl-all**	**SF-bg**	**SF-fg**	**SF-all**
PRSF [VSR13]	6.94 %	23.64 %	9.97 %	8.35 %	28.45 %	11.95 %
CSF [LBA$^+$16]	8.72 %	26.98 %	12.03 %	10.26 %	32.58 %	14.26 %
OSF [MG15]	**4.21** %	18.65 %	6.83 %	**5.52** %	24.58 %	8.93 %
PRSM [VSR15]	4.33 %	14.15 %	6.11 %	5.54 %	20.16 %	8.16 %
OSF+TC (ours)	4.34 %	**12.86** %	**5.89** %	**5.52** %	**18.02** %	**7.76** %

Table 7.6: Quantitative comparison with state-of-the-art results (**non-occluded pixels**). Columns marks categories of evaluation: scene flow as **SF**, optical flow as **F**, disparity **D1** for the first frame of the test pair and **D2** for the second; and subcategories: evaluation over **bg** background regions, evaluation over **fg** foreground regions and **all** evaluation over all ground-truth. Categories where OSF+TC performs better than the original OSF are highlighted in grey and the best results are in bold.

Chapter 8

Future Work

Since we are particularly focused on automotive industry usage of the algorithm, we want to achieve real-time scene flow computation. For this reason, we want to reimplement parts of the algorithm to GPU and achieve computation speed-up using parallelisation.

We plan to use a better way of initialisation object motion candidates by an algorithm for multi-class and multi-instances model fitting or use such an algorithm directly for object motion labelling, instead of extremely time-consuming discrete-continuous CRF optimisation, which is presented in the original algorithm. Multi-class object fitting is considered be helpful for distinguishing between big independently moving objects, like trucks or trains, and small moving objects, as pedestrians. Both categories cause mismatches in the current version of the algorithm.

We also plan cast temporary consistent superpixels (e.g. [CWF13, RJRO13]) in another way than is proposed in this thesis, to achieve temporal consistency of labels and scene flow over much longer sequences. This modification is considered be useful especially for objects close to *focus of expansion* point where motion parallax over longer sequence could help to distinguish moving objects from the background.

Chapter 9
Conclusion

The first part of the thesis shows the basic concept of scene flow estimation with a focus on several possible estimation difficulties. Then, we review related works and briefly survey scene flow datasets associated with autonomous driving. Next, The Object Scene Flow algorithm [MG15] and its individual parts are described in details. Also, time consumption analysis of the algorithm parts is performed. Finally, we introduce severe modifications to the OSF algorithm. The proposed modifications are evaluated on the KITTI'15 benchmark, and their limitations are identified.

In particular, we proposed a modification adding temporal consistency to the OSF algorithm. It uses independent motion segmentation and motion parameters estimated in previous frames to achieve better initialisation of the scene flow in the current frame. We use the assumption that the motion of an individual object is almost constant between consequent frames. Moving objects segmentation is more stable over longer sequences using this modification.

Then, we proposed the method for more precise distinguish between static background and independently moving objects. It takes an advantage that the optical flow is not the same over the whole image area but varies with the distance from the focus of expansion. It results in less false positive detections in the regions where the optical flow is big. Also, the modification reduces the number of missed detections of independently moving objects near the focus on expansion.

We also proposed a modification of hypotheses generator for independent motions. We changed the standard RANSAC algorithm for the locally optimised RANSAC to achieve a more robust estimation of motions. The main achievement of this method is the reduction of scene flow error.

Next, we proposed two variants of the modification using temporal consistency on the superpixel segmentation level. Both variants combine initial disparity estimation, superpixel segmentation and optical flow between the previous and current frame. However, the success of this modifications is only partial. Thus, the further extension of the algorithm based on temporary consistent superpixel segmentation is kept as a future work.

Furthermore, we introduced a speed-up of the algorithm's optimisation part. We reduced the search space of possible solutions removing optimisation of superpixel with ego-motion hypothesis only. This modification brought speed-up of the algorithm more than three times in average. We also experimentally verified that speed-up is approximately linearly depended on the proportion of the area with independent motion hypotheses and the whole image area.

