Optical Flow with Gabor Jets

Bachelor Thesis

Lara Fried
15-729-627

Submitted on
August 15, 2021

Thesis Supervisor
Prof. Dr. Manuel Günther

University of Zurich

Department of Informatics

Bachelor Thesis

Author: Lara Fried, lara.fried@uzh.ch

Project period: 15.02.2021 - 15.08.2021

Artificial Intelligence and Machine Learning Group
Department of Informatics, University of Zurich

Acknowledgements

I would like to thank my supervisor Prof. Dr. Manuel Günther, for giving me the opportunity to work on this interesting topic and supporting me throughout the work process. His frequent inputs and valuable feedback made this thesis possible.

Abstract

Computational vision has become more and more relevant in the last few years. Analysis of image and video input is relevant to robotics and self-driving cars to name just a few examples. Because of this relevance, algorithms for stereo matching need to get more precise and faster. This thesis considers the use of Gabor Jets for disparity estimation, evaluates its efficiency, and compares it to other algorithms. For this comparison, three different algorithms have been chosen: Horn Schunk Optical Flow, Lius Optical Flow, and StereoBM. To be able to compare different types of images, a sample from the DrivingStereo as well as the Middlebury dataset has been selected. For the evaluation, the bad matched pixels, mean relative error, and mean absolute error of all four algorithms were calculated. Gabor Jets worked better on images taken in a real-life setting than in a studio setting. Especially in images with a lot of traffic, Gabor Jets work better than the other algorithms. Based on these evaluations, Gabor Jets can keep up with the aforementioned algorithms and even surpass them in certain aspects.

Zusammenfassung

Computational Vision hat in den letzten Jahren immer mehr an Bedeutung gewonnen. Die Analyse von Bild- und Videoeingaben ist für die Robotik und selbstfahrende Autos relevant, um nur einige Beispiele zu nennen. Aufgrund dieser Relevanz müssen Algorithmen für das Stereo-Matching immer präziser und schneller werden. In dieser Arbeit wird die Verwendung von Gabor Jets für die Disparitätsschätzung untersucht, ihre Effizienz bewertet und sie mit anderen Algorithmen verglichen. Für diesen Vergleich wurden drei verschiedene Algorithmen ausgewählt: Horn Schunk Opticalflow, Lius Opticalflow und StereoBM. Um verschiedene Bildtypen vergleichen zu können, wurde ein Beispiel aus dem DrivingStereo sowie dem Middlebury-Datensatz ausgewählt. Für die Bewertung wurden die Bad Matched Pixels, der mittlere relative Fehler und der mittlere absolute Fehler aller vier Algorithmen berechnet.Gabor-Jets funktionierten bei Bildern, die in einer realen Umgebung aufgenommen wurden, besser als in einer Studioumgebung. Besonders gut funktionierte die Disparitätsschätzung mit Gabor Jets bei Bildern mit viel Verkehr. Basierend auf diesen Auswertungen können Gabor Jets mit den oben genannten Algorithmen mithalten und übertreffen sie sogar in einigen Aspekten.

Contents

1 Introduction **1**

2 Related Work **3**

2.1 Algorithms . . . 3

2.1.1 StereoBM . . . 3

2.1.2 Horn Schunck Method . . . 3

2.1.3 Liu . . . 5

2.1.4 Machine Learning Methods . . . 5

2.2 Gabor Jets . . . 5

3 Approach **9**

3.1 Implementation Details . . . 9

3.1.1 Gabor Jets . . . 9

3.1.2 StereoBM . . . 9

3.1.3 Liu . . . 9

3.1.4 Horn Schunck . . . 10

3.2 Preprocessing of Images . . . 10

3.2.1 Impact of Image Size . . . 10

3.2.2 Impact of Color . . . 10

3.2.3 Impact of Border . . . 11

3.3 Datasets . . . 11

3.3.1 DrivingStereo . . . 11

3.3.2 Middlebury . . . 12

3.4 Disparity Map Evaluation . . . 13

3.4.1 Mean Absolute Error . . . 13

3.4.2 Mean Relative Error . . . 13

3.4.3 Bad Matched Pixels . . . 14

4 Results **15**

4.1 Parameters . . . 15

4.2 Results on the Driving Stereo Dataset . . . 15

4.2.1 Mean Relative Error . . . 17

4.2.2 Mean Absolute Error . . . 17

4.2.3 Bad Matched Pixels . . . 18

4.3 Results of the Middlebury Dataset . . . 18

4.3.1 Mean Relative Error . . . 18

4.3.2 Mean Absolute Error . . . 21

4.3.3 Bad Matched Pixels 21

5 Discussion **23**

6 Conclusion **29**

List of Figures

2.1 Illustration of the brightness gradient vector (E_x, E_y) . . . 4
2.2 Gabor Wavelet . . . 7
2.3 Similarities with Gabor Wavelets . . . 7

3.1 Disparity with Gabor Jets, Image size . . . 10
3.2 Disparity with Gabor Jets, Colour channels . . . 11
3.3 Disparities with Gabor Jets, 100px border . . . 11
3.4 DrivigStereo disparity map . . . 12
3.5 Middlebury example image . . . 13

4.1 DrivingStereo Disparity with Gabor Jets and StereoBM . . . 16
4.2 Boxplots for MRE on DrivingStereo . . . 17
4.3 Boxplots for MAE on DrivingStereo . . . 18
4.4 Boxplots for the BMP of DrivingStereo . . . 19
4.5 Middlebury disparity . . . 19
4.6 Boxplots of MRE of Middlebury . . . 20
4.7 Boxplots for MAE of Middlebury . . . 22

5.1 Largest and smallest MAE for StereoBM on DrivingStereo . . . 24
5.2 Best BMP Horn Schunck and Liu . . . 25
5.3 Highest BMP of DrivingStereo . . . 25
5.4 Images with the lowest MRE value on DrivingStereo . . . 26
5.5 MRE on Middlebury . . . 27
5.6 Disparities for Recycle . . . 28

List of Tables

4.1 Calculation Times 17
4.2 MRE values on Middlebury 20
4.3 MAE values on Middlebury 21
4.4 BMP Values on Middlebury 22

5.1 MAE values on DrivingStereo 23
5.2 BMP values on DrivingStereo 24
5.3 MRE values on DrivingStereo 25

Chapter 1

Introduction

In the last two decades, computer vision has developed quickly and become increasingly relevant. The underlying goal is to reconstruct information about shape, location, and illumination of objects from images. Szeliski (2011) One of the many topics in computer vision is optical flow. Optical flow is the pattern of motion, and it can be used for a range of tasks such as gesture recognition or to represent temporal information. Lappe (2009) Ke et al. (2018) A neighbouring field to this is stereo matching. With stereo matching or disparity estimation, the aim is to find a correspondence between pixels of two stereo images. Orozco et al. (2017) While it can be used to calculate optical flow in videos, one of the main tasks of stereo matching is to estimate the depth of a binocular set of images to reconstruct a 3D version of it. There are no public video databases that provide ground truths; there are multiples that provide pairs of stereo images with ground truths.Lai et al. (2019) This is why this thesis focuses on disparity estimation of stereo images. For the practical aspect of this thesis, the datasets Middlebury and DrivingStereo were usedYang et al. (2019)Scharstein et al. (2014). Further options would have been the datasets Kitti or ETH3D Urtasun (2012) Schöps et al. (2017), but since DrivingStereo provided the most images and Middlebury seems to be the most commonly used database, these are the ones used for this thesis.

Because of its relevance in robotics and autonomous driving, amongst other fields, there are many different approaches to both stereo matching and optical flow. This thesis aims to compare the application of Gabor Jets for disparity estimation on stereo images to other established algorithms.

One of these algorithms is a version of the Horn Schunck Optical Flow proposed in 1981. The implementation used here is based on Bob and was added in 2012. Horn Schunck is considered a global method for disparity estimation.Anjos et al. (2012) Horn and Schunck (1981) Another implementation provided in bob is Liu's Optical Flow. C.Liu developed this Optical Flow estimator in 2009. Liu (2009) Lastly, the StereoBM algorithm implemented on OpenCV will be an example of a block matching algorithm. Bradski (2000)

For different points in the image, Gabor Jets are generated from the responses of Gabor Wavelets at this position. They encode the texture of the image around the point in question. For the following tests, the implementation of Gabor Jets that was added to Bob in 2011 will be used.Anjos et al. (2012)Günther et al. (2012) For the evaluation, both the calculation time and the accuracy of the results will be compared. There are different methods of determining the accuracy. The methods used in this thesis are bad matched pixels, mean relative error, and mean absolute error. All of these compare the disparities calculated by the algorithms to the ground truths provided by the datasets. Kim et al. (2021) Cabezas et al. (2012) This thesis will first introduce the algorithms used for the comparison in the Chapter "Related Work". In the chapter "Approach" the used datasets will be introduced, as well as the evaluation methods and some implementation details. Under "Results" the outcome of the different stereo matching approaches is shown. Finally all relevant results will be interpreted in the "Discussion".

