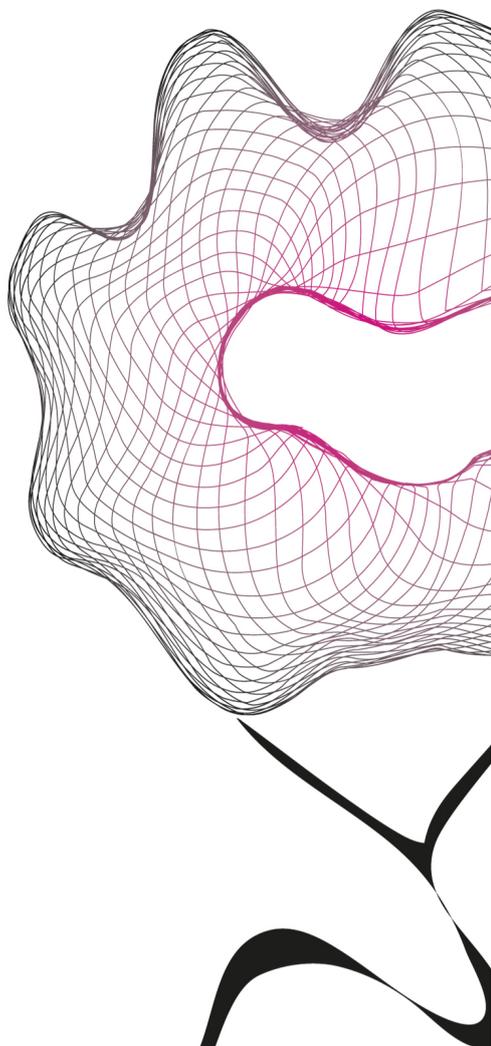


UNIVERSITY OF TWENTE.

Signals & Systems



LIKELIHOOD FUNCTIONS FOR WINDOW-BASED STEREO VISION

M.Sc. Thesis

Robert Vonk

University of Twente
Department of Electrical Engineering,
Mathematics & Computer Science (EEMCS)
Signals & Systems Group (SAS)
P.O. Box 217
7500AE Enschede
The Netherlands

Report Number:	SAS2011-20
Report Date:	January 13, 2012
Thesis Committee:	Prof. dr. ir. C.H. Slump Dr. ir. F. van der Heijden Dr. ir. L.J. Spreeuwers R. Reilink, MSc

Likelihood Functions for Window-based Stereo Vision

Master of Science Thesis

Submitted in partial fulfillment of the requirements for the degree of Master of Science

Department of Electrical Engineering,
Mathematics & Computer Science (EEMCS)
Signals & Systems Group (SAS)
University of Twente

Robert Vincent Vonk, BSc
January 13, 2012

Student Number: 0020184
Report Number: SAS2011-20
Thesis Committee: Prof. dr. ir. C.H. Slump
Dr. ir. F. van der Heijden
Dr. ir. L.J. Spreeuwens
R. Reilink, MSc

Abstract

The biological process of stereopsis — the brain is able to perceive depth from the information of two eyes — inspired researchers to bring this ability to computers and robotics. As this proves to be a complex task it led to the introduction of a whole new field: Computer Vision. Two or more cameras at different positions take pictures of the same scene. A computer compares these images to determine the shift of local features. The shift (disparity) of an object in the images is used to calculate the distance.

Most algorithms use a similarity measure to compute the disparity of local features between images. The quality of the similarity measure determines the potential of the algorithm. This research concentrates on the earlier work of Damjanović, Van der Heijden, and Spreeuwers, who took a different approach. They introduced a new likelihood function for window-based stereo matching, based on a sound probabilistic model to cope with unknown textures, uncertain gain factors, uncertain offsets, and correlated noise.

The derivation of the likelihood function is the first part. The likelihood function is obtained by marginalization of the texture and the gains. In the paper this research is based on, a solution is obtained by a few approximations. However, we show that one approximation is not allowed due to an error in the solution for the first integration step. Through several attempts it is tried to bring a (partial) solution within reach. Also, it is shown that a generalization for n -view vision does not complicate the final integration step further.

The main goal of the proposed likelihood function is to outperform the normalized cross correlation (NCC) and the sum of squared differences (SSD). A simplification of the likelihood function (in which the gains are left out) results in a metric with the Mahalanobis distance at its basis compared to the Euclidean distance for the SSD. Information within the windows (e.g. distortions, occlusions, and importance of pixels) is exploited to train the Mahalanobis distance with an optimal covariance matrix. Experiments show that the simplified likelihood function decreases the number of errors for difficult regions in the scene.

In recent research, the focus lies primarily on post-processing such as belief propagation. However, one of the main findings of this research is that a good similarity measure such as the Mahalanobis distance decreases the number of errors in stereo correspondence for difficult regions. The correct matches near occlusions and discontinuities of the disparity map provide important information that can be directly used within a probabilistic framework (HMM/BP). Although an analytic solution for the complete likelihood function remains unsolved, progress has been made. Alternative methods are suggested that could lead to a proper analytic solution for the proposed probabilistic model.

Samenvatting

Het biologische proces van stereopsis — de hersenen gebruiken informatie van beide ogen om diepte te zien — heeft onderzoekers geïnspireerd om deze kunde naar computers en robotica te brengen. Dit bleek een lastige uitdaging te zijn, waarmee een nieuwe onderzoeksrichting was geboren: Computer Vision. Twee of meer verschillend gepositioneerde camera's nemen foto's van een scène. Een computer vergelijkt deze foto's om te bepalen wat de verschuiving van lokale kenmerken is. Met de verschuiving van een voorwerp kan de afstand worden berekend.

De meeste algoritmen gebruiken een *similarity measure* om de verschuiving van kernmerken in beide afbeeldingen te bepalen. De kwaliteit van de *similarity measure* bepaalt het potentieel van het algoritme. Dit onderzoek bouwt voort op het werk van Damjanović, van der Heijden en Spreewers over een nieuwe aanpak. Zij hebben een nieuwe aanemelijkheidsfunctie geïntroduceerd voor window-based stereo matching, gebaseerd op een degelijk statistisch model dat rekening houdt met onbekende textuur, onbekende versterkingsfactoren, onbekende afwijkingen en gecorreleerde ruis.

De afleiding van de aanemelijkheidsfunctie is het eerte onderdeel. De functie is verkregen door marginalisatie van de textuur en de versterkingen. De oplossing is in het artikel verkregen door enkele benaderingen toe te passen. Hier is echter in de afleiding gebleken dat één vereenvoudiging niet kan worden toegepast door een fout in de eerste integratie. Verschillende pogingen zijn gedaan om een (deel)oplossing binnen bereik te krijgen. Het is tevens aangetoond dat een generalisatie voor meer dan twee camera's de laatste integratie niet ingewikkelder maakt.

Het hoofddoel voor de voorgestelde aanemelijkheidsfunctie is het verbeteren van de prestaties ten opzichte van de NCC en de SSD. Een vereenvoudiging van de aanemelijkheidsfunctie (zonder versterkingsfactoren) geeft een metriek met de Mahalanobis-afstand in de basis; dit in vergelijking met de Euclidische afstand voor de SSD. Informatie in de windows (zoals vervormingen, oclusies en relevantie van de verschillende pixels) is gebruikt om de Mahalanobis-afstand te trainen voor een optimale covariantie matrix. Uit experimenten blijkt dat het aantal fouten in moeilijke gebieden afneemt met de vereenvoudigde aanemelijkheidsfunctie.

De focus ligt in recent onderzoek vooral op nabewerking, zoals belief propagation. De hoofdvindingen van dit onderzoek tonen aan dat een goede *similarity measure* zoals de Mahalanobis-afstand het aantal fouten doet afnemen voor lastige gebieden. De nieuwe correcte verschuivingen in de disparity map komen vooral voor bij oclusies en discontinuïteiten. Deze informatie kan direct worden gebruikt in statistische raamwerken (HMM/BP). Helaas blijft de complete oplossing voor de aanemelijkheidsfunctie nog onopgelost, maar er is voortgang geboekt. Alternatieve methoden zijn aangedragen die kunnen leiden tot een goede analytische oplossing voor het voorgestelde statistische model.

Acknowledgements

This thesis is the final work of my master's project in partial fulfillment of the requirements for a Master's degree at the University of Twente. It serves as documentation for my work realised in the field of computer stereo vision.

Although supervisors should generally not be thanked in the acknowledgements, I have the strong urge to do so anyway. I can imagine that I have not always been an easy student to cope with. I would like to express my appreciation to my advisors: Ferdi and Luuk. They have opened my eyes to a new and interesting field of research, motivated me in difficult times, and provided me with a constant flow of ideas and advice. I would like to thank Sanja for her ideas, explanations and time.

For years, I have tried to find the right balance between study and work. I would like to thank Dio for his patience, support, and selflessness; especially during this research project. Also, it was of great help that I could abuse to server grid for countless cycles to perform my experiments.

I thank my friends for all the recreational activities and kind words that kept me motivated to keep working. I would like to thank Marijn in particular for his advice, friendship, and the discussions we have had over the years. Also, I would like to thank Leen for helping me in difficult times.

Finally, I thank my parents and sister. Their everlasting support has brought me to where I am today. Without them, I would never have come this far. And therefor I can never thank them enough.

Robert Vonk
Enschede

Contents

Abstract	5
Samenvatting	7
Acknowledgements	9
List of Figures	13
List of Tables	15
1 Introduction	19
1.1 Motivation	19
1.2 What is stereo vision	20
1.3 Problem definition	21
1.4 Outline	23
2 New likelihood function for stereo correspondence	25
2.1 Introduction	25
2.2 Derivation of the likelihood function	28
2.2.1 Gaussian Integral	29
2.2.2 Marginalization of the unknown texture	30
2.2.3 Marginalization of the camera gains	35
2.3 Approximation by power series	36
2.3.1 Computational complexity	37
2.3.2 Approximation results	37
2.4 Suggestion for alternative analytic solution	39
2.4.1 Polar coordinates	40
2.5 Generalization for multiple cameras	41
2.6 Discussion	43

3	Mahalanobis versus Euclidean distance	45
3.1	Mahalanobis likelihood derivation	46
3.2	Method of evaluation	50
3.2.1	Data selection	51
3.2.2	Window size and image scaling	52
3.2.3	Occluded regions	53
3.2.4	Experiment	54
3.3	Results	58
3.3.1	Preliminary: covariance matrix	58
3.3.2	Mahalanobis distance vs. Euclidean distance	61
3.3.3	Results for self-training	63
3.3.4	Results for occluded regions	65
3.4	Discussion	65
4	Conclusions and Discussion	67
4.1	Research questions	67
4.2	Recommendations	69
A	Proofs and Reference	71
A.1	Integral of a Gaussian function	71
A.2	Multidimensional Gaussian integral	72
A.3	Integral of a multi-dimensional Gaussian function	73
A.4	Proof symmetric positive-semidefinite covariance	77
A.5	Vector and matrix properties	78
A.6	Fubini theorem	79
A.7	Integration by substitution for multiple variables	79
A.8	Cauchy's Residue Theorem	80
B	List of project files	81
B.1	Data and file structure	81
B.2	Scripts and programs	82
B.3	Image dataset	82
C	Dataset	85
D	Supplemental results	89
D.1	Performance differences	91
D.2	Eigenvectors	93
E	Taylor expansions and marginalizations in Maple	97
E.1	Third order approximation	97
E.2	Three cameras	98
	Bibliography	101

List of Figures

1.1	Computer vision setup with two cameras	20
1.2	Image rectification	21
2.1	Marginalization on 1D-taylor approximations (n=2)	38
2.2	Marginalization on 1D-taylor approximations (n=16)	39
2.3	Marginalization on a 2D seventh-order taylor expansion	39
3.1	Stereo correspondance performance for different window sizes and image scales.	52
3.2	Correct and incorrect correspondances for the Euclidean distance	53
3.3	The precision matrix built from residuals of the Bowling2 dataset.	58
3.4	Eigenvalues of the covariance matrix built from the available datasets	60
3.5	Eigenvectors of the covariance matrix (Bowling2)	61
3.6	Number of errors for a regularized covariance matrix normalized to the Euclidean distance.	62
3.7	Difference for the Bowling2 dataset between the Mahalanobis distance for optimal regularization and the Euclidean distance.	63
3.8	Number of errors for the covariance matrix trained on the test image itself. The window size is 11-by-11 applied on the half size images.	64
3.9	Comparison of different covariance matrices for 11-by-11 windows evaluated on half size images of the Middlebury dataset.	65
A.1	Contour integral around poles z_i	80
C.1	Images selected from the Middlebury dataset	87
C.2	Overview of the image regions of the selected datasets	88
D.1	Performance difference between left: the Mahalanobis method vs. the Euclidean method; right: leave-one-out training vs. self-training	93
D.2	Eigenvectors of the covariance matrix for the Aloe dataset (grayscale)	94
D.3	Eigenvectors of the covariance matrix for the Bowling2 dataset (color)	95