We experimentally evaluated all proposed modifications of the OSF algorithm on the KITTI'15 testing dataset. We focused on the error of estimated scene flow, optical flow and disparity, as well as on the detection of independently moving objects. We also extended the annotation of moving objects with additional segmentation of bicyclist, pedestrians, trams, trucks and distant vehicles (the original annotation contains only passenger cars and vans close to the camera). We tested our modifications also on this extended dataset. The number of false negative detections was reduced by more than 35% using proposed modifications.

The combination of the proposed modifications (object motion label propagation, ego-motion outlier redefinition and robust motion hypotheses generation) was evaluated in the competitive KITTI'15 scene flow benchmark as **OSF+TC**. We achieved **absolute first place** in the scene flow estimation category over non-occluded pixels and **second place** in the scene flow estimation over the whole image area. Further, we achieve **first places** for optical and scene flow **foreground** estimation for both non-occluded pixels and whole image area.

Bibliography

[AMY09] Shuichi Asano, Tsutomu Maruyama, and Yoshiki Yamaguchi. Performance comparison of fpga, gpu and cpu in image processing. In *2009 International Conference on Field Programmable Logic and Applications*, pages 126–131, Aug 2009.

[ASS$^+$12] Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurelien Lucchi, Pascal Fua, and Sabine Süsstrunk. Slic superpixels compared to state-of-the-art superpixel methods. *IEEE transactions on pattern analysis and machine intelligence*, 34(11):2274–2282, 2012.

[BHF$^+$10] Christian Banz, Sebastian Hesselbarth, Holger Flatt, Holger Blume, and Peter Pirsch. Real-time stereo vision system using semi-global matching disparity estimation: Architecture and fpga-implementation. In *Embedded Computer Systems (SAMOS), 2010 International Conference on*, pages 93–101. IEEE, 2010.

[BM11] Thomas Brox and Jitendra Malik. Large displacement optical flow: Descriptor matching in variational motion estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(3):500–513, March 2011.

[BMK13] Tali Basha, Yael Moses, and Nahum Kiryati. Multi-view scene flow estimation: A view centered variational approach. *International journal of computer vision*, 101(1):6–21, 2013.

[BSL$^+$11] Simon Baker, Daniel Scharstein, J. P. Lewis, Stefan Roth, Michael J. Black, and Richard Szeliski. A database and evaluation methodology for optical flow. *International Journal of Computer Vision*, 92(1):1–31, 2011.

[BT99] Stan Birchfield and Carlo Tomasi. Multiway cut for stereo and motion with slanted surfaces. In *Proceedings of the Seventh IEEE International Conference on Computer Vision*, volume 1, pages 489–495 vol.1, 1999.

[BW16] Martin Buczko and Volker Willert. Flow-decoupled normalized reprojection error for visual odometry. In *Intelligent Transportation Systems (ITSC), 2016 IEEE 19th International Conference on*, pages 1161–1167. IEEE, 2016.

[BWSB12] Daniel J Butler, Jonas Wulff, Garrett B Stanley, and Michael J Black. A naturalistic open source movie for optical flow evaluation. In A. Fitzgibbon et al. (Eds.), editor, *European Conf. on Computer Vision (ECCV)*, Part IV, LNCS 7577, pages 611–625. Springer-Verlag, October 2012.

[CMK03] Ondřej Chum, Jiří Matas, and Josef Kittler. *Locally Optimized RANSAC*, pages 236–243. Springer Berlin Heidelberg, Berlin, Heidelberg, 2003.

[ČSRH11] Jan Čech, Jordi Sanchez-Riera, and Radu P. Horaud. Scene flow estimation by growing correspondence seeds. In *CVPR*, 2011.

[CWF13] Jason Chang, Donglai Wei, and John W Fisher. A video representation using temporal superpixels. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2051–2058, 2013.

[DMG06] Frederic Devernay, Diana Mateus, and Matthieu Guilbert. Multi-camera scene flow by tracking 3-D points and surfels. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 2203–2212, 2006.

[DPSLB16] Maxime Derome, Aurelien Plyer, Martial Sanfourche, and Guy Le Besnerais. A prediction-correction approach for real-time optical flow computation using stereo. In *German Conference on Pattern Recognition*, pages 365–376. Springer, 2016.

[FB81] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[FDI+15] Philipp Fischer, Alexey Dosovitskiy, Eddy Ilg, Philip Häusser, Caner Hazırbaş, Vladimir Golkov, Patrick van der Smagt, Daniel Cremers, and Thomas Brox. Flownet: Learning optical flow with convolutional networks. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 2758–2766, December 2015.