Chapter 2

Related Work

2.1 Algorithms

There are many different approaches to stereo matching. In 1981 Lucas and Kanade developed a technique for image matching by using the spatial intensity gradient of images. They also showed the application of their technique in a stereo vision system. Lucas and Kanade (1981) Today, Lucas-Kanade is still one of the most popular methods for solving optical flow for a locally constant motion. Many approaches build on this original method. Radgui et al. (2008) There are different classifications of traditional stereo vision algorithms. Methods for disparity estimations can be divided into the following classes: block-based stereo matching, graph cut-based stereo matching, and semi-global matching. Block-based stereo matching is a local method, that uses a constraint on a small number of pixels around the pixel of interest. For the semi-global matching, a block-based matching is smoothed by path-wise information from multiple directions. Graph cut-based stereo matching uses constraints on the whole image. Raza et al. (2015) There are multiple deep learning and machine learning approaches to optimize the results and make stereo matching more cost-efficient. Poggi et al. (2021)

2.1.1 StereoBM

OpenCV is an open-source library of programming functions for computer vision. It also contains multiple algorithms that can be used for stereo matching. One example that will be focused on is the class StereoBM [1]. StereoBM implements a block matching algorithm to compute disparities between stereo images. It was added to OpenCV by K.Konolige. A block matching algorithm segments each image into n x n blocks. Each block from the left image is then matched to a block in the right image.Yaakob et al. (2013) Bradski (2000) OpenCV also contains a StereoSGBM class which implements an algorithm based on the work of Hirschmuller (2008) on semi-global matching algorithms. It also matches blocks instead of single pixels. Instead of the eight directions considered in the original algorithm, this version only considers five. The complexity of the SGM method is linear to the number of pixels. It uses pathwise optimizations to perform a fast approximation .Hirschmuller (2008) Bradski (2000) Scharstein et al. (2014)

2.1.2 Horn Schunck Method

In 1980 Horn and Schunk developed an algorithm for the determination of optical flow from a sequence of images. The algorithm is based on an equation combining two components of the

[1] https://docs.opencv.org/3.4/d9/dba/classcv_1_1StereoBM.html

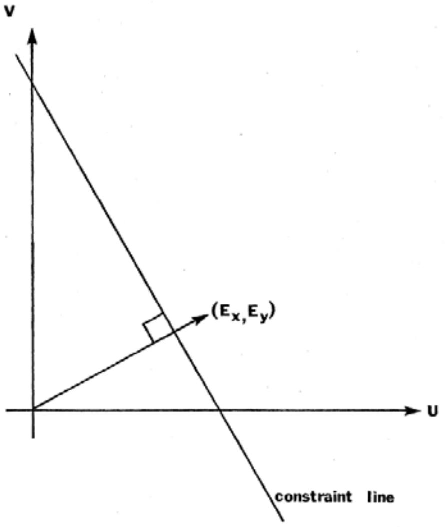


Figure 2.1: Illustration of the brightness gradient vector (E_x, E_y) for Horn Schunck. The velocity (u,v) has to lie in a right angle to the gradient vector. Horn and Schunck (1981)

flow velocity. The optical flow component in the direction of the brightness gradient (E_x, E_y) with E_t, E_x and E_y being partial derivates of image brightness is:

$$-\frac{E_t}{\sqrt{E_x^2 + E_y^2}} \tag{2.1}$$

The brightness gradient vector in relation to the velocity (u,v) is illustrated in Figure 2.1

Because there is no way to compute the flow velocity locally without additional constraints, the smoothness of the flow is used as the second constraint. Smoothness here stands for the minimization of distortions in the flow. To express this constraint, one can minimize the sum of the squares of the laplacians of the x- and y components of the flow. u and v are the vertical and horizontal components of the flow field.

$$\nabla^2 u = \frac{\delta^2 u}{\delta x^2} + \frac{\delta^2 u}{\delta y^2} \text{ and } \nabla^2 v = \frac{\delta^2 v}{\delta x^2} + \frac{\delta^2 v}{\delta y^2} \tag{2.2}$$

Horn and Schunck developed the algorithm to solve this equation iteratively. Horn and Schunck (1981) Bob contains an optical flow estimator based on this work, the functor bob.ip.optflow. hornschunck.VanillaFlow is the classical optical flow estimator based on the original work of Horn and Schunk. For this, the parameters alpha and iterations are needed. Alpha describes the weighting factor between the smoothness of the field and brightness constants. A good default for alpha values is around 200. Iterations are the number of iterations used to minimize the flow error. Anjos et al. (2012) Horn and Schunck (1981)

2.1.3 Liu

As a part of his thesis, Liu developed a method to derive optical flow. This approach differs from the other algorithms as following, for the optical flow computation, it uses a technique called Iterative Reweighted Least Squares (IRLS). It iterates between computing the weight, computing the nonlinear term, and solving a least-squares problem. The weight is defined as two diagonal matrices ϕ and ψ. and the nonlinear term is the computation of the weight based on the current estimate of dU and dV. With these to factors the linear equation

$$\begin{bmatrix} \psi' I_x^2 + \alpha L & \psi' I_x I_y \\ \psi' I_x I_y & \psi' I_y^2 + \alpha L \end{bmatrix} \begin{bmatrix} dU \\ dV \end{bmatrix} = - \begin{bmatrix} \psi' I_x I_z + \alpha L U \\ \psi' I_y I_z + \alpha L V \end{bmatrix} \tag{2.3}$$

These steps are repeated until dU and dV converge. With IRLS, the filters that have been learned from ground-truths can be used. This makes it possible to get better characteristics of flow fields. Another advantage of IRLS is its ability to handle large-magnitude flows for the temporal constraint. IRLS always converges to a local minimum which is shown by showing equivalence of IRLS to the variational upper-bound optimization Jordan (1999) Liu (2009) Bob contains two implementations of Ce Liu's Optical Flow, the Conjugate-Gradient based implementation and the Successive Over Relaxation implementation. Both implementations use a coarse-to-fine approach to compute the dense optical flow field. The function takes two grayscale input images and optionally the regularization weight as alpha. It returns a 2D double array with the same dimensions as the input, which contains the output velocities in x and y. Anjos et al. (2012)

2.1.4 Machine Learning Methods

There are several approaches to use deep learning for stereo matching. Hamid et al. (2020) Using traditional cost aggregation and disparity computation methods to get disparity maps with deep neural networks did achieve reasonable accuracy. However, they did tend to give wrong predictions in large texture less or reflective regions and occluded regions and around the edges of objects. Zhang et al. (2019) Deep learning approaches leverage on CNNs to infer confidence maps. There are two subcategories Cost-Volume CNNs and Disparity CNNs. Cost-volume CNNs performed better in experiments, but disparity CNNs were competitive, particularly with noisy stereo algorithms Poggi et al. (2021) Of course, there have also been multiple approaches using traditional machine learning techniques. Many of them use classifiers, specifically forests, to improve the cost of the Semi-Global Matching pipeline(SGM) algorithm.Poggi et al. (2021) Because of a lack of available source code this thesis does not use a deep learning or machine learning approach. The reimplementation of such an approach would have exceeded the timeframe for this bachelor thesis.

2.2 Gabor Jets

In 1946 Gabor first proposed the Gabor function, a linear filter often used for texture analysis and feature extraction. It has been found that Gabor functions can be used to model the simple cells in the brains of mammals. Because of this, the image analysis using Gabor functions is very similar to actual human perception. Daugman (1985) Barina (2016) Gabor (1946) Tai Sing Lee (1996) One directional Gabor Wavelets are Gabor functions created by dilation and shift from one Gabor Wavelet, the mother Gabor Wavelet. There are continuous Gabor Wavelets and discreet Gabor Wavelets. The two directional Wavelets have additional parameters such as the rotation of the spatial wave. The implementation of Gabor Jets on bob is based on the discreet Gabor wavelet family. To process an image using Gabor Wavelets, a Gabor transformed image is generated. This image consists of layers, each convoluting the original image with the corresponding Gabor Wavelet. Because this Gabor transformed image contains much redundant information, only some Gabor Wavelet responses are kept. A Gabor jet is generated by combining the responses of these Gabor Wavelets at specific positions into a vector. A Gabor Jet codes the texture of an image around its extraction point. Günther et al. (2012) Buhmann et al. (1989)

Disparity Map Estimation

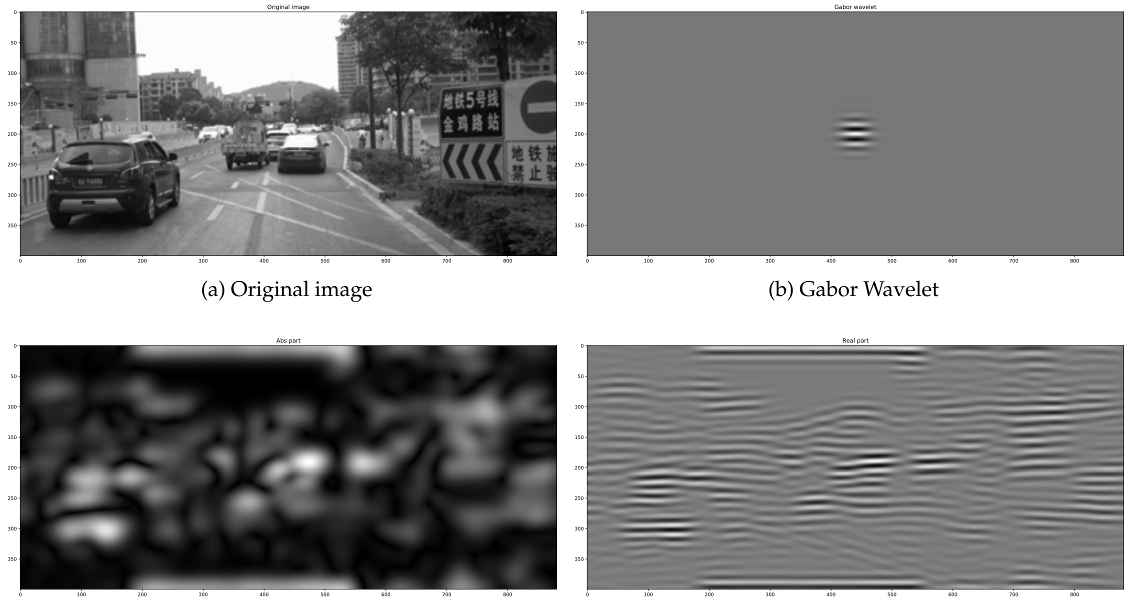
The disparity can be estimated from two Gabor Jets. Günther et al. (2012) Sanger proposed a method that used a complex Gabor filter to convolve the left and right images. He then used the difference in the complex phase at each point to indicate a local shift. This algorithm does not require a formal matching process since it senses disparity directly from a locally computed parameter. Sanger (1988) In 2015 Malathi and Bhuyan proposed a feature-based stereo matching method, in which local features of Gabor Wavelets in spatial domains are used for matching cost computation. To reduce the dimensionality of the local Gabor Wavelet coefficients, they used principal component analysis. They described the following advantages of using Gabor Wavelet in spatial domain: Plausibility to model the visual system. Processing of local areas of interest. Faster than conventional FFT implementations. Malathi and Bhuyan (2015) DeJong et al. showed that FlowNetS is based on Gabor filters. They found that a Gabor filter based on two frames limited the performance for flow velocity estimation. Performance could be improved by using more frames. De Jong et al. (2021) There have been approaches using Gabor responses as input for CNN's.Hosseini and Cho (2019)

Implementation in Bob

Gabor Wavelets and Jets are implemented in bob.ip.gabor. The function bob.ip.gabor.Jet(trafo-image,pos) can be used to create a Gabor Jet for a fixed position on an image. Günther et al. (2012)Anjos et al. (2012)

Figure 2.2b shows the Gabor Wavelet used for the calculations of the real and absolute part of the original image. In the absolute part 2.2c the border on the top and bottom of the image looks very similar despite the differences seen in the original image, 2.2a it can also be observed in the real part. 2.2d This is due to the wrap-around which happens in the Gabor Wavelet transform, meaning there are responses from one side represented on the opposite side of the image.