List of Tables

2.1	Number of coefficients (required multiplications per window).	37
3.1	Difference between leave-one-out and self-training.	64
B.1	Matlab scripts used to run the experiments	82
D.1	Incorrect pixel disparities	89
D.2	Incorrect pixel disparities with low energy windows discarded	90
D.3	Incorrect pixel disparities for images processed with covariances matrices trained on the dataset itself	90

Symbols and Abbreviations

σ Standard deviation

μ Mean value

\mathbf{Im}_k Image for camera viewpoint k

\mathbf{z}_k Serialized vector for a 2D window in image \mathbf{Im}_n for viewpoint k

$p(\mathbf{z}_1, \dots, \mathbf{z}_k | \dots)$ Similarity measure between k windows given ...

w The window base ($n = w^2$)

n The window size (the number of pixels within the window)

k The number of viewpoints or cameras

α_i Gain of camera i

\mathbf{F} The fundamental matrix

\mathbf{C} or Σ The covariance matrix

\mathbf{P} The precision matrix

\mathbf{J} The jacobian matrix with $\frac{\partial y_{ij}}{\partial x_{ij}}$

λ Eigenvalue

Λ Eigenvalue matrix with all eigenvalues on the diagonal

\mathbf{T} Eigenvector matrix

\mathbf{S}^{ij} The single-entry matrix $\mathbb{R}^{n \times n}$ where only the (i, j) -th entry is non-zero: one

\mathbf{I}_n The $n \times n$ identity matrix

NCC Normalized Cross Correlation

SAD Sum of Absolute Differences

SSD Sum of Squared Differences

WTA Winner Takes All

CBA Constant Brightness Assumption

pdf Probability density function

Introduction

It has been known for a long time that animals are able to perceive depth from a scene when it is viewed with two eyes. Leonardo da Vinci realized that objects at different distances from the eyes project images in the left and the right eye that differ in their horizontal positions. The difference in horizontal position in both views is referred to as binocular disparity. Leonardo da Vinci used his analysis of stereo vision to conclude that it is impossible for a painter to portay a realistic description of depth on a two-dimensional canvas. Stereopsis was first explained scientifically by Charles Wheatstone with his significant paper in 1838: "... the mind perceives an object of three dimensions by means of the two dissimilar pictures projected by it on the two retinae".

In the 1970's, with the rise of computers and digital imaging devices, experts in the field of Artificial Intelligence thought that making a computer see would be at the level of difficulty of a summer student's project [8]. However, forty years later an entire field called Computer Vision has emerged as a discipline itself. It appeared that visual perception is far more complex in animals and humans then was first thought. Researchers have made significant progress in the field of stereo vision; however, there is still much room for improvement.

1.1 Motivation

Computer stereo vision is an active field in which a lot of significant advances have been made in recent years. However, computer alhorithms are still not on par with the biological process of Stereopsis. The goal of this thesis is to focus on a small though important part of stereo vision to improve the overall performance of depth perception by computers. Computer stereo vision is the process of extracting depth information from digital images.

An essential step in stereo vision is to define a similarity measure for local regions between images. A likelihood function is defined and used to compute the probability of a point in the reference image to a different position in the other image(s). The most likely difference in position, or disparity, is inversely proportional to the distance to the object in the scene. The main objective of this project is to improve the similarity measure

for a stereo match through a better probabilistic model. A good similarity measure is very important for the 3D reconstruction as it provides the fundamental information for disparity optimization algorithms, and consequently the resulting disparity map.

This research project is based on the paper of Damjanović et al.: “*A new Likelihood Function for Stereo Matching - How to Achieve Invariance to Unknown Texture, Gains and Offsets?*” [4]. The new likelihood function is part of the PhD-research of Sanja Damjanović. In this paper, it was shown that a likelihood function based on a sound probabilistic model outperforms both the SSD and the NCC, and can be used within a probabilistic framework. Recent mainstream research is focused on methods such as Belief Propagation. We hope to provide a contribution to computer stereo vision by providing a better similarity measure.

Hypothesis: The similarity measure benefits from a better probabilistic model based on ground-truth training to improve block matching correctness, and consequently depth perception.

1.2 What is stereo vision

Computer stereo vision is similar to human binocular vision. Two cameras are placed at slightly different positions, and both cameras make digital images of the same scene. Objects in the scene vary slightly in position in the projections of the left and the right image. The distance of the object with respect to the cameras determines the shift in position (disparity). Nearby objects have large disparities, and objects far away have a very small disparities. The disparities can be used to reconstruct a depth map with 3D information about the scene. This model is visualized in Figure 1.1.

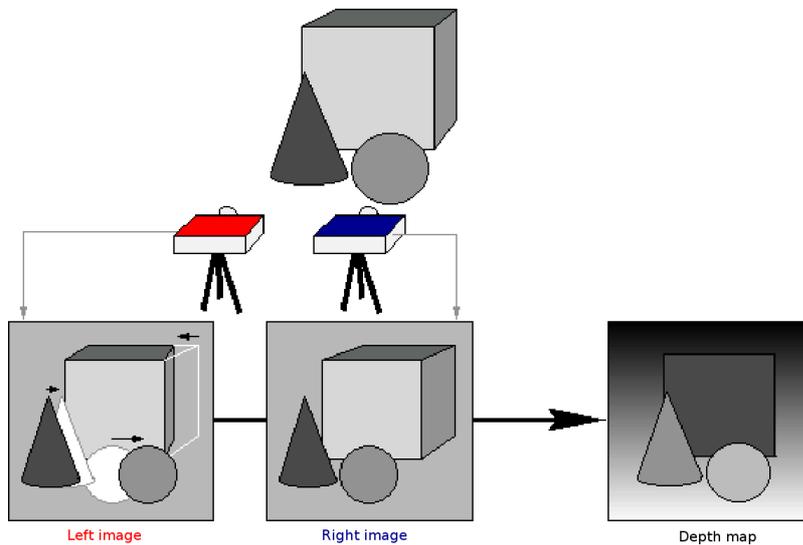


Figure 1.1: Computer vision setup with two cameras¹

The algorithm to extract depth information from the digital images can be summarized in four important steps. First, the digital images have to be repaired to remove all distortions. For example, optical systems of cameras often introduce barrel distortion. The images must be processed in such a way that the observed image is purely projectional. Second, the problem has to be reduced to one dimension. Image rectification is the transformation process of two images on a common plane. The images are transformed into a standard coordinate system. The transformation process is shown in Figure 1.2. A very good explanation of image rectification is given in [8, pp 242].

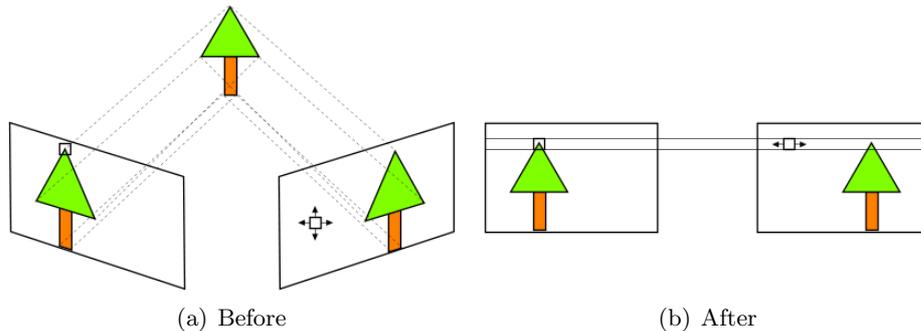


Figure 1.2: Image rectification²

In the third step a disparity map is computed from the local information between two images. This process is called the stereo correspondence problem. This research focuses on this part of the stereo vision algorithm. In this project, similarity measures are used to determine what the most likely disparity of local features is. In the final and fourth step, the disparity map is converted to a depth map.

1.3 Problem definition

Stereo correspondence is a difficult problem that suffers from several effects that generally lower performance of the similarity measure. Classical methods for block matching, such as the Sum of Squared Differences (SSD), have difficulty to adapt to varying camera gains and offsets. Also, distortion of surfaces as observed from the different viewpoints causes dilation and/or contraction of the local regions around the point of interest. This distortion has different properties for the outer pixels of the windows as opposed to the center of the window. A more severe effect occurs when (parts of) objects are visible in only a subset of the projections. This effect is known as, dependent on the situation, occlusion or overreach and indicates discontinuity in the disparity map.

We suspect an improvement in matching performance is possible if a proper likelihood function is chosen that partially takes these effects into account. Based on the earlier work of Damjanović, Van der Heijden, and Spreeuwens, we define a set of research

¹Image by Rolf Henkel, University of Bremen

²Image by Allan Ortiz

questions. The research questions are answered in this thesis and are meant to define a path to contribute to the goal of a better matching performance.

Research questions:

- Q1 How can the algorithm as introduced by Damjanović et al. be improved, taking into account the complication of the analytical derivation?

The solution in the paper uses approximations to obtain an analytical solution for the likelihood function. Also, the results in the paper are obtained by experiments that use the Euclidean distance as metric. However, for the probabilistic model, a properly trained covariance matrix is assumed. Use of the covariance matrix generalizes the Euclidean distance to a Mahalanobis distance.

- (Q1a) Are approximations sensible to obtain a solution for the likelihood function?

A complete analytical solution for the statistical model implies a very complicated integral for the gains. Several terms are assumed to be constant during the derivation of the likelihood function. The influence of these assumptions is small for low-order terms; however, approximation of very high-order terms can result in significant errors. Also, it is always possible to integrate a Laurent expansion of a complicated function. Unfortunately, the result is always limited by the order of the expansion.

- (Q1b) How is the new likelihood function generalized for more than two camera views?

Two digital images contain the minimum amount of information to reconstruct the 3D information. Extra camera views supply additional information that can be exploited to obtain a better estimate.

- Q2 Does a simplified version of the likelihood function improve performance?

We are curious whether a proper covariance matrix improves the matching performance. For the simplified likelihood function, the unknown gains are omitted to inspect the effect the Mahalanobis distance

If the simplified likelihood function appears to be useful:

- (Q2a) How significant is the reduction of errors in the stereo correspondence?

The Mahalanobis distance is not free in terms of computational power. Every improvement in performance is good; however, the computational complexity is often an important factor in design decisions.

- (Q2b) Is it possible to improve the matching performance of the simplified likelihood function?

The covariance matrix is generated during a training stage of the algorithm. The chosen data determines the sensitivity of the covariance matrix for various effects.

1.4 Outline

The other chapters of this thesis are written to answer the research questions that were formulated in the problem definition. The contents of the thesis are divided in two important parts that each describe a phase in the research project.

Chapter two is written to describe the derivation of the likelihood function. The introductory section describes the statistical model that is used to obtain a new likelihood function. The derivation should be read as an extension to the paper that forms the basis of this research. The encountered problems and errors are described, as well as the complications for an analytical solution. An attempt has been made to approximate certain parts of the equation in Section 2.3. The chapter concludes with a suggestion for an alternative method and a short proof to show that a likelihood function for n -views is not necessarily more difficult to solve.

The third chapter describes the second part of the research project. In order to show that elements of the new likelihood function contribute to a better matching performance, it describes a simplification of the likelihood function. In the theoretical section, it appears that the simplification produces a monotonically decreasing function of the Mahalanobis distance. The experiment is meant to answer the research question concerning the performance of the Mahalanobis distance versus the Euclidean distance. Different methods to generate the covariance matrix come to pass to research the effect on the matching performance. The chapter concludes with results and conclusions.

The final and fourth chapter concludes the thesis with a discussion of the important findings of the research project. One section is devoted to the research questions that have been formulated in this section. Finally, a short summation of suggestions is given for future research.

New likelihood function for stereo correspondence

This chapter describes the derivation of a new likelihood function that was introduced in the paper: “*A new Likelihood Function for Stereo Matching - How to Achieve Invariance to Unknown Texture, Gains and Offsets?*” [4]. The new likelihood function is part of the PhD-research of Sanja Damjanović. The structure for the Section that describes the derivation of the likelihood function follows the paper, and takes different directions for the complications that have arisen in the search for an analytical solution.