[FKG13] Jannik Fritsch, Tobias Kuehnl, and Andreas Geiger. A new performance measure and evaluation benchmark for road detection algorithms. In *International Conference on Intelligent Transportation Systems (ITSC)*, 2013.

[FM08] James Fung and Steve Mann. Using graphics devices in reverse: Gpu-based image processing and computer vision. In *2008 IEEE International Conference on Multimedia and Expo*, pages 9–12, June 2008.

[GK16] Spyros Gidaris and Nikos Komodakis. Detect, replace, refine: Deep structured prediction for pixel wise labeling. *arXiv preprint arXiv:1612.04770*, 2016.

[GLSU12] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 3354–3361, June 2012.

[GLU13] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013.

[GMAB16] Clément Godard, Oisin Mac Aodha, and Gabriel J Brostow. Unsupervised monocular depth estimation with left-right consistency. *arXiv preprint arXiv:1609.03677*, 2016.

[GR10] Stefan K Gehrig and Clemens Rabe. Real-time semi-global matching on the cpu. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Workshops*, pages 85–92, June 2010.

[GZS11] Andreas Geiger, Julius Ziegler, and Christoph Stiller. Stereoscan: Dense 3d reconstruction in real-time. In *Intelligent Vehicles Symposium (IV), 2011 IEEE*, pages 963–968. IEEE, 2011.

[HD07] Frédéric Huguet and Frédéric Devernay. A variational method for scene flow estimation from stereo sequences. In *2007 IEEE 11th International Conference on Computer Vision*, pages 1–7, October 2007.

[HFR14] Michael Hornacek, Andrew Fitzgibbon, and Carsten Rother. Sphereflow: 6 dof scene flow from rgb-d pairs. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3526–3533. IEEE, 2014.

[Hir05] Heiko Hirschmuller. Accurate and efficient stereo processing by semi-global matching and mutual information. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 2, pages 807–814. IEEE, 2005.

[HS81] Berthold KP Horn and Brian G Schunck. Determining optical flow. *Artificial intelligence*, 17(1-3):185–203, 1981.

[HXJ13] Chun Ho Hung, Li Xu, and Jiaya Jia. Consistent binocular depth and scene flow with chained temporal profiles. *International Journal of Computer Vision*, 102(1):271–292, 2013.

[HZ03] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision.* Cambridge university press, 2003.

[IMS+16] Eddy Ilg, Nikolaus Mayer, Tonmoy Saikia, Margret Keuper, Alexey Dosovitskiy, and Thomas Brox. Flownet 2.0: Evolution of optical flow estimation with deep networks. *CoRR*, abs/1612.01925, 2016.

[Ira02] Michal Irani. Multi-frame correspondence estimation using subspace constraints. *International Journal of Computer Vision*, 48(3):173–194, 2002.

[K+60] Rudolph Emil Kalman et al. A new approach to linear filtering and prediction problems. *Journal of basic Engineering*, 82(1):35–45, 1960.

[KMD+17] Alex Kendall, Hayk Martirosyan, Saumitro Dasgupta, Peter Henry, Ryan Kennedy, Abraham Bachrach, and Adam Bry. End-to-end learning of geometry and context for deep stereo regression. *arXiv preprint arxiv:1703.04309*, 2017.

[KNH+16] Daniel Kondermann, Rahul Nair, Katrin Honauer, Karsten Krispin, Jonas Andrulis, Alexander Brock, Burkhard Gussefeld, Mohsen Rahimimoghaddam, Sabine Hofmann, Claus Brenner, and Bernd Jahne. The hci benchmark suite: Stereo and flow ground truth with uncertainties for urban autonomous driving. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2016.

[Kol06] Vladimir Kolmogorov. Convergent tree-reweighted message passing for energy minimization. *IEEE transactions on pattern analysis and machine intelligence*, 28(10):1568–1583, 2006.

[KSH12] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[LBA+16] Zhaoyang Lv, Chris Beall, Pablo F. Alcantarilla, Fuxin Li, Zsolt Kira, and Frank Dellaert. A continuous optimization approach for efficient and accurate scene flow. *CoRR*, abs/1607.07983, 2016.