Further there is a function bob.ip.Gabor.Similarity("Disparity", gwt) that uses these Jets and calculates the similarities within an image.Figure 2.3 shows how Gabor wavelets can be used to find similarities in an image. Figure 2.3b shows the similarity of each pixel in the image to the selected point marked with an X in Figure 2.3a . By using these functions it is possible to create a disparity map of two stereo images.



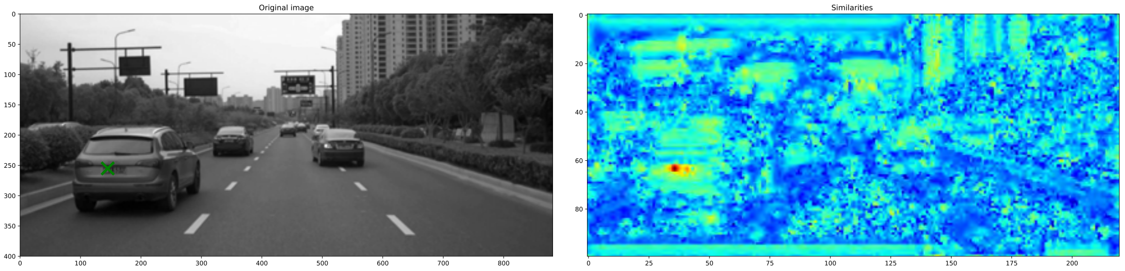
(a) Original image

(b) Gabor Wavelet

(c) Absolute part

(d) Real part

Figure 2.2: The response of Gabor Wavelets to an image from DrivingStereo



(a) Original Image

(b) Similarities

Figure 2.3: Gabor Wavelets can be used to find similarities within an image.

Chapter 3

Approach

The practical part of this thesis calculates disparity maps for images from the DrivingStereo and the Middlebury dataset, using all four introduced approaches. And evaluates them using the MRE, MAE and BMP values. Because for the DrivingStereo images the ground truth is only one value per pixel, all disparities in this thesis are calculated in horizontal direction only.

3.1 Implementation Details

3.1.1 Gabor Jets

To estimate the disparity between the two stereo images, the bob.ip.Gabor API can be used.[1]Günther et al. (2012) The function bob.ip.gabor.Jet(trafo-image,pos) can be used to create a Gabor jet for a fixed position on an image. Günther et al. (2012)Anjos et al. (2012)

To transform an image with bob.ip.Gabor.Transform there are two relevant parameters: The number of scales and k_max. These parameters can increase the region from which a Gabor Jet is extracted. There are also parameters to set the number of directions, the spatial resolution of Gabor Wavelets, and the distance between two scales of Gabor Wavelets. For these parameters, the default values were used.

3.1.2 StereoBM

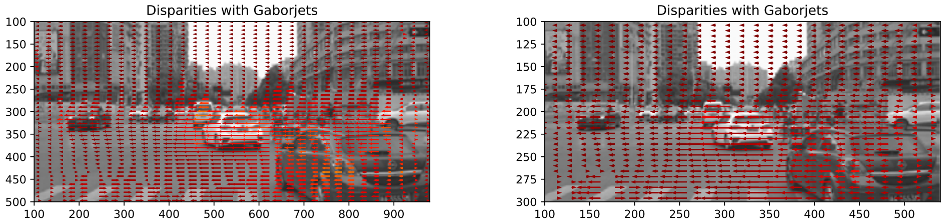
The function StereoBM_create takes the parameters numDisparities and blockSize. With numDisparities, the search range is given. This means that for each pixel, the algorithm searches for a number of best disparities. The second parameter is blockSize, and this parameter gives the linear size of the blocks used for comparison. This parameter has to be odd. A smaller block size gives a more accurate disparity map but allows for more wrong correspondences. A larger block size results in a smoother but less accurate disparity map.

3.1.3 Liu

The the cg version, which was used for this thesis is implemented on bob as well[2]. For Liu's algorithm, multiple optional parameters can be set. They give regularization weight, downsample

[1] https://www.idiap.ch/software/bob/docs/bob/docs/stable/bob/bob.ip.gabor/doc/py_api.html

[2] https://www.idiap.ch/software/bob/docs/bob/bob.ip.optflow.liu/master/py_api.html



(a) Original size

(b) Rescaled

Figure 3.1: Disparity with Gabor Jets, original and rescaled image.

ratio, width of coarsest level, number of outer fixed-point iterations, number of inner fixed-point iterations, and number of cg iterations.

3.1.4 Horn Schunck

For the VanillaFlow function in bob.ip.optflow.hornschunck the parameters alpha and iterations can be set [3]. Alpha is the weighting factor between brightness costs and field smoothness. The number of iterations is used to minimize the flow error

3.2 Preprocessing of Images

The preprocessing of the images is important because not every algorithm tested is capable of handling color. Further, the size of images can have a significant impact on the duration of calculation.

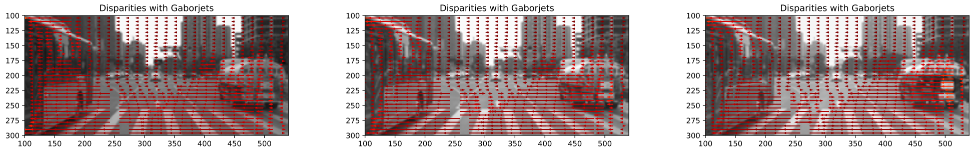
3.2.1 Impact of Image Size

The original images in the DrivingStereo dataset are 881 x 400 pixels large. The disparity calculation with Gabor Jets took around 125 s for the large images. To reduce this calculation time the images were rescaled to half the size. This resulted in faster calculation but only slightly worse results.

3.2.2 Impact of Color

One aspect inspected during this thesis was the influence of the colour channels on the results. Since the algorithm worked with black and white images, all data had to be converted to black and white. In this process, it is possible to give different weights to the different colour channels. To compare eventual differences between the color channels three versions of the images have

[3] https://www.idiap.ch/software/bob/docs/bob/bob.ip.optflow.hornschunck/stable/py_api.html

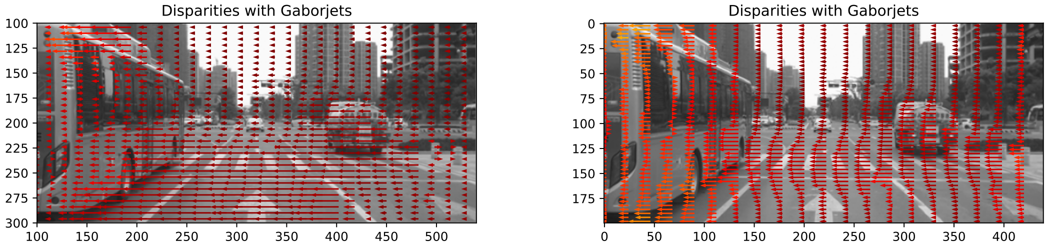


(a) Red channel

(b) Green channel

(c) Blue channel

Figure 3.2: Disparities on images where only one channel was used.



(a) Disparities with border

(b) Disparities without border

Figure 3.3: Example of an image with a black border.

been prepared using the library PIL, which offers a function to extract only one colour channel from an image. In Figure 3.2 three graphs created with each only one color channel are shown.

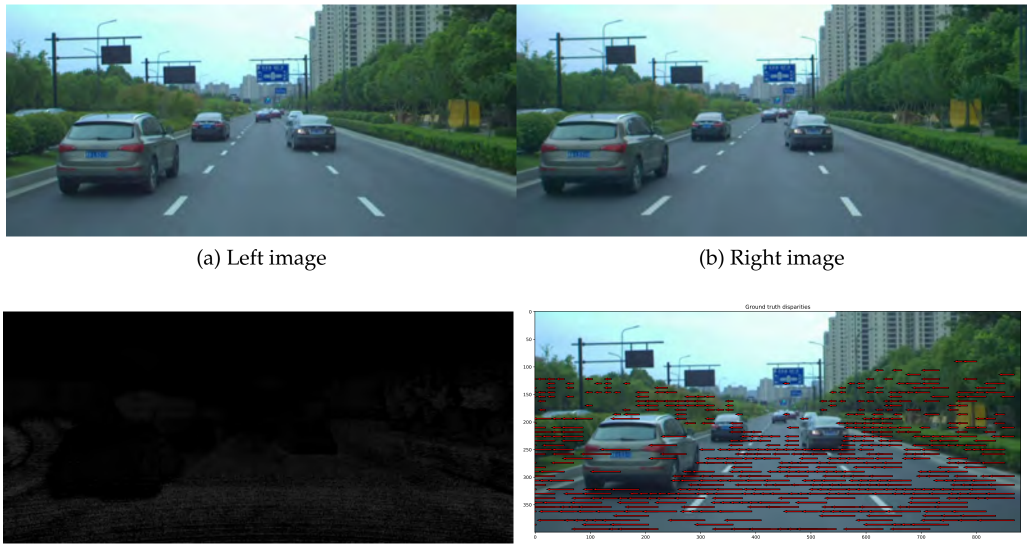
3.2.3 Impact of Border

Gabor Jets extract local frequency information around the target position. At the border of the images, by default, the images are wrapped around, this leads to inaccuracies in the border region. To minimize this the black border of 100 pixels was added (see Figure 3.3). Instead of random pixels the black border is now used to calculate the Gabor Jets for the border pixels.