First, an introduction of, and the motivation for the new likelihood function is given in Section 2.1. The derivation of the likelihood function that is based on the new statistical model is given in Section 2.2. Several mathematical theorems have to be proved to create the necessary tools for the derivation. The proofs for these theorems are given in Appendix A. Section 2.3 presents an attempt to solve the problem with power series approximations. In Section 2.4, a suggestion is given that could lead to an alternative analytical solution for the problem. The implications of a generalization for more than two camera views is given in Section 2.5. Finally, Section 2.6 concludes the chapter with a discussion of the findings and the consequences for the newly proposed likelihood function. Also, it discusses several suggestions for future research.

2.1 Introduction

Digital images are projections of a 3D world. Computer stereo vision uses several images obtained by cameras of known relative positions and orientations to extract 3D-information of a scene. A difficult part in this process is to find a good solution for the correspondence problem. Given a token in the left image, the problem is to find the matching token in the right image [5]. The solution to this problem gives the displacement (or disparity) between the tokens in both images. The disparity is inversely proportional to depth; hence, the token’s depth is computed as function of the disparity. The goal of this project is to investigate and improve similarity measures for pixel-based

stereo.

We consider stereo matching for a known camera geometry that operates on two or more camera views to produce a dense disparity map $d(x, y)$. For dense stereo matching, disparity for each pixel in the reference image is estimated [18]. It is assumed that all images are taken on a linear path with the optical axis perpendicular to the camera displacement. The optical axes of all cameras are parallel. The row-directions, i.e. the x -axes of all image planes are also parallel, and the positions of all cameras are on a line that is also parallel to the row-directions. Alternatively, a perfect camera alignment is obtained with image rectification that transforms the images into a standard coordinate system.

The correspondence between a pixel (x, y) in the reference image \mathbf{I}_1 and a pixel (x_2, y_2) in the matching image \mathbf{I}_2 is given by

$$x_2 = x + d(x, y), \quad y_2 = y, \quad (2.1)$$

For window-based stereo matching, a similarity measure is used to compare the contents of the windows around the candidate points. In the classical approach, disparities are estimated on an individual basis, point by point. This local method is known as the *Winner Takes All* (WTA): at each pixel the disparity is chosen with the lowest cost. However, modern algorithms often use semi-global or global optimization methods based on mutual information and approximation of a global smoothness constraint [9]. For example, popular methods that perform one-dimensional optimizations are the Viterbi algorithm and the forward-backward algorithm. Other dynamic programming algorithms such as belief propagation (BP) and graph cuts (GC) perform two-dimensional energy optimizations. All methods, be it local or (semi-)global, rely on a good similarity measure. The goal of this chapter is to improve the matching performance of the similarity measure by using a sound probabilistic model [4]. We expect that the matching performance can be improved even further if we incorporate more than two camera views in the likelihood function. An n -view extension of the likelihood function has more information available than a two-view likelihood function. However, the complexity increases significantly, because more combinations of images have to be compared to each other. This implies that every window combination has to be multiplied with the precision matrix, which is a computationally costly process.

The *Normalized Cross Correlation* (NCC) [7] is one of the first and still commonly used window-based matching techniques. Gains and offsets in the images are neutralized; however, NCC tends to blur depth discontinuities more than other similarity measures because outliers lead to high errors within the NCC calculation [13, 10]. The NCC is computed by:

$$p_{\text{NCC}}(\mathbf{I}_1, \mathbf{I}_2 \mid i, j, x, w) = C \sum_{k=-w}^w \sum_{l=-w}^w \frac{(\mathbf{I}_1(i+k, j+l) - \mu_1(i, j)) (\mathbf{I}_2(i+x+k, j+l) - \mu_2(i+x, j))}{\sigma_1(i, j) \sigma_2(i, j)}, \quad (2.2)$$

with a constant $C = \frac{1}{N-1}$ with $N = (2w+1)^2$. The mean μ_n and variance σ_n to compute the NCC are defined by:

$$\mu_n(i, j) = \frac{1}{N} \sum_{k=-w}^w \sum_{l=-w}^w \mathbf{Im}_n(i+k, j+l) \quad (2.3)$$

$$\sigma_n(i, j) = \sqrt{\frac{1}{N} \sum_{k=-w}^w \sum_{l=-w}^w (\mathbf{Im}_n(i+k, j+l) - \mu_n(i, j))^2}. \quad (2.4)$$

Here, i is the row index, j is the column index, l and k are both local window counters, and x is the horizontal disparity for which the NCC is applied.

In the 1996, both Cox et al. [3] and Belhumeur [1] introduced methods based on models within a Bayesian framework. The optimization criterion is expressed in terms of probability density functions. In the probabilistic approach to the stereo correspondence problem, The similarity measure is described as a likelihood function. It is the conditional probability density of the data given the disparities. The models introduced by Cox et al. [3] and Belhumeur [1] lead both to a monotonically decreasing function of the *Sum of Squared Differences* (SSD). Only the difference in likelihood is important. Therefore, the scaling constant of Belmuheur's model can be omitted. The likelihood function for the models of Belhumeur and Cox et al. can be expressed as:

$$p(\mathbf{z}_1, \mathbf{z}_2 | x) \propto \exp\left(-\frac{1}{4\sigma_n^2} \|\mathbf{z}_1 - \mathbf{z}_2\|^2\right), \quad (2.5)$$

where \mathbf{z}_k are the measurement vectors of the windows in the digital images \mathbf{Im}_k . The measurement vectors \mathbf{z}_k are a one-dimensional representations of the windows around the (candidate) points. The columns of the two-dimensional window contain w pixels each and are stacked in an n -dimensional measurement vector \mathbf{z} . Therefore, $n = w^2$. The disparity for which the probability function generates a likelihood is given by x .

A new likelihood function was introduced by Sanja Damjanović et al. in 2009 [4]. A sound probabilistic model is used to produce a likelihood function that copes with unknown textures, uncertain gain factors, uncertain offsets, and correlated noise. The goal of this research is to validate, analyze and generalize this likelihood function to present a better solution for the stereo correspondence problem than the classical methods such as the NCC and the SSD. The likelihood function allows similarity measures between two digital images; however, we would like to generalize this likelihood for more than two camera views, to three-view or n-view 3D-reconstruction. Unfortunately, the complexity of the assumed model complicates the analytical derivation of the probability density function. It was assumed that an analytical solution that satisfies the model was successfully derived, but, as will appear in this chapter, problems arise. This chapter presents partial solutions for the likelihood function, discusses approximations, and suggests alternative methods that could complete the analytical derivation in the future.

It appears that the simplified model in which we omit the camera gains and offsets reduces the likelihood function to a monotonically decreasing function of the Mahalanobis

distance. This function is the subject of the next Chapter, where it is derived as an extension of the likelihood function with the Euclidean distance as basis.

2.2 Derivation of the likelihood function

The proposed likelihood function by Sanja Damjanović [4] is based on an extended model that uses the same Bayesian approach as used by Cox et al. [3]. Stereo matching is usually an ill-posed problem due to occlusions, specularities and lack of texture [5]. Solving the stereo correspondence problem, therefore, requires that we impose certain assumptions on the matching process. The epipolar constraint transforms the stereo matching to a one-dimensional problem. This implies that matching points lie on corresponding epipolar lines. The second constraint, the constant brightness assumption (CBA), implies that surfaces in the scene are ideally diffuse without specular properties. The objects brightness is independent of the viewing angle (Lambertian illumination). Also, we assume that a point in one image matches at most one point in another image, and is called the uniqueness constraint.

The basic model assumes a system with two cameras that (indirectly) produces rectified digital images. The likelihood function uses two measurement vectors \mathbf{z}_1 and \mathbf{z}_2 that represent the image data that surrounds the two points in the images. The pixel intensities within the windows depend on the texture and the radiometric properties of the observed surface patch, on the illumination of the surface, and on the properties of the imaging device [4]. This model is defined by:

$$\mathbf{z}_k = \alpha_k \mathbf{s} + \mathbf{n}_l + \beta_k \mathbf{e}, \quad k \in \mathbb{N}. \quad (2.6)$$

In this model, \mathbf{s} is the result of mapping the texture on the surface of the two image planes. The camera gain factors are represented by α_k , and the offsets are β_k . Also, \mathbf{e} are unity vectors and \mathbf{n}_k are noise vectors that are assumed to be Gaussian and uncorrelated [4].

The expression for the likelihood function is obtained by several marginalization steps. The joint distribution is obtained by marginalizing the unknown texture and the camera gains out of the distribution. First, the probability density functions of \mathbf{z}_k is marginalized with respect to the unknown texture \mathbf{s} . This implies a multivariate integral, because the dimensionality of \mathbf{s} is defined by the window size n . The second step requires marginalization of the expression with respect to the camera gains α_k . The covariance matrix of the Gaussian is rewritten to include the white noise terms \mathbf{n}_l and the offsets β_k .

The derivation of the expression requires a few theorems to obtain a solution. The marginalization of the unknown texture \mathbf{s} implies a multivariate integral. It appears that the expression can be rewritten to a multivariate Gaussian function for which an analytical solution is known to exist (Appendix A.3). An essential part to complete the proof is the Gaussian integral. The method used in Section 2.2.1 is a useful method to simplify all sorts of Gaussian integrals with polar coordinates. It appears that this method to obtain a solution for the Gaussian integral also allows simplification of the

likelihood expression in a later stage (Section 2.4). The solution is well known, but supplied nonetheless to clarify the suggested method for an alternative solution.

First, the solution to the Gaussian integral is proved. This result is used in the next section to obtain a solution for the multivariate Gaussian integral. The theorem for the multivariate Gaussian function is used in Section 2.2.2 to solve the marginalization of the unknown texture \mathbf{s} . Finally, it is concluded in the last part that marginalization of the camera gains is very problematic and requires a different method. Unfortunately, this section does not conclude with an analytical solution.

2.2.1 Gaussian Integral

To satisfy the research goal of an improved likelihood function, the proposed likelihood function will require marginalization of the conditional probabilities. This implies the integration of the chosen normal distributions for certain assumed a-priori variables. Therefore, an analytical solution for the improper integral over the Gaussian function is required¹. Also, it will be shown in Section 2.4 that same method and transformation can be used (as the first step) to solve the integral of Equation 2.46 that remains unsolved in Section 2.2.3.

The Gaussian integral taken from minus infinity to infinity can be rewritten to a product of two integrals. These two integrals can then be merged to a double integral with bivariate exponent:

$$\int_{-\infty}^{\infty} \exp(-x^2) dx = \sqrt{\left(\int_{-\infty}^{\infty} \exp(-x^2) dx\right) \left(\int_{-\infty}^{\infty} \exp(-x^2) dx\right)} \quad (2.7)$$

$$= \sqrt{\left(\int_{-\infty}^{\infty} \exp(-y^2) dy\right) \left(\int_{-\infty}^{\infty} \exp(-x^2) dx\right)} \quad (2.8)$$

$$= \sqrt{\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \exp(-(y^2 + x^2)) dy dx} \quad (2.9)$$

According to Fubini's theorem, a double integral can be seen as an area integral (Appendix A.6):

$$\int_{-\infty}^{\infty} \exp(-x^2) = \sqrt{\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \exp(-(x^2 + y^2)) d(x, y)} \quad (2.10)$$

The area integral in equation 2.10 can consequently be transformed from Cartesian coordinates to Polar coordinates to produce a much easier integral. With parametrization x and y are replaced with:

$$\begin{aligned} x &= r \cos \theta \\ y &= r \sin \theta \\ d(x, y) &= r d(r, \theta) \end{aligned} \quad (2.11)$$

¹The parametrization with polar coordinates that is used to prove the Gaussian integral simplifies later steps of the derivation as well

The change of variables in the integral requires a multiplication with the determinant of the jacobian matrix², the ‘Jacobian’, that is defined as follows:

$$J(r, \phi) = \begin{bmatrix} \frac{\partial x}{\partial r} & \frac{\partial x}{\partial \theta} \\ \frac{\partial y}{\partial r} & \frac{\partial y}{\partial \theta} \end{bmatrix} = \begin{bmatrix} \frac{\partial(r \cos \theta)}{\partial r} & \frac{\partial(r \cos \theta)}{\partial \theta} \\ \frac{\partial(r \sin \theta)}{\partial r} & \frac{\partial(r \sin \theta)}{\partial \theta} \end{bmatrix} = \begin{bmatrix} \cos \theta & -r \sin \theta \\ \sin \theta & r \cos \theta \end{bmatrix} \quad (2.12)$$

$$|J(r, \phi)| = \left| \begin{bmatrix} \cos \theta & -r \sin \theta \\ \sin \theta & r \cos \theta \end{bmatrix} \right| = r \cos^2 \theta - (-r \sin^2 \theta) = r(\cos^2 \theta + \sin^2 \theta) = r \quad (2.13)$$

Substitution of the expression in the exponent results in:

$$x^2 + y^2 = r^2 \sin^2 \theta + r^2 \cos^2 \theta, \quad (2.14)$$

in which the sinusoids conveniently disappear by applying the pythagorean identity which states that $\sin^2 x + \cos^2 x = 1$, thus:

$$r^2 \sin^2 \theta + r^2 \cos^2 \theta = r^2 (\sin^2 \theta + \cos^2 \theta) = r^2. \quad (2.15)$$

The proof is completed by computation of the transformed integrals:

$$\begin{aligned} \int_{-\infty}^{\infty} \exp(-x^2) dx &= \sqrt{\int_0^{2\pi} \int_0^{\infty} \exp(-r^2) r dr d\theta} \\ &= \sqrt{2\pi \left[-\frac{1}{2} \exp(-r^2) \right]_0^{\infty}} \\ &= \sqrt{2\pi \left(0 - \left(-\frac{1}{2} \exp(0) \right) \right)} = \sqrt{\pi} \end{aligned} \quad (2.16)$$

The solution and proof for the integral of the multivariate gaussian function is given in Appendix A.2 as a generalization of the scalar version. The multivariate version is required to solve the marginalization for the window vectors of the similarity function.