[LK+81] Bruce D Lucas, Takeo Kanade, et al. An iterative image registration technique with an application to stereo vision. 1981.

[MB87] David W Murray and Bernard F Buxton. Scene segmentation from visual motion using global optimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-9(2):220–228, March 1987.

[MG15] Moritz Menze and Andreas Geiger. Object scene flow for autonomous vehicles. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[MHG15] Moritz Menze, Christian Heipke, and Andreas Geiger. Joint 3D estimation of vehicles and scene flow. In *ISPRS Workshop on Image Sequence Analysis (ISA)*, 2015.

[MIH$^+$16] Nikolaus Mayer, Eddy Ilg, Philip Hausser, Philipp Fischer, Daniel Cremers, Alexey Dosovitskiy, and Thomas Brox. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. arXiv:1512.02134.

[MP76] David Marr and Tomaso Poggio. Cooperative computation of stereo disparity. In *From the Retina to the Neocortex*, pages 239–243. Springer, 1976.

[NŠ17] Michal Neoral and Jan Šochman. Object scene flow with temporal consistency. In *22nd Computer Vision Winter Workshop (CVWW)*. Pattern Recongition and Image Processing Group, TU Wien & PRIP Club, Vienna, Austria, February 2017. ISBN: 978-3-200-04969-7.

[PZBS14] Jason Pacheco, Silvia Zuffi, Michael J Black, and Erik B Sudderth. Preserving modes and messages via diverse particle selection. In *ICML*, pages 1152–1160, 2014.

[RJRO13] Matthias Reso, Jorn Jachalsky, Bodo Rosenhahn, and Jorn Ostermann. Temporally consistent superpixels. In *The IEEE International Conference on Computer Vision (ICCV)*, December 2013.

[RKVT16] Christian Richardt, Hyeongwoo Kim, Levi Valgaerts, and Christian Theobalt. Dense wide-baseline scene flow from two handheld video cameras. In *3DV*, October 2016.

[RMWF10] Clemens Rabe, Thomas Müller, Andreas Wedel, and Uwe Franke. *Dense, Robust, and Accurate Motion Field Estimation from Stereo Image Sequences in Real-Time*, pages 582–595. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010.

[RWHS15] Jerome Revaud, Philippe Weinzaepfel, Zaid Harchaoui, and Cordelia Schmid. Epicflow: Edge-preserving interpolation of correspondences for optical flow. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1164–1172, 2015.

[SLAR14] Robert Spangenberg, Tobias Langner, Sven Adfeldt, and Raúl Rojas. Large scale semi-global matching on the CPU. In *2014 IEEE Intelligent Vehicles Symposium Proceedings*, pages 195–201, June 2014.

[SRB14] Deqing Sun, Stefan Roth, and Michael J Black. A quantitative analysis of current practices in optical flow estimation and the principles behind them. *International Journal of Computer Vision*, 106(2):115–137, 2014.

[SW16] Amit Shaked and Lior Wolf. Improved stereo matching with constant highway networks and reflective confidence learning. *arXiv preprint arXiv:1701.00165*, 2016.

[TSS17] Tatsunori Taniai, Sudipta N. Sinha, and Yoichi Sato. Fast multi-frame stereo scene flow with motion segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2017)*, 2017.

[UWPB12] Markus Unger, Manuel Werlberger, Thomas Pock, and Horst Bischof. Joint motion estimation and segmentation of complex scenes with label costs and occlusion modeling. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 1878–1885. IEEE, 2012.

[VBR$^+$99] Sundar Vedula, Simon Baker, Peter Rander, Robert Collins, and Takeo Kanade. Three-dimensional scene flow. In *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, volume 2, pages 722–7292, 1999.

[VBVZ11] Sebastian Volz, Andres Bruhn, Levi Valgaerts, and Henning Zimmer. Modeling temporal coherence for optical flow. In *2011 International Conference on Computer Vision*, pages 1116–1123, November 2011.

[VBZ$^+$10] Levi Valgaerts, Andrés Bruhn, Henning Zimmer, Joachim Weickert, Carsten Stoll, and Christian Theobalt. *Joint Estimation of Motion, Structure and Geometry from Stereo Sequences*, pages 568–581. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010.

[VRCK05] Sundar Vedula, Peter Rander, Robert Collins, and Takeo Kanade. Three-dimensional scene flow. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(3):475–480, 2005.