3.3 Datasets

3.3.1 DrivingStereo

The Driving Stereo dataset was published in 2019. It consists of training and testing datasets. The training set consists of 174431 image pairs, and the testing dataset contains 7751 image pairs. Each image is 881x400 px in size. They provide disparity and depth maps for all frames. The disparities



(a) Left image

(b) Right image

(c) Disparity map provided by DrivingStereo

(d) Disparity map with the ground truth values visualized with arrows

Figure 3.4: Disparity maps for an example image of DrivingStero

are given as values on a pixel. This means the disparity is not directed. For the following results the disparity is assumed to be horizontally directed. The DrivingStereo dataset provides disparity maps for the training data. The disparity maps are saved as uint16 PNG images. This means the disparity value can be computed for each pixel by converting the uint16 value to float and dividing it by 256. The labels of disparity have been produced by using multi-frame LiDAR points. Yang et al. (2019) LiDAR, short for Light Detection and Ranging, is a technology that can be mounted on a vehicle. It records data about the surface of the Earth using a laser. Li and Ibanez-Guzman (2020) The resulting ground truth only contains information about points where the laser is reflected back to the sensing units. Because of this, there are points where no data are available in the ground truth. The disparity maps can only be evaluated where data are available.

3.3.2 Middlebury

The Middlebury Stereo Datasets provide stereo images complete with disparity maps. Since 2001 only 71 image pairs collected have been collected. The images vary slightly in size but most of them are around 2864 x 1924 px large. As the techniques of obtaining these images have been improved over time, only the images from 2014 have been used for this comparison. There was a part of the images used for evaluation where no ground truths were provided, meaning for this thesis 14 images were actually used. The ground truths for the Middlebury dataset have been acquired by using structured light. This structured-light technique uses a projector to project one or more light patterns on a scene. With this, dense and pixel-accurate correspondences can be produced. The disparity values d can be converted to depth with the following equation, where Z is the depth in mm, d is the disparity value in pixels, doffs the x-difference of principal points, and f is the pixel's focal length. Scharstein et al. (2014)

$$Z = baseline * f/(d + doffs). \tag{3.1}$$

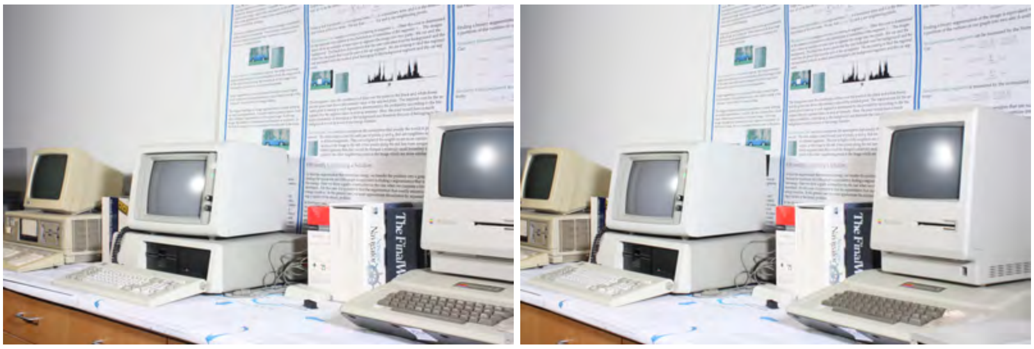


Figure 3.5: Example image from the Middlebury Dataset, left and right image.

3.4 Disparity Map Evaluation

The main point of evaluation for disparity maps is accuracy. To get the accuracy of the calculated disparity maps, they can be compared to the disparity maps provided by the dataset. There are multiple ways to calculate the accuracy of disparity maps to get values that can be compared. For these results, the mean absolute error, the mean relative error and the bad matched pixels percentage were calculated. For this calculations the values of a grid from the ground truth and estimated image were used, this grid contains every 10 pixel in width and height. This grid was implemented because there was to much data if every pixel was considered. The grid is completely unconditional to the availability of disparity values. Because of this a lot of the values were 0 since both ground truth and the disparity maps calculated by the algorithms do not have results for every pixel.

3.4.1 Mean Absolute Error

The mean absolute error calculates the absolute difference between the ground truth value and the disparity value. X and y are the coordinates used to refer to the disparity values in the grid of values created for each image.

$$MEA = \frac{1}{N}\sum_{(x,y)}^{N} |D_{true}(x,y) - D_{estimated}(x,y)| \tag{3.2}$$

Kim et al. (2021)

3.4.2 Mean Relative Error

The MRE calculates the ratio of error magnitudes against actual disparity values. With this method, every value that deviates from the ground truth is considered an error. Cabezas et al. (2012)

$$MRE = \frac{1}{N} \sum_{(x,y)}^{N} \frac{|D_{true}(x,y) - D_{estimated}(x,y)|}{D_{true}(x,y)} \tag{3.3}$$

X and y are coordinates of the disparity values D in the grid used for the evaluation.

3.4.3 Bad Matched Pixels

BMP is often used for the evaluation of disparity maps. It is given as a percentage and calculated by counting the number of differences between the ground truth and estimated disparity values that exceed a given threshold. Cabezas et al. (2012) X and y are coordinates in the grid used for the evaluation. The BMP value is calculated as follows:

$$\epsilon(x,y) = \begin{cases} 1 \text{ if } |D_{true}(x,y) - D_{estimated}(x,y)| > \delta \\ 0 \text{ if } |D_{true}(x,y) - D_{estimated}(x,y)| \leq \delta \end{cases} \tag{3.4}$$

with epsilon being the difference value between the disparity values D for threshold delta.

$$BMP = \frac{100\%}{N} \sum_{(x,y)}^{N} \epsilon(x,y) \tag{3.5}$$

TN is the number of points used for the comparison.

Chapter 4

Results

In the following, the results of the disparity estimations with Gabor Jets, Horn Schunck, Liu, and StereoBM for images from the Driving Stereo Dataset and images from the Middlebury dataset are given. For each of the algorithms, the disparity maps of the different images are created. These disparity maps are then compared to the ground truths and evaluated by using the previously introduced BMP, MAE and MRE formulas.

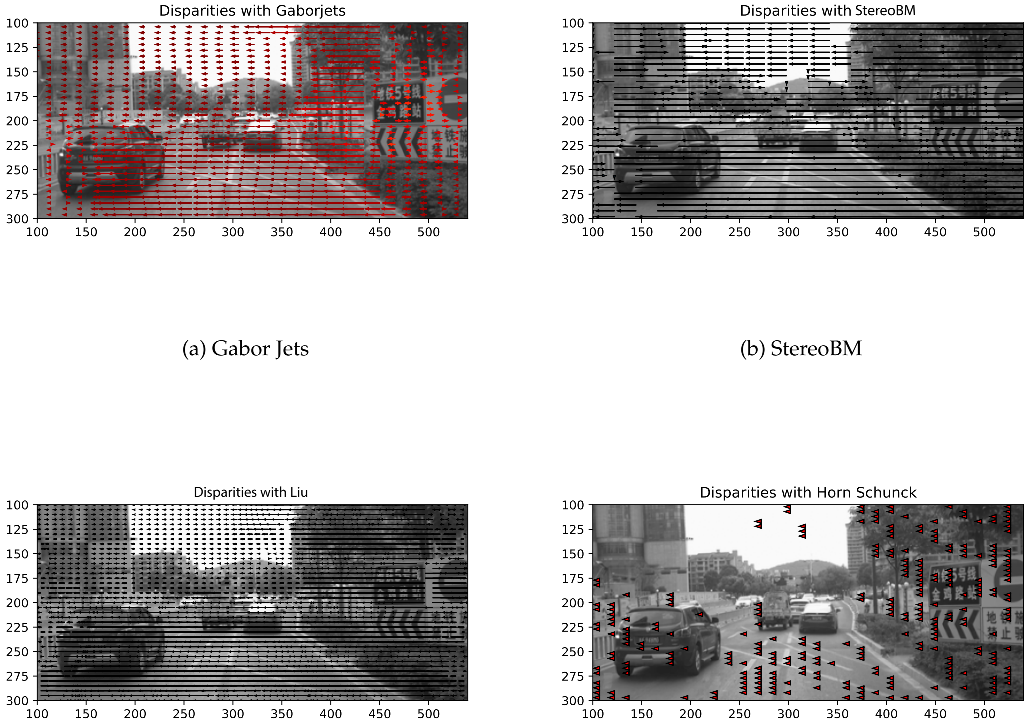
4.1 Parameters

Preliminary to calculating all the values, the best parameters were determined for all of the algorithms. For all the results in this thesis, the same parameters have been used. For Gabor Jets, the number_of_scales was set to 12, and k_max was set to pi/4. With StereoBM, the parameter numDisparities was set to 32 and blockSize to 5, which is a relatively small blockSize. For Liu, only the alpha parameter, which sets the regularization weight, has been manually set to 0.5. The parameters of Horn Schunck have been set to alpha = 200 and iterations = 20. The recommended default value of alpha = 200 proves to work best in this case. For all parameters not mentioned here, the default values have been used.