2.2.2 Marginalization of the unknown texture

This section is an extension of the section ‘Texture marginalization’ as presented in the paper[4] of Sanja Damjanović. It features a complete derivation of the likelihood function and highlights an error in the approximation that complicates the next stages of the derivation.

The likelihood function for the proposed model is obtained by marginalizing several variables out of the probability density function. The model assumes the measurement vectors \mathbf{z}_i to be normal distributed random vectors with mean \mathbf{s} , covariance matrix \mathbf{C} and gain factor α_i . The expression for the probability density is a Gaussian function:

²The change of variables in the derivation of [4] lacks the Jacobian

$G(\mathbf{z}_i - \alpha_i \mathbf{s})$. Also, it is assumed that the measurements in the camera views, $\mathbf{z}_1, \dots, \mathbf{z}_k$, are uncorrelated. Therefore, we can define the conditional probability as:

$$p(\mathbf{z}_1, \mathbf{z}_2 \mid x, \mathbf{s}, \alpha_1, \alpha_2) = G(\mathbf{z}_1 - \alpha_1 \mathbf{s})G(\mathbf{z}_2 - \alpha_2 \mathbf{s}) \quad (2.17)$$

The goal is to find an expression for the likelihood function for \mathbf{z}_1 and \mathbf{z}_2 : $p(\mathbf{z}_1, \mathbf{z}_2 \mid x)$, where \mathbf{z}_2 depends on x . The initial probability density in Equation 2.17 depends on the camera gain factors, α_1 and α_2 , and the unknown texture, \mathbf{s} ; however, these parameters are unknown and have to be marginalized out of the joint probability density. This section solves the marginalization of the expression with respect to the multivariate vector \mathbf{s} . The marginalization is obtained by the integral of the probability density of the multivariate variable for the unknown texture \mathbf{s} :

$$p(\mathbf{z}_1, \mathbf{z}_2 \mid x, \alpha_1, \alpha_2) = \int_{-\infty}^{\infty} p(\mathbf{z}_1, \mathbf{z}_2 \mid x, \mathbf{s}, \alpha_1, \alpha_2) p(\mathbf{s} \mid x) d\mathbf{s} \quad (2.18)$$

The prior probability density function for the texture \mathbf{s} is assumed to be based on a complete lack of prior knowledge. It is written as a normalization constant K that depends on the width of $p(\mathbf{s})$. We assume:

$$p(\mathbf{s} \mid x) = K \quad (2.19)$$

The normalization constant K depends on the width of $p(\mathbf{s})$. Any width for $p(\mathbf{s})$ is sufficient as long as it covers the range of interest of \mathbf{z}_1 and \mathbf{z}_2 . Therefore, K is undetermined, but this is of no importance since K does not depend on the measurement vectors \mathbf{z}_i , and we are only interested in differences of the likelihood.

The probability densities for \mathbf{z}_1 and \mathbf{z}_2 are (with \mathbf{s} fixed) two uncorrelated normal distributed random vectors with mean \mathbf{s} and covariance matrix \mathbf{C} . The probability density function for the random vector is defined as:

$$G(\mathbf{x}) = G(\mathbf{z}_i, \mathbf{0}, \mathbf{C}) = \sqrt{\frac{1}{(2\pi)^k |\mathbf{C}|}} \exp\left(-\frac{1}{2} \mathbf{z}_i^T \mathbf{C}^{-1} \mathbf{z}_i\right) = \sqrt{\frac{|\mathbf{P}|}{(2\pi)^k}} \exp\left(-\frac{1}{2} \mathbf{z}_i^T \mathbf{P} \mathbf{z}_i\right), \quad (2.20)$$

where \mathbf{C} is the covariance matrix and \mathbf{P} its precision matrix counterpart. The notation with the precision matrix simplifies the expression in later stages of the derivation. Also, this notation is used in Chapter 3 to describe the contribution of the individual weights for the residuals of the measurement vectors.

The theorem for the integral of a multivariate Gaussian function can be applied to the expression after substitution of \mathbf{z}_1 and \mathbf{z}_2 by \mathbf{h} and \mathbf{y} as follows:

$$\mathbf{h} = \frac{\mathbf{z}_1}{\alpha_1} - \mathbf{s} \quad (2.21a)$$

$$\mathbf{y} = \frac{\mathbf{z}_1}{\alpha_1} - \frac{\mathbf{z}_2}{\alpha_2} \quad (2.21b)$$

$$\mathbf{h} - \mathbf{y} = \frac{\mathbf{z}_2}{\alpha_2} - \mathbf{s} \quad (2.21c)$$

However, the unknown texture \mathbf{s} is the variable of integration. Therefore, the substitution introduces a Jacobian to the integral as described in Appendix A.7.

$$p(\mathbf{z}_1, \mathbf{z}_2 | x, \alpha_1, \alpha_2) = K \int_{-\infty}^{\infty} p(\mathbf{z}_1, \mathbf{z}_2 | x, \mathbf{s}, \alpha_1, \alpha_2) d\mathbf{s} \quad (2.22)$$

$$= K \int_{-\infty}^{\infty} p(\mathbf{z}_1, \mathbf{z}_2 | x, \frac{\mathbf{z}_1}{\alpha_1} - \mathbf{h}, \alpha_1, \alpha_2) |\mathbf{J}| d\mathbf{h} \quad (2.23)$$

The probability density function is obtained by substitution of the Gaussian function with the variables \mathbf{h} and \mathbf{y} . This yields the expression:

$$\begin{aligned} p(\mathbf{z}_1, \mathbf{z}_2 | x, \mathbf{s}, \alpha_1, \alpha_2) &= G(\alpha_1 \mathbf{h}) G(\alpha_2 (\mathbf{h} - \mathbf{y})) & (2.24) \\ &= a \exp\left(-\frac{1}{2} \alpha_1^2 \mathbf{h}^T \mathbf{P} \mathbf{h}\right) \exp\left(-\frac{1}{2} \alpha_2^2 (\mathbf{h} - \mathbf{y})^T \mathbf{P} (\mathbf{h} - \mathbf{y})\right) & (2.25) \end{aligned}$$

where the constant of the density function, a , is:

$$a = \frac{K}{(2\pi)^n \det(\mathbf{C})} = \frac{K \det(\mathbf{P})}{(2\pi)^n}. \quad (2.26)$$

The theorem of Section A.3 can be used to solve Equation 2.25; however, it is necessary to merge the expression within the exponents to obtain a single expression in the form of a Gaussian function. This rewrite yields the following expression:

$$p(\mathbf{z}_1, \mathbf{z}_2 | x, \mathbf{s}, \alpha_1, \alpha_2) = a \exp\left(-\frac{1}{2} (\alpha_1^2 + \alpha_2^2) \mathbf{h}^T \mathbf{P} \mathbf{h} + \alpha_2^2 \mathbf{y}^T \mathbf{P} \mathbf{h} - \frac{1}{2} \alpha_2^2 \mathbf{y}^T \mathbf{P} \mathbf{y}\right) \quad (2.27)$$

The joint density function, $p(\mathbf{z}_1, \mathbf{z}_2 | x, \alpha_1, \alpha_2)$, is obtained by marginalization of the unknown texture out of Equation 2.27. The substitution of variables, however, changes the variable of integration from \mathbf{s} to \mathbf{h} . For the remainder of the derivation in this section, we use a short-hand notation to keep the expressions short and clear. The joint density function to obtain is referred to as F . The integral we have to solve to obtain the marginalized expression $p(\mathbf{z}_1, \mathbf{z}_2 | x, \alpha_1, \alpha_2) = F$ is given by:

$$F = \int_{-\infty}^{\infty} p(\mathbf{z}_1, \mathbf{z}_2 | x, \mathbf{s}, \alpha_1, \alpha_2) p(\mathbf{s} | x) d\mathbf{s} \quad (2.28)$$

$$= aK \int_{-\infty}^{\infty} \exp\left(-\frac{1}{2} (\alpha_1^2 + \alpha_2^2) \mathbf{h}^T \mathbf{P} \mathbf{h} + \alpha_2^2 \mathbf{y}^T \mathbf{P} \mathbf{h} - \frac{1}{2} \alpha_2^2 \mathbf{y}^T \mathbf{P} \mathbf{y}\right) \det(\mathbf{J}) d\mathbf{h} \quad (2.29)$$

The change of variables switched the bound of integration. The expression is rewritten to make it compatible with the theorem of Section A.3; however, this introduces an alternating coefficient with respect to the window size to the expression:

$$F = (-1)^n aK \int_{-\infty}^{\infty} \exp\left(-\frac{1}{2} (\alpha_1^2 + \alpha_2^2) \mathbf{h}^T \mathbf{P} \mathbf{h} + \alpha_2^2 \mathbf{y}^T \mathbf{P} \mathbf{h} - \frac{1}{2} \alpha_2^2 \mathbf{y}^T \mathbf{P} \mathbf{y}\right) \det(\mathbf{J}) d\mathbf{h}. \quad (2.30)$$

The Jacobian matrix in this expression is obtained by:

$$\mathbf{J} = \begin{bmatrix} \frac{\partial s_1}{\partial h_1} & \cdots & \frac{\partial s_1}{\partial h_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial s_n}{\partial h_1} & \cdots & \frac{\partial s_n}{\partial h_n} \end{bmatrix} = \begin{bmatrix} \frac{\partial \left(\frac{z_1}{\alpha_1} - h_1\right)}{\partial h_1} & \cdots & \frac{\partial \left(\frac{z_1}{\alpha_1} - h_1\right)}{\partial h_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial \left(\frac{z_n}{\alpha_n} - h_n\right)}{\partial h_1} & \cdots & \frac{\partial \left(\frac{z_n}{\alpha_n} - h_n\right)}{\partial h_n} \end{bmatrix} = -\mathbf{I}_n. \quad (2.31)$$

Because \mathbf{J} is a negative identity matrix, it follows that the determinant of the Jacobian matrix also results in an alternating constant that depends on the window size:

$$\det(\mathbf{J}) = \det(-\mathbf{I}_n) = \prod_{i=1}^n (-1) = (-1)^n. \quad (2.32)$$

Although the Jacobian was not included in the derivation of the reference paper, it turns out in Equation 2.35 that it does not influence the final solution of the integral. It is, however, required for a proper and complete proof.