[VRS14] Christoph Vogel, Stefan Roth, and Konrad Schindler. *View-Consistent 3D Scene Flow Estimation over Multiple Frames*, pages 263–278. Springer International Publishing, Cham, 2014.

[VRS$^+$17] Sudheendra Vijayanarasimhan, Susanna Ricco, Cordelia Schmid, Rahul Sukthankar, and Katerina Fragkiadaki. Sfm-net: Learning of structure and motion from video. *arXiv preprint arXiv:1704.07804*, 2017.

[VSR11] Christoph Vogel, Konrad Schindler, and Stefan Roth. 3D scene flow estimation with a rigid motion prior. In *2011 International Conference on Computer Vision*, pages 1291–1298. IEEE, 2011.

[VSR13] Christoph Vogel, Konrad Schindler, and Stefan Roth. Piecewise rigid scene flow. In *The IEEE International Conference on Computer Vision (ICCV)*, December 2013.

[VSR15] Christoph Vogel, Konrad Schindler, and Stefan Roth. 3D scene flow estimation with a piecewise rigid scene model. *International Journal of Computer Vision*, 115(1):1–28, 2015.

[WB15] Jonas Wulff and Michael J Black. Efficient sparse-to-dense optical flow estimation using a learned basis and layers. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 120–130. IEEE, 2015.

[WBV^{+}11] Andreas Wedel, Thomas Brox, Tobi Vaudrey, Clemens Rabe, Uwe Franke, and Daniel Cremers. Stereoscopic scene flow computation for 3D motion understanding. *International Journal of Computer Vision*, 95(1):29–51, 2011.

[WRV^{+}08] Andreas Wedel, Clemens Rabe, Tobi Vaudrey, Thomas Brox, Uwe Franke, and Daniel Cremers. Efficient dense scene flow from sparse or dense stereo data. In *European conference on computer vision*, pages 739–751. Springer, 2008.

[YHMU12] Koichiro Yamaguchi, Tamir Hazan, David McAllester, and Raquel Urtasun. *Continuous Markov Random Fields for Robust Stereo Estimation*, pages 45–58. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.

[YMU13] Koichiro Yamaguchi, David McAllester, and Raquel Urtasun. Robust monocular epipolar flow estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1862–1869, 2013.

[YMU14] Koichiro Yamaguchi, David McAllester, and Raquel Urtasun. Efficient joint segmentation, occlusion labeling, stereo and flow estimation. In *European Conference on Computer Vision*, pages 756–771. Springer, 2014.

[ZBSL17] Tinghui Zhou, Matthew Brown, Noah Snavely, and David G Lowe. Unsupervised learning of depth and ego-motion from video. *arXiv preprint arXiv:1704.07813*, 2017.

Appendix A
DVD content

```
root
 ├── neoral_master_thesis_2017.pdf
 └── code
      ├── directiories_with_codes
      └── demo.m
```

Appendix B

Tables

Method	**D1-bg**	**D1-fg**	**D1-all**	**D2-bg**	**D2-fg**	**D2-all**
ISF (unpublished)	4.12 %	6.17 %	4.46 %	4.88 %	11.34 %	5.95 %
PRSM ([VSR15])	3.02 %	10.52 %	4.27 %	5.13 %	15.11 %	6.79 %
OSF+TC (ours)	4.11 %	9.64 %	5.03 %	5.18 %	15.12 %	6.84 %
SSF (unpublished)	3.55 %	8.75 %	4.42 %	4.94 %	17.48 %	7.02 %
SOSF (unpublished)	4.30 %	8.72 %	5.03 %	5.13 %	15.27 %	6.82 %
OSF ([MG15])	4.54 %	12.03 %	5.79 %	5.45 %	19.41 %	7.77 %
FSF+MS ([TSS17])	5.72 %	11.84 %	6.74 %	7.57 %	21.28 %	9.85 %
CSF ([LBA$^+$16])	4.57 %	13.04 %	5.98 %	7.92 %	20.76 %	10.06 %
PR-Sceneflow ([VSR13])	4.74 %	13.74 %	6.24 %	11.14 %	20.47 %	12.69 %
SGM+SF ([Hir05, HFR14])	5.15 %	15.29 %	6.84 %	14.10 %	23.13 %	15.60 %
PCOF-LDOF ([DPSLB16])	6.31 %	19.24 %	8.46 %	19.09 %	30.54 %	20.99 %
PCOF+ACTF ([DPSLB16])	6.31 %	19.24 %	8.46 %	19.15 %	36.27 %	22.00 %
SGM+C+NL ([Hir05, SRB14])	5.15 %	15.29 %	6.84 %	28.77 %	25.65 %	28.25 %
SGM+LDOF ([Hir05, BM11])	5.15 %	15.29 %	6.84 %	29.58 %	23.48 %	28.56 %
DWBSF ([RKVT16])	19.61 %	22.69 %	20.12 %	35.72 %	28.15 %	34.46 %
GCSF ([ČSRH11])	11.64 %	27.11 %	14.21 %	32.94 %	35.77 %	33.41 %
VSF ([HD07])	27.31 %	21.72 %	26.38 %	59.51 %	44.93 %	57.08 %