4.2 Results on the Driving Stereo Dataset

In the following section, the results of the data from DrivingStereo are described. For these results, only the folder 18-07-09-16-11-56 from the training data was used. This means that for the following results, 2776 pairs of stereo images were used. Only a part of the available data was used for efficiency reasons because some of the algorithms have a very long calculation time, and 2776 is a large enough sample to get representative results while still having reasonable running times. Because of the number of images used, the complete tables are not represented in this chapter.

Figure 4.1 shows examples of disparity maps computed by the different algorithms. There is a difference in both the length and placement of the disparity arrows. Both Gabor Jets and Liu provide disparities for each of the chosen pixels, while StereoBM and Horn Schunck do not. Horn Schunck results in zero for many pixels. Because of that, only a few arrows are plotted.



(a) Gabor Jets

(b) StereoBM

(c) Liu

(d) Horn Schunck

Figure 4.1: Disparity maps on an example image of DrivingStereo

Table 4.1: Calculation Times

Algorithm	Rescaled image	Original Image
Gabor Jets	5.39 s	124 s
Horn Schunck	2.28 s	5.82 s
Liu	22.6 s	68.1 s
StereoBM	0.5 s	1.7 s

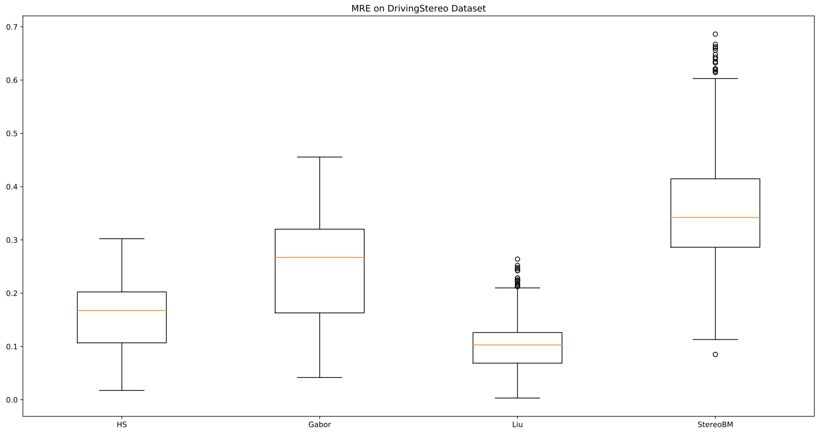


Figure 4.2: Boxplot of the MRE values for the DrivingStereo Dataset

Calculation time

As mentioned previously, the running times are problem points for some of the algorithms. Rescaling the images already took the running time of the Gabor Jets from 124 s down to 5.39 s on average.
Table 4.1 shows the average calculation times measured after the preprocessing.

4.2.1 Mean Relative Error

Figure 4.2 shows the boxplots for the MRE values on the DrivingStereo dataset for all four approaches. The smallest distribution and lowest mean is achieved by Liu. StereoBM has the highest mean. Gabor Jets have a distribution span between 0.1 and 0.48. This leads to a mean below the mean of StereoBM but above the means of Horn Schunck and Liu.

4.2.2 Mean Absolute Error

Considering the mean value for all the used images, Horn Schunck did perform the best with a mean of 74.0739. Followed by Gabor Jets, which had a mean of 91.3251 overall images. While StereoBM has the lowest MAE value for a single image, its mean over all images was 373.2638, which is considerably higher than both Horn Schunck and Gabor Jets.

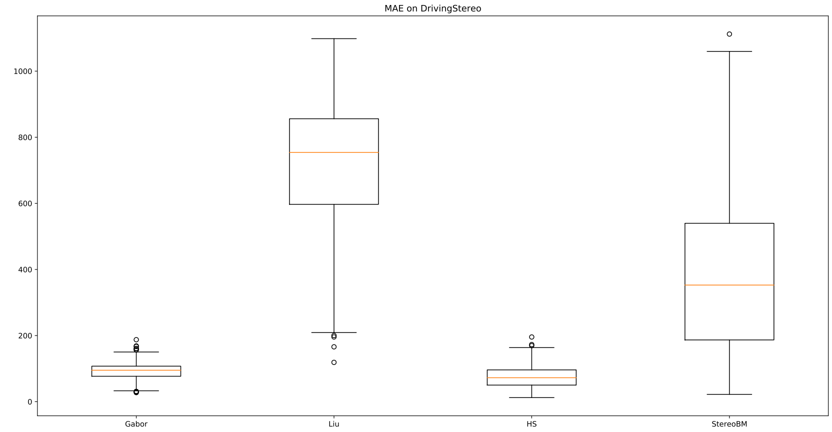


Figure 4.3: Boxplot of the MAE values for the DrivingStereo Dataset

In the boxplots in 4.3 one can see that Gabor Jets and Horn Schunck have a much smaller distribution and lower mean than Liu and StereoBM. While the mean of Horn Schunck is lower than the mean of Gabor Jets, the Gabor Jet has a smaller distribution.

4.2.3 Bad Matched Pixels

The delta used to calculate the BMP was 1. Overall, Horn Schunck has the lowest percentage of bad matched pixels with only 13.55% on average. Gabor Jets follows with an average of 21.77 % bad matched pixels and Liu with an average of 23.43%. The highest mean is 28.52 % for StereoBM.

In accordance with the average values, StereoBM has the most outliers, while Horn Schunck has no outliers.

4.3 Results of the Middlebury Dataset

The following results are calculated on a sample from the Middlebury dataset. For the Middlebury dataset, only 14 images were used, because of this all results can be represented in this chapter. Figure 4.5 is the Middlebury image Vintage with Disparities calculated by Gabor Jets and OpenCV For the visualization of the StereoBM disparity map, the disparity values have been reduced by a factor of 10 to provide a clearer image as the disparity values are large and result in very long arrows.

4.3.1 Mean Relative Error

The following table shows the MRE values for the 14 pictures used from the Middlebury dataset.

As can be observed in the following boxplots, even though it has the most extensive distribution, StereoBM has the smallest mean. Gabor Jets has the highest mean, closely followed by Liu.

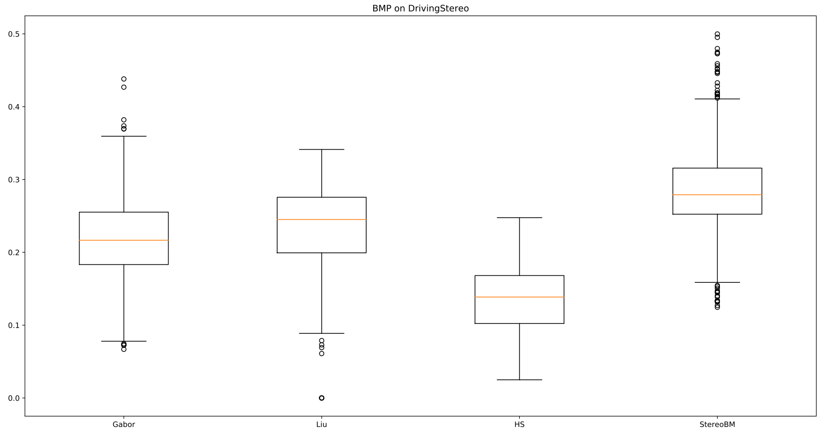
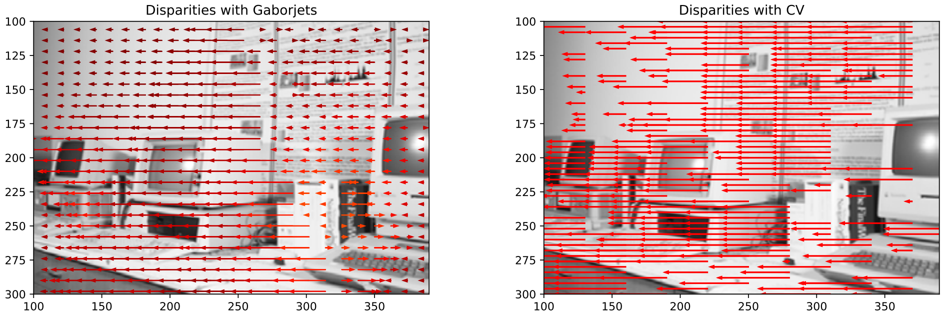


Figure 4.4: Boxplot of the BMP values for the DrivingStereo Dataset



(a) Gabor Jet

(b) Stereo BM

Figure 4.5: Disparity maps calculated with Gabor Jets and OpenCV StereoBM

Table 4.2: MRE values on Middlebury

	Gabor	Horn Schunck	Liu	StereoBM
Adriondeck	91.5783	81.0361	89.8556	53.2569
artL	96.4428	79.6193	94.497	106.7138
Jadeplant	89.6413	79.096	88.7431	71.7772
Motorcycle	90.1609	81.2317	88.429	61.9353
MotorcycleE	90.1095	81.2315	87.0907	61.8583
Piano	85.7692	76.5815	83.0779	61.9474
PianoL	85.7478	76.5812	82.3521	74.7148
Pipes	87.931	74.1561	85.287	79.1265
Playroom	87.8542	**73.3085**	84.8268	52.8175
Playtable	88.0375	80.9813	86.4987	54.2823
Recycle	90.4157	79.8748	88.4476	76.5829
Shelves	**81.9842**	74.6605	**80.1102**	49.3717
Teddy	87.1957	77.6905	85.9676	55.3391
Vintage	90.9087	75.2396	88.4163	**48.5351**
Mean	88.8412	77.9492	86.6857	64.8756
Standard deviation	3.3649	2.8551	3.5690	15.8169