The integral for the marginalization is solved by applying the theorem of Equation A.18 (proved in Appendix A.3) to the probability density function. The theorem has several input variables that have to be extracted from Equation 2.30. The required variables (a , b , \mathbf{d} , c , and f) for the theorem are given by:

$$a = \frac{K \det(\mathbf{P})}{(2\pi)^n} (-1)^n \quad (2.33a)$$

$$b = \frac{1}{2}(\alpha_1^2 + \alpha_2^2) \quad (2.33b)$$

$$\mathbf{d} = (\mathbf{y}^T \mathbf{P})^T = \mathbf{P}^T \mathbf{y} = \mathbf{P} \mathbf{y} \quad (2.33c)$$

$$c = \alpha_2^2 \quad (2.33d)$$

$$f = -\frac{1}{2} \alpha_2^2 \mathbf{y}^T \mathbf{P} \mathbf{y} \quad (2.33e)$$

The solution for the integral of Equation 2.30 is solved by substitution of Equation 2.33a-e in Equation A.38. This results in the following expression:

$$\begin{aligned} F &= \frac{(-1)^n (-1)^n}{(2\pi)^n \frac{1}{|\mathbf{P}|}} \sqrt{\frac{1}{|\mathbf{P}|} \left(\frac{2\pi}{(\alpha_1^2 + \alpha_2^2)} \right)^n} \\ &\quad \cdot \exp \left(-\frac{1}{2} \alpha_2^2 \mathbf{y}^T \mathbf{P} \mathbf{y} \right) \\ &\quad \cdot \exp \left(\frac{\alpha_2^4}{2(\alpha_1^2 + \alpha_2^2)} \mathbf{y}^T \mathbf{P} \mathbf{P}^{-1} \mathbf{P} \mathbf{y} \right) \end{aligned} \quad (2.34)$$

As it appears, both the Jacobian and the switching of integral bounds introduce alternating constants with respect to the window size. However, both effects stem from the substitution of Equation 2.21 and should cancel each other out. The two terms combined introduce a square power to the negative coefficient. Therefore, the expressions is always positive for every window size (n):

$$(-1)^n (-1)^n = ((-1)^2)^n = 1 \quad (2.35)$$

The solution of Equation 2.34 can be simplified further by combination of the exponents to a single expression:

$$\begin{aligned} & -\frac{1}{2}\alpha_2^2 \mathbf{y}^T \mathbf{P} \mathbf{y} + \frac{\alpha_2^4}{2(\alpha_1^2 + \alpha_2^2)} \mathbf{y}^T \mathbf{P} \mathbf{P}^{-1} \mathbf{P} \mathbf{y} \\ = & \mathbf{y}^T \mathbf{P} \mathbf{y} \left(\frac{-\alpha_2^2(\alpha_2^2 - \alpha_1^2) + \alpha_2^4}{2(\alpha_1^2 + \alpha_2^2)} \right) = -\mathbf{y}^T \mathbf{P} \mathbf{y} \left(\frac{\alpha_1^2 \alpha_2^2}{2(\alpha_1^2 + \alpha_2^2)} \right) \end{aligned} \quad (2.36)$$

The expression in terms of \mathbf{z}_1 and \mathbf{z}_2 is obtained by substituting \mathbf{y} back with its original value of Equation 2.21b. This yields the expression:

$$(\dots) = - \left(\frac{\mathbf{z}_1}{\alpha_1} - \frac{\mathbf{z}_2}{\alpha_2} \right)^T \mathbf{P} \left(\frac{\mathbf{z}_1}{\alpha_1} - \frac{\mathbf{z}_2}{\alpha_2} \right) \left(\frac{\alpha_1^2 \alpha_2^2}{2(\alpha_1^2 + \alpha_2^2)} \right) \quad (2.37)$$

The final expression is obtained by rewriting the expression to one fraction:

$$(\dots) = - \frac{1}{\alpha_1^2 \alpha_2^2} (\alpha_2 \mathbf{z}_1 - \alpha_1 \mathbf{z}_2)^T \mathbf{P} (\alpha_2 \mathbf{z}_1 - \alpha_1 \mathbf{z}_2) \left(\frac{\alpha_1^2 \alpha_2^2}{2(\alpha_1^2 + \alpha_2^2)} \right) \quad (2.38)$$

$$= - \frac{(\alpha_2 \mathbf{z}_1 - \alpha_1 \mathbf{z}_2)^T \mathbf{P} (\alpha_2 \mathbf{z}_1 - \alpha_1 \mathbf{z}_2)}{2(\alpha_1^2 + \alpha_2^2)} \quad (2.39)$$

Because the multiplication of the measurement vectors \mathbf{z}_i and the precision matrix \mathbf{P} does not depend on the camera gains, it is possible to precompute these values. The weighting of the measurement vectors with the precision matrix \mathbf{P} —the matrix multiplication— is referred to by the variable ρ_{ij} :

$$\rho_{ij} = \mathbf{z}_i^T \mathbf{P} \mathbf{z}_j, \quad (2.40)$$

where i and j are indices that refer to the camera index. The precision matrix \mathbf{P} is symmetric, therefore: $\rho_{ij} = \rho_{ji}$. The final notation for the expression within the exponent is finally given by:

$$(\dots) = - \frac{\alpha_2^2 \rho_{11} + \alpha_1^2 \rho_{22} - 2\alpha_1 \alpha_2 \rho_{12}}{2(\alpha_1^2 + \alpha_2^2)} \quad (2.41)$$

If we substitute this expression for within the exponent back, we obtain the expression for the probability density function:

$$p(\mathbf{z}_1, \mathbf{z}_2 \mid x, \alpha_1, \alpha_2) = K \sqrt{\frac{|\mathbf{P}|}{(2\pi)^n}} \sqrt{\left(\frac{1}{(\alpha_1^2 + \alpha_2^2)} \right)^n} \exp \left(- \frac{\alpha_2^2 \rho_{11} + \alpha_1^2 \rho_{22} - 2\alpha_1 \alpha_2 \rho_{12}}{2(\alpha_1^2 + \alpha_2^2)} \right). \quad (2.42)$$

Unfortunately, the result of the marginalization differs from the solution of the reference paper. The first part of the probability function (highlighted in blue) does not depend on the camera gain α_i or the measurement vectors \mathbf{z}_i . When comparing the likelihoods, it is merely a scaling that does not influence the ratio, and is therefore omitted in the paper. However, the second part of the equation does depend on the camera gains. The derivation in this section reveals that the probability function in the paper lacks the power term n , as illustrated in red in Equation 2.42. The difference in the solutions originates from the multivariate integral that is required to marginalize for the unknown texture \mathbf{s} . The integral over the n -dimensional vector \mathbf{s} implies n integrals for every element of \mathbf{s} . The solution in the paper, as it appears, is obtained by applying a one-dimensional integral only. The size of windows, n , is usually large. Therefore, the behavior of the probability function is significantly different for a pole of order one compared to a pole of order n .

2.2.3 Marginalization of the camera gains

This conditional likelihood still depends on the camera gains. These conditional variables must be marginalized out of the expression to obtain the final likelihood function. In the previous section, the probability density function $p(\mathbf{z}_1, \mathbf{z}_2 | x, \alpha_1, \alpha_2)$ was obtained in Equation 2.42.

The unknown gains are neutralized by marginalization of the camera gains α_1 and α_2 . It is defined by:

$$p(\mathbf{z}_1, \mathbf{z}_2 | x) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} p(\mathbf{z}_1, \mathbf{z}_2 | x, \alpha_1, \alpha_2) p(\alpha_1) p(\alpha_2) d\alpha_2 d\alpha_1 \quad (2.43)$$

The prior pdfs for $p(\alpha_1)$ and $p(\alpha_2)$ are chosen to reflect the prior knowledge about the unknown gains. In accordance with the paper: ‘‘The gain factors should not deviate too much from 1. For that reason, we chose for $p(\alpha_k)$ a normal distribution, centred around 1, and with a standard deviations σ_α ’’ [4]. Therefore, the pdfs $p(\alpha_1)$ and $p(\alpha_2)$ are defined as:

$$p(\alpha_1)p(\alpha_2) = \frac{1}{\sigma_1\sqrt{2\pi}} \exp\left(-\frac{(\alpha_1 - \mu_1)^2}{2\sigma_1^2}\right) \cdot \frac{1}{\sigma_2\sqrt{2\pi}} \exp\left(-\frac{(\alpha_2 - \mu_2)^2}{2\sigma_2^2}\right) \quad (2.44)$$

$$= \frac{1}{\sigma_1\sigma_2 2\pi} \exp\left(-\frac{(\alpha_1 - \mu_1)^2}{2\sigma_1^2} - \frac{(\alpha_2 - \mu_2)^2}{2\sigma_2^2}\right). \quad (2.45)$$

In the paper, it is concluded that an analytical solution for Equation 2.43 is easily obtained by the approximation of the term $\frac{1}{\alpha_1^2 + \alpha_2^2}$ by $\frac{1}{2}$. However, this assumption has a severe impact on the repaired solution of Equation 2.42 with an n^{th} -order term. Although the approximation might ‘not be too rough’ for a simple pole, it certainly is for an n^{th} -order pole. The error of the approximation is amplified by the power n .

We conclude that the term should not be approximated; however, this results in an integral of a very complicated function. To obtain the likelihood function, we have to

solve the integral³ of a very high order pole multiplied with a Gaussian function.

$$p(\mathbf{z}_1, \mathbf{z}_2 | x) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \sqrt{\left(\frac{1}{(\alpha_1^2 + \alpha_2^2)}\right)^n} \exp(f(\alpha_1, \alpha_2, \dots)) d\alpha_2 d\alpha_1 \quad (2.46)$$

The improper integral for the Gaussian function itself is complicated, but solvable. However, Equation 2.46 remains unsolved due to the red-colored term.

2.3 Approximation by power series

The previous section concluded with the unsolved integral. To obtain the the likelihood function, it is necessary to complete the final integration step. As an alternative to a proper analytical solution, we have decided to approximate the expression with a power series to obtain an approximate likelihood function. Because the series is of a pure polynomial notation, the integral is always solvable for every chosen orde. The taylor series for $f(x)$ is defined as [17, pp 242]:

$$f(x) = f(a) + \frac{f'(a)}{1!}(x-a) + \frac{f''(a)}{2!}(x-a)^2 + \frac{f^{(3)}(a)}{3!}(x-a)^3 + \dots = \sum_{n=0}^{\infty} \frac{f^{(n)}(a)}{n!}(x-a)^n \quad (2.47)$$

However, since we have more than one camera gain that needs to be marginalized, the multivariate taylor series is required:

$$T(x_1, \dots, x_k) = \sum_{n_1=0}^{\infty} \dots \sum_{n_k=0}^{\infty} \frac{(x_1 - a_1)^{n_1} \dots (x_k - a_k)^{n_k}}{n_1! \dots n_k!} \left(\frac{\partial^{n_1 + \dots + n_k} f}{\partial x_1^{n_1} \dots \partial x_k^{n_k}} \right) (a_1, \dots, a_k). \quad (2.48)$$

Due to the complexity of the expression, the multivariate taylor expansion is computed with the use of Maple. The required scripts are given in Appendix E. For example, the first order expansion about $\alpha_1 = 1$ and $\alpha_2 = 1$ is given by:

$$f_0(\rho) = \frac{K}{\sqrt{2^n}} \exp\left(-\frac{1}{4}(\rho_{11} + \rho_{22} - 2\rho_{12})\right) \quad (2.49)$$

The even order terms of the expansion are zero if the integration boundaries about α_1 and α_2 are chosen symmetric. The third order expansion is given by:

$$\int_{-b}^b \int_{-a}^a \text{taylor}(f(\dots), 3) d\alpha_1 d\alpha_2 = c_0 \left(\begin{array}{c} c_1(\rho_{11} - \rho_{22})^2 - (4n + 4)c_2(\rho_{11} + \rho_{22}) \dots \\ - 8c_1\rho_{12} - 32\left(\frac{a^2}{\sigma_1^2} + \frac{b^2}{\sigma_2^2}\right) + 4n^2c_1 + 96 \end{array} \right), \quad (2.50)$$

³This double integral can be written as a multivariate integral, as shown in the generalization of Section 2.5

with the constants c_0 , c_1 , and c_2 :

$$c_0 = \frac{K}{24} ab \exp\left(-\frac{1}{4}(\rho_{11} + \rho_{22} - 2\rho_{12})\right) \quad (2.51)$$

$$c_1 = a^2 + b^2 \quad (2.52)$$

$$c_2 = a^2 - b^2 \quad (2.53)$$

2.3.1 Computational complexity

The complexity of the first and the third order expansion seems acceptable. However, the number of terms increases significantly for higher order terms. This is especially the case if more than two cameras are involved in the likelihood function. The number of coefficients (that have to be computed for every window) for k cameras and order o is given by:

$$c(k, o) = bc(M(k) + o - 1, o - 1) = \frac{(\frac{1}{2}k^2 + \frac{1}{2}k + o - 1)!}{(o - 1)! (\frac{1}{2}k(k + 1))!}, \quad (2.54)$$

with $bc(n, k)$ as the number of terms:

$$bc(n, k) = \frac{n!}{k!(n - k)!}, \quad (2.55)$$

and $M(k)$ as:

$$M(k) = bc(k + 1, k - 1) = \frac{1}{2}k(k + 1) \quad (2.56)$$

Table 2.1 gives an impression of the enormous number of coefficients that have to be computed for higher order approximations. Every

Order	2 cams	3 cams	4 cams	5 cams
1	1	1	1	1
3	10	28	66	136
5	35	210	1001	3876
7	84	924	8008	54264
9	165	3003	43758	490314
11	286	8008	184756	3268760
13	455	18564	646646	17383860
15	680	38760	1961256	77558760

Table 2.1: Number of coefficients (required multiplications per window).