Table B.1: KITTI'15 evaluation over all pixels. Columns marks categories of evalution: disparity **D1** for the first frame of the test pair and **D2** for the second; and subcategories: evaluation over **bg** background regions, evaluation over **fg** foreground regions and **all** evaluation over all ground-truth.

Method	**Fl-bg**	**Fl-fg**	**Fl-all**	**SF-bg**	**SF-fg**	**SF-all**
ISF (unpublished)	5.40 %	10.29 %	6.22 %	6.58 %	15.63 %	8.08
PRSM ([VSR15])	5.33 %	13.40 %	6.68 %	6.61 %	20.79 %	8.97
OSF+TC (ours)	5.76 %	13.31 %	7.02 %	7.08 %	20.03 %	9.23
SF (unpublished)	5.63 %	14.71 %	7.14 %	7.18 %	24.58 %	10.07
SOSF (unpublished)	5.42 %	17.24 %	7.39 %	6.95 %	25.78 %	10.08
OSF ([MG15])	5.62 %	18.92 %	7.83 %	7.01 %	26.34 %	10.23
FSF+MS ([TSS17])	8.48 %	25.43 %	11.30 %	11.17 %	33.91 %	14.96
CSF ([LBA$^+$16])	10.40 %	25.78 %	12.96 %	12.21 %	33.21 %	15.71
PR-Sceneflow ([VSR13])	11.73 %	24.33 %	13.83 %	13.49 %	31.22 %	16.44
SGM+SF ([Hir05, HFR14])	20.91 %	25.50 %	21.67 %	23.09 %	34.46 %	24.98
PCOF-LDOF ([DPSLB16])	14.34 %	38.32 %	18.33 %	25.26 %	49.39 %	29.27
PCOF+ACTF ([DPSLB16])	14.89 %	60.15 %	22.43 %	25.77 %	67.75 %	32.76
SGM+C+NL ([Hir05, SRB14])	34.24 %	42.46 %	35.61 %	38.21 %	50.95 %	40.33
SGM+LDOF ([Hir05, BM11])	40.81 %	31.92 %	39.33 %	43.99 %	42.09 %	43.67
DWBSF ([RKVT16])	40.74 %	31.16 %	39.14 %	46.42 %	40.76 %	45.48
GCSF ([ČSRH11])	47.38 %	41.50 %	46.40 %	52.92 %	56.68 %	53.54
VSF ([HD07])	50.06 %	45.40 %	49.28 %	67.69 %	62.93 %	66.90

Table B.2: KITTI'15 evaluation over all pixels. Columns marks categories of evalution: scene flow as **SF** and optical flow as **F**; and subcategories: evaluation over **bg** background regions, evaluation over **fg** foreground regions and **all** evaluation over all ground-truth.