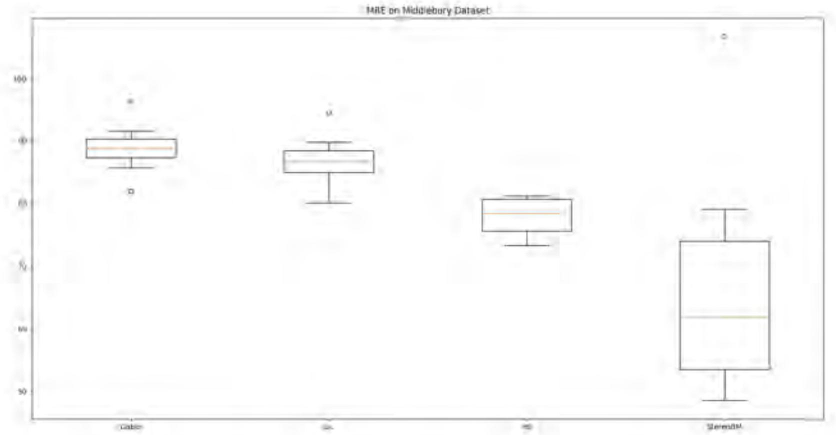


Figure 4.6: Boxplot of the MRE values for the Middlebury Dataset

Table 4.3: MAE values on Middlebury

	Image	Gabor Jets	Liu	Horn Schunck	StereoBM
0	adriondeck.png	10567.7803	10427.1794	5603.8633	10897.5880
1	artL.png	13002.8934	11814.1713	6842.4836	18166.42
2	jadeplant.png	23672.0341	23691.2211	10185.6195	24620.4516
3	motorcycle.png	13223.8054	13545.7312	7194.3181	14949.0959
4	motorcycleE.png	13224.8821	13309.7557	7194.309	14920.1306
5	piano.png	**9243.4309**	8706.6926	4807.2843	9295.3849
6	pianoL.png	9346.3648	**8646.9028**	4807.2659	10926.5457
7	pipes.png	11138.1309	10693.5352	5219.9251	8850.5846
8	playroom.png	13304.8526	12264.7048	7644.9797	12858.5404
9	playtable.png	12044.2622	11974.8094	6017.5559	12266.6762
10	recycle.png	9394.9474	8933.602	**3977.4386**	10604.8647
11	shelves.png	9885.5844	9191.0689	4913.0965	**9672.1493**
12	teddy.png	9798.3586	9648.8783	5131.6742	10223.68
13	vintage.png	22600.192	19919.7811	9569.3032	14308.2233

4.3.2 Mean Absolute Error

The mean absolute error for all the algorithms was generally very high for the Middlebury dataset. As can be seen in the boxplots in Figure 4.7, the Horn Schunck algorithm results in the lowest mean value. The results for Gabor Jets and Liu are very similar. They have almost the same mean and distribution. In table 4.3 one can see that they performed well on the same images. The mean result of StereoBM is close to the mean of Liu and Gabor Jets, but the distribution is a bit wider, it performed well on different images than Gabor Jets, and Liu did.

4.3.3 Bad Matched Pixels

In the table 4.4 one can see the BMP values of all the approaches. For the following results the delta for the BMP was set to 1

Both Horn Schunck and Liu's BMP values are extremely high. The BMP values of Gabor Jets and StereoBM are closer to their MRE values. The BMP values for Horn Schunck and Liu do not get lower when a smaller threshold is used to calculate the BMP.

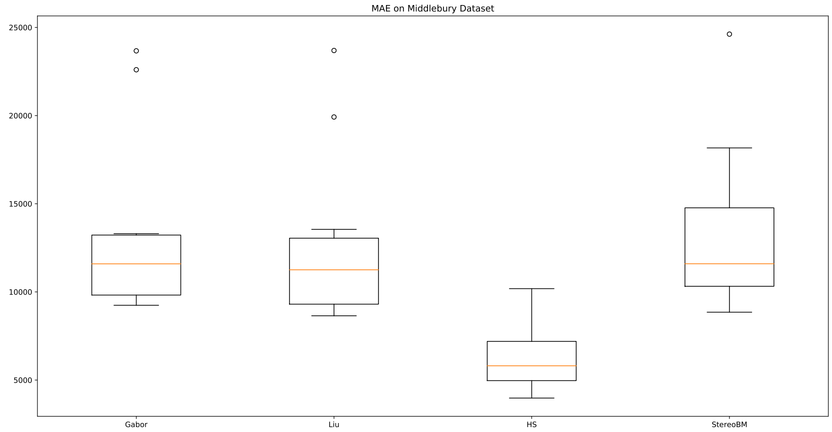


Figure 4.7: Boxplot of the MAE values for the Middlebury Dataset

Table 4.4: BMP Values on Middlebury

	Gabor Jets	Horn Schunck	Liu	StereoBM
Adriondeck	0.7407	0.9995	0.9998	0.5274
artL	0.7836	0.9995	0.9998	0.3538
Jadeplant	0.7177	0.9995	0.9998	0.6234
Motorcycle	0.7128	0.9995	0.9998	0.43201
MotorcyceE	0.7128	0.9995	0.9994	0.4347
Piano	0.7419	0.9995	0.9998	0.3060
PianoL	0.8738	0.9995	0.9998	0.5187
Pipes	0.7033	0.9995	0.9998	0.3259
Playroom	0.7345	0.9995	0.9998	0.5439
Playtable	0.8142	0.9995	0.9998	0.3684
Recycle	0.7695	0.9995	0.9998	0.3709
Shelves	0.7767	0.9995	0.9998	0.3873
Teddy	**0.6973**	0.9995	0.9998	**0.2869**
Vintage	0.7171	0.9995	0.9998	0.4806
Mean	0.7568	0.9995	0.9998	0.4327

Chapter 5

Discussion

For many of the applications of stereo matching, the speed of the algorithm is crucial. While the calculation duration for the Gabor Jets was very slow for the original images, it did significantly improve by rescaling the images. For the original images, it took around 124 s, for the rescaled images the average calculation time was reduced to 5.39 s. Although this is still slower than the 0.5 s of StereoBM or 2.28 s of Horn Schunck, it did surpass Liu, which takes on average 22.6 s on the rescaled images.

Considering the results of the evaluations, there is a difference between the two datasets in general as well as for each of the algorithms. Looking at MAE 5.1 results for the DrivingStereo dataset, one can see that Horn Schunck outperformed the other algorithms in both measurements. For both evaluation methods, Gabor Jets has better results than StereoBM but worse results than Horn Schunck and Liu. Comparing the images for which Gabor Jets and Horn Schunck had both highest and lowest MAE values, it can be observed that Gabor Jets performed best on an image with many different cars in it 5.1a and worst on a calmer image mainly consisting of an almost empty road5.1b. This is because Gabor Jets work best with the edges of the objects. The opposite case is witnessed for the Horn Schunck algorithm, with the best MAE being achieved for a calmer image with only two cars in it 5.1a and the worst for an image with more traffic and more variations in texture 5.1b.

For the BMP, the table 5.2 shows the best and worst results for all algorithms. In 5.3 the images where the disparity calculations with Gabor Jets resulted in the highest percentage of bad matched pixels are shown. All of the images consist primarily of an empty road. This does not offer a lot of edges or textures for the Gabor Jets.

With the MRE evaluation, Liu had the lowest mean and distribution. Horn Schunck again had the best result for an image with only a few cars and prominent structures 5.4a, while both Liu

Table 5.1: MAE values on DrivingStereo

Name	Gabor Jets	Liu	Horn Schunck	StereoBM
54-37-143.png	**27.4668**	422.8741	15.4592	46.0471
51-42-424.png	52.3406	**118.8669**	61.8797	395.3169
41-56-697.png	67.3103	536.0222	**12.3092**	**22.1378**
44-13-691.png	**187.4997**	775.8986	172.7955	455.2240
49-07-908.png	102.6489	**1098.1899**	105.7062	778.03
55-00-470.png	157.1728	1077.8844	**195.6546**	505.17
51-49-525.png	142.4025	689.4080	132.5607	**1112.1928**

(a) Gabor Jet, best MAE value

(b) Gabor Jet, worst MAE value

(a) Horn Schunck, best MAE value

(b) Horn Schunck, best MAE value

Table 5.2: BMP values on DrivingStereo

Name	Gabor Jets	Liu	Horn Schunck	StereoBM
54-41-453.png	**0.0668**	0.0729	0.0643	0.2311
54-41-053.png	0.0746	**0.0609**	0.0662	0.2349
41-56-697.png	0.1497	0.17021	**0.0249**	0.3561
49-09-309.png	0.1942	0.2853	0.1832	**0.1245**
55-33-802.png	**0.4380**	0.2807	0.2003	0.2965
55-39-508.png	0.2690	**0.34124**	0.1934	0.2894
44-13-691.png	0.4267	0.2738	**0.2476**	0.3102
41-53-396.png	0.1390	0.1967	0.0882	**0.4997**

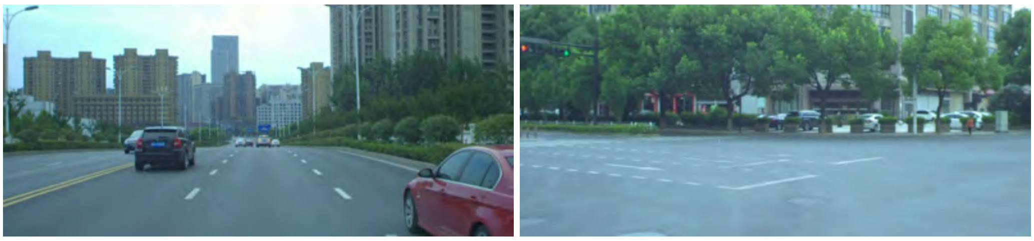
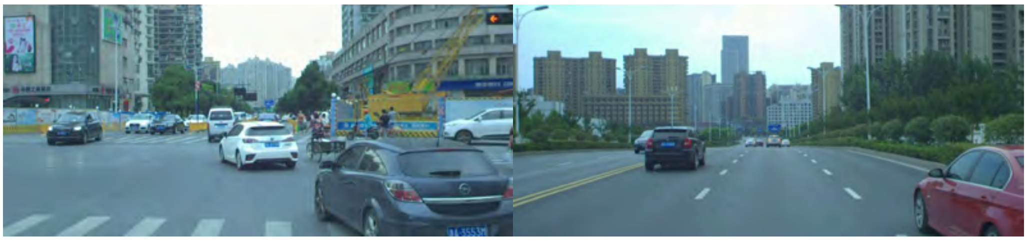


Figure 5.1: On the left is image 41-56-697.png, which is the image on which the StereoBM algorithm performed the best on. On the right is the image 51-49-525.png for which the StereoBM algorithm had the largest MAE.