2.3.2 Approximation results

It was deemed that investigation of the effects of the Taylor expansion is useful. It was already concluded in the previous section that the complexity of the expansion increases

significantly for higher orders. The approximation can be used only if the error is small enough.

We investigate how good the Taylor expansions of certain orders fit to the original function. In Figure 2.1, the Taylor expansion for the 3rd, the 5th and the 16th order is given for a likelihood function with a window size of 2. It became clear in the previous section that the Taylor expansion is practical for low expansion orders only. However, it appears in Figure 2.1 that even a very high order (16) expansion fits the original function poorly.

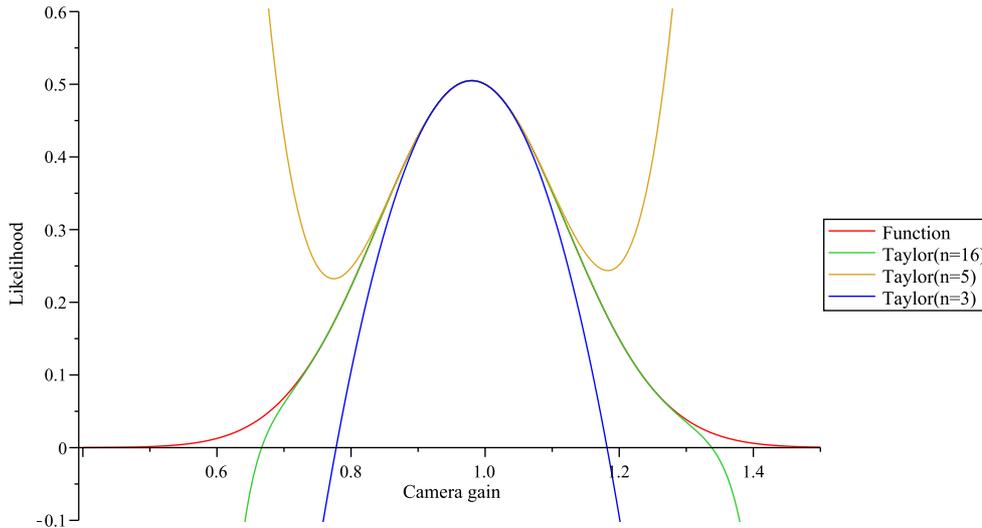


Figure 2.1: Marginalization on one-dimensional Taylor approximations about $\alpha = (1, 1)$ for $n = 2$

Next, we simulate the effects of the Taylor expansion for larger windows. The results shown in Figure 2.2 indicate that the expansions fit even worse for a likelihood function with a larger window. This is very unfortunate, because the likelihood function should work for very large window sizes.

Further simulations indicate that the Taylor expansions sometimes even become unstable for likelihood functions with certain window sizes. The results for a seventh order Taylor expansion are shown in Figure 2.3. Some window sizes are selected that become unstable in the range of interest. The use of approximations is useful only if the error is not large; however, it has become clear from the simulations that a Taylor expansion is not a good method to solve the last integration step.

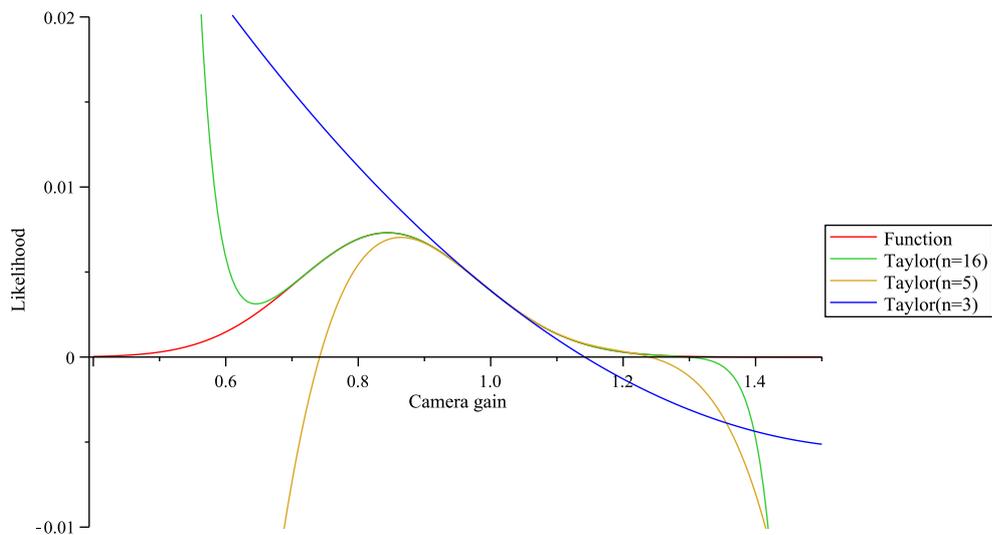


Figure 2.2: Marginalization on one-dimensional Taylor approximations about $\alpha = (1, 1)$ for $n = 16$

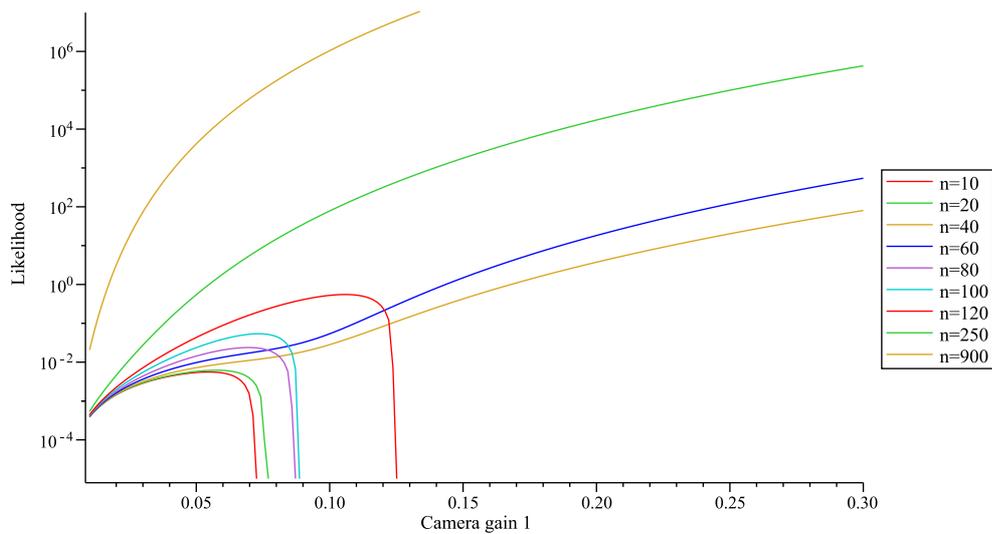


Figure 2.3: Marginalization on a two-dimensional seventh-order Taylor expansion

2.4 Suggestion for alternative analytic solution

This section should be read as a suggestion for future research. It appeared that the method that was used to solve the Gaussian function in the first place, also simplifies the expression that has to be solved to obtain a solution for the likelihood function. Although the transformed expression is more complex and remains unsolved, we expect

that is a step in the right direction to solve the integral in the future.

Our proposal for an alternative solution is:

- ▷ Step one: transform the expression to polar coordinates.
- ▷ Step two: apply Cauchy's integral formula on the transformed integral.

2.4.1 Polar coordinates

The first step is (partially) given in this section. We define the following variables for the transformation:

$$\alpha_1 = r \cos \theta \quad (2.57)$$

$$\alpha_2 = r \sin \theta, \quad (2.58)$$

for which the jacobian is defined by:

$$|J(r, \phi)| = \left| \begin{bmatrix} \frac{\partial \alpha_1}{\partial r} & \frac{\partial \alpha_1}{\partial \theta} \\ \frac{\partial \alpha_2}{\partial r} & \frac{\partial \alpha_2}{\partial \theta} \end{bmatrix} \right| = \left| \begin{bmatrix} \cos \theta & -r \sin \theta \\ \sin \theta & r \cos \theta \end{bmatrix} \right| = r(\cos^2 \theta + \sin^2 \theta) = r \quad (2.59)$$

The expression for the Gaussian function that model \mathbf{z}_1 and \mathbf{z}_2 with a normal distribution:

$$\begin{aligned} p(\mathbf{z}_1, \mathbf{z}_2 | x, \alpha_1, \alpha_2) &= \sqrt{\frac{|\mathbf{P}|}{(2\pi)^n}} \sqrt{\left(\frac{1}{r^2}\right)^n} \exp\left(-\frac{r^2 \sin^2 \theta \rho_{11} + r^2 \cos^2 \theta \rho_{22} - 2r^2 \sin \theta \cos \theta \rho_{12}}{2r^2}\right) \\ &= \sqrt{\frac{|\mathbf{P}|}{(2\pi)^n}} \frac{1}{r^n} \exp\left(-\frac{1}{2} (\sin^2 \theta \rho_{11} + \cos^2 \theta \rho_{22} - 2 \sin \theta \cos \theta \rho_{12})\right) \end{aligned} \quad (2.60)$$

This results in the following rewritten expression for the likelihood function:

$$K \int_0^\infty \int_0^{2\pi} \frac{1}{r^n} \exp\left(-\frac{1}{2} (\rho_{11} \sin^2 \theta + \rho_{22} \cos^2 \theta - 2\rho_{12} \sin \theta \cos \theta)\right) p(\alpha_1)p(\alpha_2) r \, d\theta \, dr \quad (2.61)$$

The parametrization of the camera gains yields:

$$\begin{aligned} p(\alpha_1)p(\alpha_2) &= \frac{1}{\sigma_1 \sigma_2 2\pi} \exp\left(-\frac{r^2 (\sin^2 \theta + \cos^2 \theta) - r (\mu_2 \sin \theta + \mu_1 \cos \theta) + \mu_1 \mu_2}{2\sigma_1^2 \sigma_2^2}\right) \\ &= \frac{1}{\sigma_1 \sigma_2 2\pi} \exp\left(-\frac{1}{2} \frac{r^2 - r (\mu_2 \sin \theta + \mu_1 \cos \theta) + \mu_1 \mu_2}{\sigma_1^2 \sigma_2^2}\right) \\ &= \frac{1}{\sigma_1 \sigma_2 2\pi} \exp\left(-\frac{1}{2} \frac{r^2 - r\sqrt{2} \cos\left(\theta - \frac{\pi}{4}\right) + 1}{\sigma_1^2 \sigma_2^2}\right) \end{aligned} \quad (2.62)$$

2.5 Generalization for multiple cameras

One of the initial goals of this research project was to explore the possibilities to extend the algorithm of Sanja for more than two cameras. It was hypothesized that taking advantage of the information of more camera views would improve the performance of the similarity measure. More information is available, and partially occluded surfaces of objects can be visible in a subset of the available set of camera views. It was concluded in Section 2.2.3 that an analytical solution for the statistical model that was proposed in the paper of S. Damjanović [4] is far more difficult than first expected. The problem of the similarity measure remains unsolved, and even a proper approximate solution appeared to be out of reach. However, an alternative method was proposed in Section 2.4 in which a transformation of the integral simplifies the expression. We expect that a solution to the last marginalization step is possible if Cauchy's integral formula is used to handle the singularity in the expression.

In this section, we show that the extension of the model for more than two cameras leads to an expression that is more complex. The marginalization steps required to obtain the analytical solution for the similarity measure are, however, possible if the problem is solved for two cameras.