Method	D1-bg	D1-fg	D1-all	D2-bg	D2-fg	D2-all
ISF (unpublished)	3.74 %	5.46 %	4.02 %	4.06 %	9.04 %	4.95 %
OSF+TC (ours)	3.79 %	8.66 %	4.59 %	4.18 %	12.06 %	5.59 %
PRSM ([VSR15])	2.93 %	10.00 %	4.10 %	4.13 %	12.85 %	5.69 %
SSF (unpublished)	3.30 %	7.74 %	4.03 %	4.12 %	14.57 %	5.99 %
SOSF (unpublished)	3.98 %	7.82 %	4.62 %	4.26 %	12.31 %	5.70 %
OSF ([MG15])	4.14 %	11.12 %	5.29 %	4.49 %	16.33 %	6.61 %
PR-Sceneflow ([VSR13])	4.41 %	13.09 %	5.84 %	6.35 %	16.12 %	8.10 %
FSF+MS ([TSS17])	5.42 %	10.76 %	6.30 %	6.55 %	16.65 %	8.35 %
CSF ([LBA$^+$16])	4.03 %	11.82 %	5.32 %	6.39 %	16.75 %	8.25 %
SGM+SF ([Hir05, HFR14])	4.75 %	14.22 %	6.31 %	8.34 %	18.71 %	10.20 %
PCOF-LDOF ([DPSLB16])	5.98 %	18.40 %	8.03 %	8.40 %	26.59 %	11.66 %
PCOF+ACTF ([DPSLB16])	5.98 %	18.40 %	8.03 %	8.36 %	32.86 %	12.74 %
SGM+C+NL ([Hir05, SRB14])	4.75 %	14.22 %	6.31 %	15.72 %	20.79 %	16.63 %
SGM+LDOF ([Hir05, BM11])	4.75 %	14.22 %	6.31 %	17.08 %	18.66 %	17.36 %
DWBSF ([RKVT16])	18.76 %	21.14 %	19.16 %	23.92 %	21.88 %	23.55 %
GCSF ([ČSRH11])	11.24 %	26.26 %	13.72 %	21.88 %	31.66 %	23.63 %
VSF ([HD07])	26.38 %	19.88 %	25.31 %	52.30 %	40.83 %	50.24 %

Table B.3: KITTI'15 evaluation over non-occluded pixels. Columns marks categories of evalution: disparity **D1** for the first frame of the test pair and **D2** for the second; and subcategories: evaluation over **bg** background regions, evaluation over **fg** foreground regions and **all** evaluation over all ground-truth.

Method	Fl-bg	Fl-fg	Fl-all	SF-bg	SF-fg	SF-all
ISF (unpublished)	4.21 %	6.83 %	4.69 %	5.31 %	11.65 %	6.45
OSF+TC (ours)	4.34 %	9.67 %	5.31 %	5.52 %	15.57 %	7.32
PRSM ([VSR15])	4.33 %	10.80 %	5.50 %	5.54 %	17.65 %	7.71
SSF (unpublished)	4.20 %	10.81 %	5.40 %	5.70 %	19.93 %	8.25
SOSF (unpublished)	4.04 %	13.18 %	5.70 %	5.44 %	21.11 %	8.25
OSF ([MG15])	4.21 %	15.49 %	6.26 %	5.52 %	22.31 %	8.52
PR-Sceneflow ([VSR13])	6.94 %	20.24 %	9.36 %	8.35 %	26.08 %	11.53
FSF+MS ([TSS17])	6.53 %	20.72 %	9.11 %	9.23 %	28.03 %	12.60
CSF ([LBA$^+$16])	8.72 %	22.38 %	11.20 %	10.26 %	28.68 %	13.56
SGM+SF ([Hir05, HFR14])	13.36 %	21.78 %	14.89 %	15.28 %	29.68 %	17.86
PCOF-LDOF ([DPSLB16])	9.24 %	34.40 %	13.80 %	14.21 %	44.79 %	19.69
PCOF+ACTF ([DPSLB16])	9.77 %	57.63 %	18.45 %	14.67 %	64.73 %	23.63
SGM+C+NL ([Hir05, SRB14])	23.03 %	38.80 %	25.89 %	26.22 %	46.44 %	29.84
SGM+LDOF ([Hir05, BM11])	30.41 %	27.62 %	29.90 %	33.00 %	36.59 %	33.64
DWBSF ([RKVT16])	30.13 %	26.68 %	29.50 %	35.65 %	34.86 %	35.51
GCSF ([ČSRH11])	38.12 %	37.77 %	38.05 %	43.64 %	52.41 %	45.21
VSF ([HD07])	41.15 %	41.85 %	41.28 %	61.14 %	59.17 %	60.78

Table B.4: KITTI'15 evaluation over non-occluded pixels. Columns marks categories of evalution: scene flow as **SF** and optical flow as **F**; and subcategories: evaluation over **bg** background regions, evaluation over **fg** foreground regions and **all** evaluation over all ground-truth.