(a) Liu,best BMP

(b) Liu,best BMP

Figure 5.2: Images with the best BMP for Liu and Horn Schunck

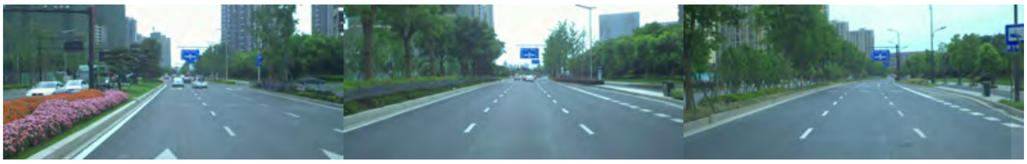


Figure 5.3: Images for which Gabro Jets has the highest BMP

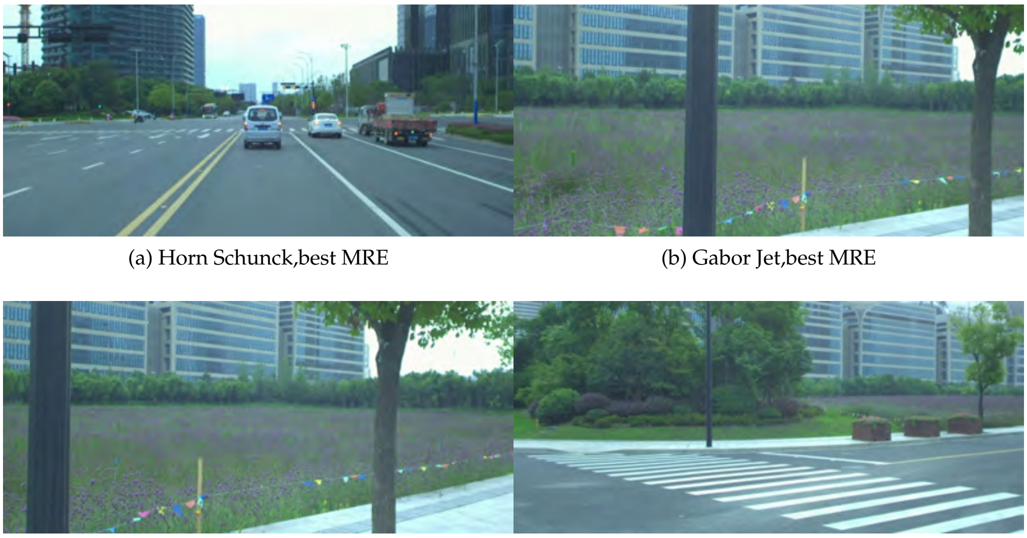
5.4c and Gabor Jets 5.4b performed the best on images consisting of more than half of a flower meadow. For Liu, this is likely because it repeatedly looks at brightness and smoothness, which works better on images with a few distinctive areas. For Gabor Jets it can be explained by the very textured nature of the meadow, unlike an empty street, a meadow has many structures that can be considered.

Both Gabor Jets and Liu have the highest MRE value for artL 5.5a and the lowest MRE for shelves 5.5b . Horn Schunck has the lowest MRE value for Playroom 5.5d and the highest for Motorcycle. StereoBM got the lowest result for Vintage 5.5c and the highest for artL 5.5a. All algorithms besides Horn Schunck had the lowest MRE value for artL, this is likely because it is a very clearly divided image. It has a lot of contrast and large objects, which leads to both clean edges, which are good for the performance of Gabor Jets and the large differences in colour, which seems to be beneficial for Liu and StereoBM. Horn Schunck performed the best on the image with smaller areas, yet still an image with a lot of contrast and edges.

For the Middlebury dataset, the BMP values seem unfitting. The BMP is generally very high

Table 5.3: MRE values on DrivingStereo

Name	Horn Schunck	Gabor Jets	Liu	StereoBM
12-06-908.png	**0.0174**	0.0835	0.005	0.2479
18-51-021.png	0.0394	0.0452	**0.0031**	0.3925
16-23-480.png	0.2572	0.265	0.2056	**0.085**
18-50-119.png	0.0409	**0.0417**	0.0084	0.4479
25-42-228.png	**0.3023**	0.2994	**0.2638**	0.22
29-18-002.png	0.2511	**0.4555**	0.1722	0.3583
23-59-319.png	0.0803	0.1168	0.0380	**0.6863**



(a) Horn Schunck,best MRE

(b) Gabor Jet,best MRE

(c) Liu,best MRE

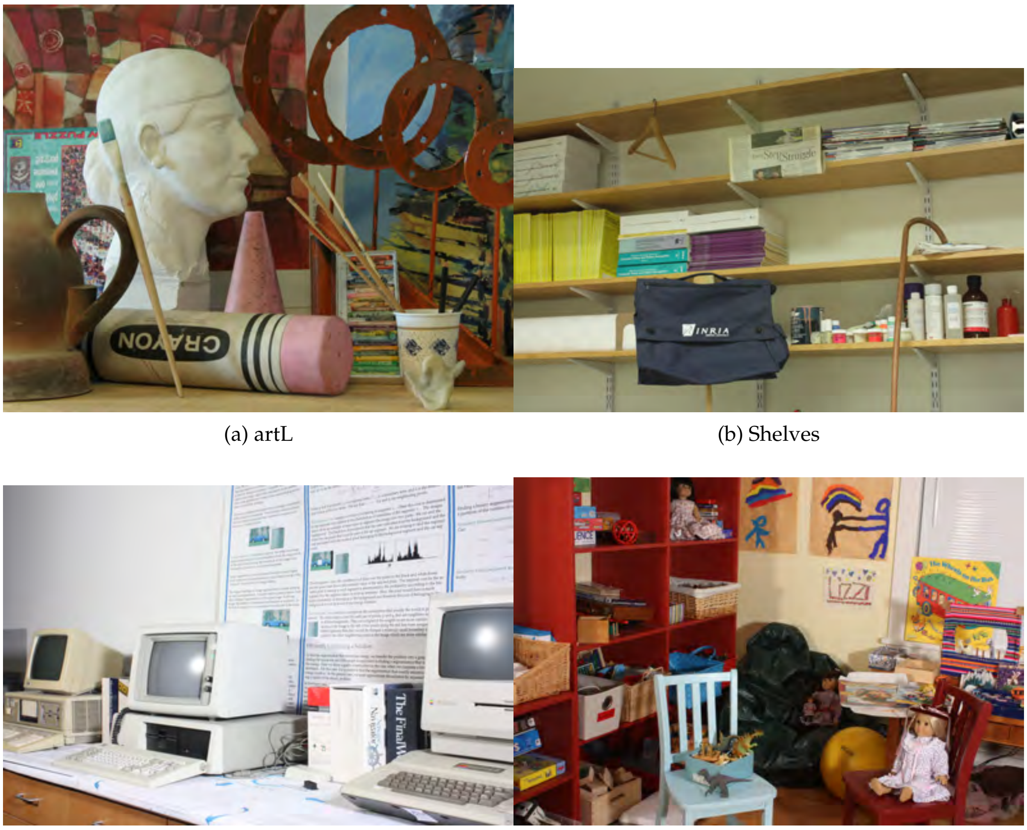
(d) StereoBM,best MRE

Figure 5.4: Images for which the respective algorithms have the lowest MRE

and, despite having good results for the MAE and MRE calculations, the Horn Schunck algorithm gets 99.5 % bad matches. It is likely that there is an error in the calculations that has not been found yet.

If the results from the DrivingStereo and the Middlebury datasets are compared, one can see that the block matching algorithm from StereoBM outperformed the other approaches with the Middlebury images. These images are constructed and consist of multiple objects, while the DrivingStere images are real-life images that often have large parts where not a lot goes on, like empty streets and meadows or bushes. In image 5.6a one can observe that Gabor Jets estimated the most disparities around the object, which stands out in front of the background. The Horn Schunck algorithm performed second best on both DrivingStereo and Middlebury.

While Gabor Jets performed better on the DrivingStereo dataset, StereoBM performed better on the Middlebury dataset. The type of images can explain the difference in performance between DrivingStereo and Middlebury. While the DrivingStereo dataset has been captured in real traffic, the images for the Middlebury dataset have been constructed under studio conditions. Because of this, the images in the Middlebury dataset consist of mainly a few items in a closed room, while the images DrivingStereo dataset show scenes where many objects are further from the lens. There is also a difference in the capture of the ground truths for both datasets. While DrivingStereo uses Lidar, Middlebury uses structured light. Because of this difference, the ground truths for Middlebury contain information for more pixels of the image.