In the first marginalization step we remove the unknown texture \mathbf{s} from the expression. The generalized expression we have to obtain for k cameras is defined as:

$$p(\mathbf{z}_1, \dots, \mathbf{z}_k \mid d_1, \dots, d_{k-1}, \alpha_1, \dots, \alpha_k) = p(\mathbf{Z} \mid \mathbf{d}, \alpha), \quad (2.63)$$

where the variables in the expression have a higher dimensionality as defined by:

$$\begin{array}{ll} \text{Windows} & \mathbf{Z} \in \mathbb{R}^{k \times n} \\ \text{Unknown texture} & \mathbf{s} \in \mathbb{R}^n \\ \text{Disparity} & \mathbf{d} \in \mathbb{R}^{k-1} \\ \text{Camera gains} & \alpha \in \mathbb{R}^k \end{array} \quad (2.64)$$

The integral that has to be solved for marginalization of the unknown texture \mathbf{s} is (in this section) referred to as F . F is obtained by the improper multivariate integral:

$$F = p(\mathbf{Z} \mid \mathbf{d}, \alpha) = K \int_{-\infty}^{\infty} p(\mathbf{Z} \mid \mathbf{d}, \mathbf{s}, \alpha) p(\mathbf{s} \mid \mathbf{d}) \, d\mathbf{s}. \quad (2.65)$$

This integral is solved in an identical way as the integral in Section 2.2.2, as defined in Equation 2.17. This extended derivation has several trivial steps compared to the derivation for two cameras. Therefore, this derivation should be read as an extension to Section 2.2.2 where all steps are explained in more detail. For example, the substitution for n cameras is omitted. The expression of conditional probabilities for k cameras generalizes to a product of Gaussians:

$$F = K \int_{-\infty}^{\infty} \prod_{i=1}^k G(\mathbf{z}_i - \alpha_i \mathbf{s}, 0, \mathbf{P}^{-1}) \, d\mathbf{s} \quad (2.66)$$

$$= K \int_{-\infty}^{\infty} \prod_{i=1}^k \sqrt{\frac{|\mathbf{P}|}{(2\pi)^n}} \exp\left(-\frac{1}{2} (\mathbf{z}_i - \alpha_i \mathbf{s})^T \mathbf{P} (\mathbf{z}_i - \alpha_i \mathbf{s})\right) \, d\mathbf{s} \quad (2.67)$$

First, the integral is rewritten to exclude the constant part. Then, the expression can be rewritten to bring the product into the exponent, because a product of exponents can be rewritten as an exponent of additions. This yields the expression:

$$F = K \sqrt{\frac{|\mathbf{P}|^k}{(2\pi)^{nk}}} \int_{-\infty}^{\infty} \prod_{i=1}^k \exp\left(-\frac{1}{2} (\mathbf{z}_i - \alpha_i \mathbf{s})^T \mathbf{P} (\mathbf{z}_i - \alpha_i \mathbf{s})\right) \quad (2.68)$$

$$= K \sqrt{\frac{|\mathbf{P}|^k}{(2\pi)^{nk}}} \int_{-\infty}^{\infty} \exp\left(-\frac{1}{2} \sum_{i=1}^k (\mathbf{z}_i - \alpha_i \mathbf{s})^T \mathbf{P} (\mathbf{z}_i - \alpha_i \mathbf{s})\right) \quad (2.69)$$

The integral is solved by applying the theorem of Appendix A.3. This theorem requires a rewrite of the expression to the form:

$$F = K \sqrt{\frac{|\mathbf{P}|^k}{(2\pi)^{nk}}} \int_{-\infty}^{\infty} \exp\left(-\left(\sum_{i=1}^k \frac{\alpha_i^2}{2}\right) \mathbf{s}^T \mathbf{P} \mathbf{s} + \left(\sum_{i=1}^k \alpha_i \mathbf{z}_i^T \mathbf{P}\right) \mathbf{s} - \sum_{i=1}^k \frac{\mathbf{z}_i^T \mathbf{P} \mathbf{z}_i}{2}\right) \quad (2.70)$$

The input parameters (a , b , c , \mathbf{d} , and f) for the formula of Appendix A.3 are extracted from Equation 2.70. This results in:

$$a = K \sqrt{\frac{|\mathbf{P}|^k}{(2\pi)^{nk}}} \quad (2.71a)$$

$$b = \frac{1}{2} \sum_{i=1}^k \alpha_i^2 \quad (2.71b)$$

$$c = 1 \quad (2.71c)$$

$$\mathbf{d} = \sum_{i=1}^k \alpha_i \mathbf{z}_i^T \mathbf{P} = \sum_{i=1}^k \alpha_i \mathbf{P} \mathbf{z}_i \quad (2.71d)$$

$$f = -\frac{1}{2} \sum_{i=1}^k \mathbf{z}_i^T \mathbf{P} \mathbf{z}_i \quad (2.71e)$$

Finally, the solution for the integral is:

$$\begin{aligned} F &= K \sqrt{\frac{|\mathbf{P}|^k}{(2\pi)^{nk}}} \sqrt{\left(\frac{2\pi}{\sum_{i=1}^k \alpha_i^2}\right)^n \frac{1}{|\mathbf{P}|}} \\ &\cdot \exp\left(-\frac{1}{2} \sum_{i=1}^k \mathbf{z}_i^T \mathbf{P} \mathbf{z}_i\right) \\ &\cdot \exp\left(\frac{1}{2 \sum_{i=1}^k \alpha_i^2} \left(\sum_{i=1}^k \alpha_i \mathbf{z}_i^T\right) \mathbf{P} \left(\sum_{i=1}^k \alpha_i \mathbf{z}_i\right)\right) \end{aligned} \quad (2.72)$$

The marginalization of the camera gains introduces a second integration step with respect to α_i . Therefore, it is practical to rewrite Equation 2.72 to a constant part, and

a part that depends on a_i . This last part contains the same $(\frac{n}{2})^{\text{th}}$ -order pole as the high-order pole that was discussed in Section 2.2.3:

$$F = K \sqrt{\frac{|\mathbf{P}|^{k-1}}{(2\pi)^{n(k-1)}}} \exp\left(-\frac{1}{2} \sum_{i=1}^k \rho_{ii}\right) \quad (2.73)$$

$$\cdot \left(\frac{1}{\sum_{i=1}^k \alpha_i^2}\right)^{\frac{n}{2}} \exp\left(\frac{\sum_{i=1}^k \sum_{j=1}^k \alpha_i \alpha_j \rho_{ij}}{2 \sum_{i=1}^k \alpha_i^2}\right) \quad (2.74)$$

The symmetry of the covariance matrix requires the computation of the upper triangle of the coefficients only. Therefore, a simplified expression reduces to:

$$F = K \sqrt{\frac{|\mathbf{P}|^{k-1}}{(2\pi)^{n(k-1)}}} \exp\left(-\frac{1}{2} \sum_{i=1}^k \rho_{ii}\right) \quad (2.75)$$

$$\cdot \left(\frac{1}{\sum_{i=1}^k \alpha_i^2}\right)^{\frac{n}{2}} \exp\left(\frac{\sum_{i=1}^k \left(\alpha_i^2 \rho_{ii} + 2 \sum_{j=i+1}^k \alpha_i \alpha_j \rho_{ij}\right)}{2 \sum_{i=1}^k \alpha_i^2}\right) \quad (2.76)$$

This integral has the same properties as the integral of Equation 2.46.

Also, we can apply the same transformation for the generalized expression as we did apply for the expression with two cameras. For example: we have to transform to a spherical coordinate system instead of polar coordinate system for three cameras:

$$\alpha_1 = r \sin \theta \cos \phi \quad (2.77)$$

$$\alpha_2 = r \sin \theta \sin \phi \quad (2.78)$$

$$\alpha_3 = r \cos \theta \quad (2.79)$$

The same method can be used for more than three cameras (hyperspherical coordinates).

2.6 Discussion

The goal of this Chapter was to improve and generalize the likelihood function that was derived in the paper of Damjanović et al. The derivation of the likelihood function started very promising. The assumptions in the paper [4] to formulate the sound probabilistic model are valid and promising. Unfortunately, it became clear that a small error was made in the paper during the first part of the derivation: the marginalization of the unknown texture. The integral that is required to perform the marginalization of the unknown texture was (probably) assumed to be univariate; however because the unknown texture is a multi-dimensional variable, a multivariate integral should have been used. It should also be noted that the derivation in the paper does not include a Jacobian to compensate for the change of variables. However, this is just a sidenote for a complete proof, because that part of the final likelihood function is correct.

In the paper, the solution for the marginalization of the gains is obtained by some approximations. The term that was assumed to be constant (a first order pole) actually

appeared to be a very high order pole (See Equation 2.46). The order of this pole depends on the window size. This led us to the conclusion that this term should not be assumed to be constant, with severe consequences for the solvability of the integral. Literature research for integrals of complex Gaussian functions and several brainstorming sessions have been futile in the quest for a complete analytical solution.

In our search for a sufficient solution, we have tried to perform several types of approximations. In Section 2.3, it became clear that Taylor expansions for (parts of) the expression introduced very large errors. Also, the expansion became very complex for even low order expansions. We have to conclude that this method is useless to obtain a proper likelihood function with (that satisfies the probabilistic model).

Research question (Q1b) was formulated to investigate the possibilities for a multi-camera likelihood function: *‘How is the new likelihood function generalized for more than two camera views?’*. The absence of a solution for the two-camera likelihood function, unfortunately, has the consequence that we are unable to generalize the likelihood function for more than two cameras. However, we have shown in Section 2.5 that the derivation process becomes more abstract, but not more difficult to integrate. Once a solution is found for the two-camera case, it is possible to use the same method to find a solution for the generalized case.

The probabilistic model is built around normal distributions. We suspect that one of the methods that is used to prove the Gaussian integral can be used to transform and simplify the integral of Equation 2.46. Transformation of the function to polar coordinates results in an expression that is easier to integrate. We suspect that it is possible to solve the integral of the high-order pole with complex analysis. Cauchy’s integral formula can be used to transform the integral to the complex domain and integrate around the singularity. This method is recommended for future research.

Finally, we observe that the Mahalanobis appears as the core metric for the likelihood function. Although we did not succeed in obtaining a complete solution, it is clear that some assumptions have led to the generalization of the SSD to a Mahalanobis distance. The difference in performance between the Mahalanobis distance and the Euclidean distance is the main subject of the next chapter.

Mahalanobis distance versus Euclidean distance

In previous chapters, the importance of a good similarity measure for windows around candidate points in stereo views is addressed. The quality of the similarity measure lays the foundation for the performance of the depth perception in the stereo vision system. Section 2.1 introduced several matching metrics for block similarity amongst which the *Sum of Squared Differences* (SSD) that is used as the basis for a stereo algorithm with a Bayesian approach. It is based on the *constant brightness assumption* (CBA) that states that the pixel values in two matching windows in the left and the right image are equal apart from white noise [4]. The SSD is by definition the squared Euclidean distance.

The Euclidean distance of the residual error between pixel blocks in two images appeared to provide a good basis for a similarity measure in [3] and [1]. The matching is performed on the individual pixel intensities and provides a dense disparity map; however, the correspondance problem remains difficult for occluded regions and smooth texture. With an Euclidean distance, the pixel values within windows contribute equally without utilizing additional information such as: the distance of the pixel location to the window center, the mutual information between the primary color values of a pixel and the information between different pixel intensities. We expect that replacing the Euclidean distance with a Mahalanobis distance as the metric for similarity measure offers a method to include these effects and improve the stereo correspondence. The advantage of the Mahalanobis distance is that it takes into account the correlations of the data set.

In this chapter, we research whether a modification of the similarity measure with a Mahalanobis distance increases the stereo correspondence performance. The model with the Euclidean distance that is based on the CBA appears to be incomplete. An extension of the model with a Mahalanobis distance allows incorporation of geometric distortion, sampling errors and other effects in the covariance matrix of the metric. Therefore, a training stage is required to obtain a proper covariance matrix that describes the effects that occur between the corresponding pixel blocks of the images. However, the covariance matrix has to be inverted to its precision matrix counterpart to complete the algorithm.

The components of the precision matrix are weights for all error residuals. Inversion of a matrix requires the matrix to be positive definite. Unfortunately, an estimated covariance matrix does not necessarily have to be invertible. It is possible to force the positive definite requirement by regularization of the covariance matrix with spectral decomposition. The matrix is invertible if and only if all its eigenvalues are positive. After eigendecomposition, the eigenvalues of the covariance matrix are increased slightly with a constant to suffice the requirement. The amount of regularization affects the performance. Because the amount of regularization affects the performance of the correspondance, the optimization of the regularization constant is an important part of the experiment.

The model introduced by Belhumeur et al. [1] leads to a maximum likelihood cost function based on the Euclidean distance as its metric. A more extensive model derived by Cox et al. in [3] includes the possibility of a Mahalanobis distance, but the covariance matrix is assumed to be a diagonal matrix with a single constant weight to simplify the algorithm. Also, the new likelihood cost function in Chapter 2.2.2 as proposed by Damjanović et al. [4] includes the Mahalanobis distance as metric. The probabilistic model describes the existence of an unknown texture, and uncertain gains and offsets. Simplification of the gains and offsets reduces the likelihood function to the form that is introduced by Cox et al. This generic model that includes a multivariate normal distribution for the unknown texture is expected to provide improved stereo correspondance; however, no experiments to validate this claim have been done.