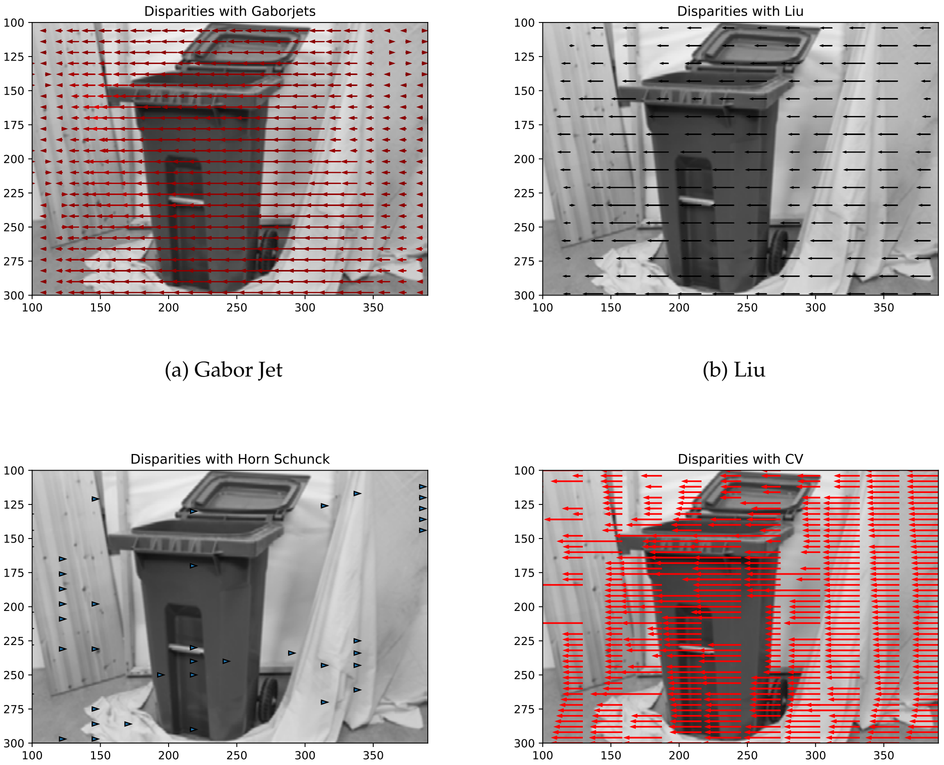
(a) artL

(b) Shelves

(c) Vintage

(d) Playroom

Figure 5.5: Images with the highest and lowest values on Middlebury



(a) Gabor Jet

(b) Liu

(c) Horn Schunck

(d) StereoBM

Figure 5.6: Disparity maps for the image Recycle.

Chapter 6

Conclusion

As there are many possible applications for optical flow, more efficient algorithms for calculating stereo matching are needed. In this thesis, the use of Gabor Jets for stereo matching was analysed using images from two datasets, DrivingStereo and Middlebury. To optimize the results all images are preprocessed. They are first converted to black and white, as some of the algorithms can not process coloured images, then rescaled to shorten the calculation time, and finally a border of 100 black pixel is added around the image to avoid inaccuracies due to wrap around. By comparing the results of Gabor Jets to the results of Horn Schunck, Liu, and StereoBM, this thesis showed that Gabor Jets could be an alternative for disparity estimation. On lower-resolution images, Gabor Jets achieve times that can keep up with the other algorithms. Considering accuracy Gabor Jets performed well, especially for images from the DrivingStereo dataset with a lot of traffic.

For a more complete assessment it would be interesting to compare Gabor Jets' results to the results of machine learning approaches. It would also be interesting to calculate the disparities in all directions, not only horizontally. Further the application for calculating the optical flow in video sequences would be interesting. Finally, the presented results show that Gabor Jets can be considered an alternate method for stereo matching. Especially for working with traffic images, Gabor Jets could be an interesting approach.

Bibliography

Anjos, A., El Shafey, L., Wallace, R., Günther, M., McCool, C., and Marcel, S. (2012). Bob: a free signal processing and machine learning toolbox for researchers. In *20th ACM Conference on Multimedia Systems (ACMMM), Nara, Japan*. ACM Press.

Barina, D. (2016). Gabor wavelets in image processing. *CoRR*, abs/1602.03308.

Bradski, G. (2000). The OpenCV Library. *Dr. Dobb's Journal of Software Tools*.

Buhmann, Lange, and von der Malsburg (1989). Distortion invariant object recognition by matching hierarchically labeled graphs. In *International 1989 Joint Conference on Neural Networks*, pages 155–159 vol.1.

Cabezas, I., Padilla, V., and Trujillo, M. (2012). BMPRE: An error measure for evaluating disparity maps. In *2012 IEEE 11th International Conference on Signal Processing*, volume 2, pages 1051–1055.

Daugman, J. G. (1985). Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters. *Journal of the Optical Society of America A, volume 2 (7)*, page 1160–1169.

De Jong, D. B., Paredes-Valles, F., and De Croon, G. C. H. E. (2021). How do neural networks estimate optical flow a neuropsychology-inspired study. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, page 1–1.

Gabor, D. (1946). Theory of communication. *Journal of the Institution of Electrical Engineers-Part III: Radio and Communication Engineering*, 93(26):429–441.

Günther, M., Haufe, D., and Würtz, R. P. (2012). Face recognition with disparity corrected Gabor phase differences. In Villa, A. E. P., Duch, W., Érdi, P., Masulli, F., and Palm, G., editors, *Artificial Neural Networks and Machine Learning*, volume 7552 of *Lecture Notes in Computer Science*, pages 411–418. Springer.

Hamid, M. S., Manap, N. A., Hamzah, R. A., and Kadmin, A. F. (2020). Stereo matching algorithm based on deep learning: A survey. *Journal of King Saud University - Computer and Information Sciences*.

Hirschmuller, H. (2008). Stereo processing by semiglobal matching and mutual information. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(2):328–341.

Horn, B. and Schunck, B. (1981). Determining optical flow. *Artificial Intelligence*, 17:185–203.

Hosseini, S. and Cho, N. I. (2019). Gf-capsnet: Using gabor jet and capsule networks for facial age, gender, and expression recognition. In *2019 14th IEEE International Conference on Automatic Face Gesture Recognition (FG 2019)*, pages 1–8.

Jordan, M. I., editor (1999). *Learning in Graphical Models*. MIT Press, Cambridge, MA, USA.

Ke, Q., Liu, J., Bennamoun, M., An, S., Sohel, F., and Boussaid, F. (2018). Chapter 5 - computer vision for human–machine interaction. In Leo, M. and Farinella, G. M., editors, *Computer Vision for Assistive Healthcare*, Computer Vision and Pattern Recognition, pages 127–145. Academic Press.

Kim, W.-S., Lee, D.-H., Kim, Y.-J., Kim, T., Lee, W.-S., and Choi, C.-H. (2021). Stereo-vision-based crop height estimation for agricultural robots. *Computers and Electronics in Agriculture*, 181:105937.

Lai, H.-Y., Tsai, Y.-H., and Chiu, W.-C. (2019). Bridging stereo matching and optical flow via spatiotemporal correspondence.

Lappe, M. (2009). Optic flow. In Binder, M. D., Hirokawa, N., and Windhorst, U., editors, *Encyclopedia of Neuroscience*, pages 3035–3039, Berlin, Heidelberg. Springer Berlin Heidelberg.

Li, Y. and Ibanez-Guzman, J. (2020). Lidar for autonomous driving: The principles, challenges, and trends for automotive lidar and perception systems. *IEEE Signal Processing Magazine*, 37(4):50–61.

Liu, C. (2009). *Beyond Pixels: Exploring New Representations and Applications for Motion Analysis*. Ph.D. Thesis, Massachusetts Institute of Technology.

Lucas, B. and Kanade, T. (1981). An iterative image registration technique with an application to stereo vision. In *IJCAI*, volume 81.

Malathi, T. and Bhuyan, M. (2015). Estimation of disparity map of stereo image pairs using spatial domain local gabor wavelet. *IET Computer Vision*, 9.

Orozco, R., Loscos, C., Martin, I., and Artusi, A. (2017). Chapter 3 - hdr multiview image sequence generation: Toward 3d hdr video. In Chalmers, A., Campisi, P., Shirley, P., and Olaizola, I. G., editors, *High Dynamic Range Video*, pages 61–86. Academic Press.

Poggi, M., Kim, S., Tosi, F., Kim, S., Aleotti, F., Min, D., Sohn, K., and Mattoccia, S. (2021). On the confidence of stereo matching in a deep-learning era: a quantitative evaluation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. Publisher Copyright: IEEE.

Radgui, A., Demonceaux, C., Mouaddib, E., Rziza, M., and Aboutajdine, D. (2008). An adapted lucas-kanade's method for optical flow estimation in catadioptric images. In *In OMNIVIS'2008, the Eighth Workshop on Omnidirectional Vision, Camera Networks and Non-classical Cameras, in conjunction with ECCV 2008*, pages 0–0, Marseille, France.

Raza, S. e. A., Prince, G., Clarkson, J., and Rajpoot, N. (2015). Automatic detection of diseased tomato plants using thermal and stereo visible light images. *PLoS ONE*, 10:e0123262.

Sanger, T. (1988). Stereo disparity computation using gabor filters. *Biological Cybernetics*, 59:405–418.

Scharstein, D., Hirschmüller, H., Kitajima, Y., Krathwohl, G., Nesic, N., Wang, X., and Westling, P. (2014). High-resolution stereo datasets with subpixel-accurate ground truth. In *GCPR*.

Schöps, T., Schönberger, J. L., Galliani, S., Sattler, T., Schindler, K., Pollefeys, M., and Geiger, A. (2017). A multi-view stereo benchmark with high-resolution images and multi-camera videos. In *Conference on Computer Vision and Pattern Recognition (CVPR)*.

Szeliski, R. (2011). Computer vision algorithms and applications.

Tai Sing Lee (1996). Image representation using 2d gabor wavelets. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(10):959–971.

Urtasun, A. G. L. (2012). Are we ready for autonomous driving? the kitti vision benchmark suite. In *Conference on Computer Vision and Pattern Recognition (CVPR)*.

Yaakob, R., Aryanfar, A., Halin, A. A., and Sulaiman, N. (2013). A comparison of different block matching algorithms for motion estimation. *Procedia Technology*, 11:199–205. 4th International Conference on Electrical Engineering and Informatics, ICEEI 2013.

Yang, G., Song, X., Huang, C., Deng, Z., Shi, J., and Zhou, B. (2019). Drivingstereo: A large-scale dataset for stereo matching in autonomous driving scenarios. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Zhang, F., Prisacariu, V. A., Yang, R., and Torr, P. H. S. (2019). Ga-net: Guided aggregation net for end-to-end stereo matching. *CoRR*, abs/1904.06587.