The goal of this chapter is to research a Bayesian extension of the SSD metric with the Mahalanobis distance and to answer: ‘is it useful to include a covariance matrix built from a training set to enhance the generic SSD-likelihood function to improve the stereo correspondance?’ The addition of the Mahalanobis distance to the model increases the computational complexity of the model significantly; therefore, it should increase performance substantially for it to be useful. The algorithm for the stereo depth perception should not be unnecessarily complex.

Firstly, the Bayesian approach is described in Section 3.1 to lay a foundation for the experiment. In Section 3.2, the implementation and design decisions are discussed. Finally, the results presented in Section 3.3 are discussed in Section 3.4.

3.1 Mahalanobis likelihood derivation

In this section, a mathematical description of the maximum likelihood cost function is given, as well as a method to generate a covariance matrix for it. This probabilistic approach to the stereo correspondance problem formulates a probability density for the observed data given a ground truth [4].

Let two cameras from different viewpoints produce corresponding images \mathbf{I}_1 and \mathbf{I}_2 of a scene and let $p(\mathbf{z}_1, \mathbf{z}_2 | x)$ be a maximum likelihood function that evaluates local measurements \mathbf{z}_1 and \mathbf{z}_2 in these images for a given disparity x . The respective windows contain all local pixels within the region of interest reshaped to a one dimensional column

vector:

$$\mathbf{z} = [z_1, \dots, z_n]^T \quad (3.1)$$

Both images are assumed to be preprocessed and rectified; the search space is reduced to one dimension. The conditional variable x of the likelihood function denotes the horizontal shift of the local right window that is to be compared to the local left window. In the *Winner Takes All* (WTA) model, the disparity x that results in the highest likelihood is assumed to be the most likely. The pixel-based algorithm provides a dense horizontal disparity map because the likelihood function provides a disparity candidate for every pixel coordinate in the images. The depth is inversely proportional to the disparity.

The expected improvement for the stereo correspondance with the Mahalanobis distance is compared to the reference method that was first introduced by Belhumeur [1]; it uses the Euclidean distance. The probabilistic model for the combined joint density assumes independent identically distributed Gaussian noise processes having mean zero and variance σ^2 , and is defined as:

$$P(\mathbf{z}_1, \mathbf{z}_2 | \gamma) = \frac{1}{(2\pi\sigma^2)^n} \prod_{i=1}^n \exp\left(-\frac{(z_{1,i} - z_{2,i})^2}{4\sigma^2}\right). \quad (3.2)$$

A rewrite of this function results in the baseline likelihood function for the benchmarks in this chapter. The monotonically decreasing function of the SSD with the squared Euclidean distance is given by:

$$p(\mathbf{z}_1, \mathbf{z}_2 | x) \propto K \exp\left(-\frac{1}{4\sigma_n^2} \|\mathbf{z}_1 - \mathbf{z}_2\|^2\right), \quad (3.3)$$

where $\|\mathbf{z}_1 - \mathbf{z}_2\|^2$ is the SSD and a predefined variance σ denotes an equal weight for every pixel contribution. The models of Cox et al. and Danjanović et al. are extensions of this model.

Similarly to Chapter 2.2.2, if it is assumed that both measurement vectors are normally distributed about their ideal value \mathbf{s} we can use the multivariate Gaussian distribution for the conditional probability $p(\mathbf{z}_i | \mathbf{s}, x)$ such that [3]:

$$p(\mathbf{z}_i | \mathbf{s}, x) = G(\mathbf{z}_i, \mathbf{s}, \mathbf{C}_i) = \frac{1}{\sqrt{(2\pi)^k \det(\mathbf{C}_i)}} \exp\left(-\frac{1}{2}(\mathbf{z}_i - \mathbf{s})^T \mathbf{C}_i^{-1} (\mathbf{z}_i - \mathbf{s})\right), \quad (3.4)$$

where k is the dimension of the measurement vectors \mathbf{z}_i , and \mathbf{C}_i are the covariance matrices associated with the residual errors $(\mathbf{s} - \mathbf{z}_i)$. Because the true value of \mathbf{s} is unknown, it has to be approximated by the maximum likelihood estimate $\hat{\mathbf{s}}$ obtained from the measurement pair \mathbf{z}_1 and \mathbf{z}_2 as follows¹:

$$\mathbf{s} \approx \hat{\mathbf{s}} = \mathbf{C}_2 (\mathbf{C}_1 + \mathbf{C}_2)^{-1} \mathbf{z}_1 + \mathbf{C}_1 (\mathbf{C}_1 + \mathbf{C}_2)^{-1} \mathbf{z}_2. \quad (3.5)$$

¹The derivation of the likelihood function and the covariance matrix is derived in detail by Cox et al. in [3].

Because the models for both cameras are assumed to be the same, from this point on it is assumed that the covariance matrices \mathbf{C}_1 and \mathbf{C}_2 are equal as well. The approximation for the unknown texture \mathbf{s} in equation 3.5 then simplifies to

$$\mathbf{s} \approx \hat{\mathbf{s}} = \frac{1}{2}(\mathbf{z}_1 + \mathbf{z}_2). \quad (3.6)$$

The approximation of the unknown texture is useful to complete the definition for the observations that will be used to estimate the covariance matrix. Substitution of the approximate unknown texture $\hat{\mathbf{s}}$ in the associated residual errors gives the expression:

$$(\mathbf{s} - \mathbf{z}_1) \approx \frac{1}{2}(\mathbf{z}_2 - \mathbf{z}_1). \quad (3.7)$$

The accurate ground truths for the disparity maps included in the datasets are used to extract a large set of measurements or observations from the left and the right images. The same window that is used to compute the stereo correspondance defines every observation as a block of pixels in the left image that corresponds to a shifted block of pixels in the right image. Together, the measurements of the errors form a set of m samples of random variables:

$$\frac{1}{2}(\mathbf{z}_{2,i} - \mathbf{z}_{1,i}), \quad \text{with } i = 1, \dots, m. \quad (3.8)$$

The estimation of the covariance matrix from a sample of m observations of n -dimensional random variables is then given by:

$$\mathbf{C} = \frac{1}{m} \sum_{i=1}^m (\mathbf{z}_{2,i} - \mathbf{z}_{1,i}) (\mathbf{z}_{2,i} - \mathbf{z}_{1,i})^T, \quad (3.9)$$

with $(\mathbf{z}_{2,i} - \mathbf{z}_{1,i})$ as the i -th observation of the residual error. The final derivation to obtain the likelihood function is proportionally equivalent to the first order expansion of the proposed likelihood function in Chapter 2.2.2. Because the magnitude of the likelihood is superfluous for the optimization it suffices to express the likelihood without the exact scaling constants. The simplified likelihood function has a certain constant K , includes a trainable matrix of weights \mathbf{C}^{-1} , and is given by:

$$p(\mathbf{z}_1, \mathbf{z}_2 | x) \propto K \exp \left(-\frac{1}{4}(\mathbf{z}_1 - \mathbf{z}_2)^T \mathbf{C}^{-1}(\mathbf{z}_1 - \mathbf{z}_2) \right). \quad (3.10)$$

The resulting expression includes a Mahalanobis distance as its metric, where the covariance matrix \mathbf{C} describes the (co)variances between all errors. Inversion of the covariance matrix; however, is required after estimation only once. The inverted covariance matrix is known as the precision matrix \mathbf{P} and holds the coefficients for all multiplication pairs. A final rewrite of the likelihood function $p(\mathbf{z}_1, \mathbf{z}_2 | x)$ results in:

$$\rho_{ij} = \mathbf{z}_i^T \mathbf{C}^{-1} \mathbf{z}_j = \mathbf{z}_i^T \mathbf{P} \mathbf{z}_j \quad (3.11)$$

$$p(\mathbf{z}_1, \mathbf{z}_2 | x) \propto \exp \left(-\frac{1}{4}(\rho_{11} + \rho_{22} - 2\rho_{12}) \right), \quad (3.12)$$

that will be used in the experiment to research the effects of a trained covariance matrix \mathbf{C} . The model from Cox et al. for the total cost for all pairs also includes costs for occlusions; however, in this experiment we will focus on the cost metric for matching pixel blocks only.

The covariance matrix has several important properties that guarantee invertibility of the covariance matrix \mathbf{C} to its counterpart \mathbf{P} , the precision matrix. As proved in Appendix A.4, every covariance matrix satisfies:

- \mathbf{C} is symmetric, thus $\rho_{12} = \rho_{21}$, and
- \mathbf{C} is positive-semidefinite, and therefore invertible if non-singular.

However, due to numerical instability, the high dimensionality of the window, and the in comparison small set of available samples available for estimation, this often results in an estimated covariance matrix that does not satisfy the strict conditions specified above. Unfortunately, a singular covariance matrix is unsuitable for the Gaussian distribution, but this problem can be solved by regularization.

Factorization and regularization

The estimated covariance matrix is a square n -by- n that is obtained with the expression of Equation 3.9. The definition of the estimator implies that the estimated covariance matrix is diagonalizable. Every diagonalizable matrix can be factorized into a canonical form, whereby the covariance matrix is represented in terms of eigenvalues and eigenvectors. Let \mathbf{A} be a square ($n \times n$) diagonalizable matrix, then \mathbf{A} can be factorized as

$$\mathbf{A} = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^{-1}, \quad (3.13)$$

where \mathbf{Q} is a square ($n \times n$) matrix whose i^{th} column is the eigenvector q_i of \mathbf{A} and $\mathbf{\Lambda}$ is a diagonal matrix whose diagonal elements are the corresponding eigenvalues, i.e. $\Lambda_{ii} = \lambda_i$. Corollary: the symmetric property of the estimated covariance matrix allows $\mathbf{Q}^{-1} = \mathbf{Q}^T$. Also, if none of the eigenvalues of \mathbf{A} is zero, its inverse is given by:

$$\mathbf{A}^{-1} = \mathbf{Q}\mathbf{\Lambda}^{-1}\mathbf{Q}^{-1}. \quad (3.14)$$

The inverse of $\mathbf{\Lambda}$ is easy to calculate because it is a diagonal matrix, and only the eigenvalues have to be inverted:

$$(\mathbf{\Lambda}^{-1})_{ii} = \frac{1}{\lambda_i}. \quad (3.15)$$

Therefore, once the singular eigenvalues are dissolved, it is easy to obtain the required precision matrix from the inverted regularized eigenvalues and the eigenvectors as $\mathbf{P} = \mathbf{C}^{-1} = \mathbf{Q}\mathbf{\Lambda}^{-1}\mathbf{Q}^T$.

The matrix \mathbf{A} is positive definite if and only if all eigenvalues of \mathbf{A} are positive. The regularization process forces the eigenvalues to be sufficiently positive by adding a small

fraction c of the largest eigenvalue λ_{\max} to all eigenvalues. The regularized matrix $\hat{\mathbf{A}}$ is then given by:

$$\hat{\mathbf{A}} = \mathbf{Q} \frac{(\mathbf{\Lambda} + c\lambda_{\max}\mathbf{I}_n)}{1 + c} \mathbf{Q}^{-1}, \quad (3.16)$$

where \mathbf{I}_n is the n -dimensional identity matrix. The regularized matrix \mathbf{A} converges to a diagonal matrix as the regularization constant increases and the diagonal matrix with eigenvalues converges to a scaled identity matrix:

$$\mathbf{Q}\mathbf{Q}^{-1} = \mathbf{Q}\mathbf{Q}^T = \mathbf{Q}\mathbf{I}_n\mathbf{Q}^T = \mathbf{I}_n. \quad (3.17)$$

If Equation 3.10 is substituted with Equation 3.16, we obtain the final likelihood function for the Mahalanobis distance:

$$p(\mathbf{z}_1, \mathbf{z}_2 \mid x, c) \propto \exp\left(-\frac{1}{4}(\mathbf{z}_1 - \mathbf{z}_2)^T \hat{\mathbf{C}}_c^{-1}(\mathbf{z}_1 - \mathbf{z}_2)\right), \quad (3.18)$$

where $\hat{\mathbf{C}}_c$ is the regularized covariance matrix with regularization factor c .

Once the regularization c approaches infinity, Equation 3.18 (the Mahalanobis distance) converges to Equation 3.3 (with the Euclidean distance) as follows:

$$\lim_{c \rightarrow \infty} \exp\left(-\frac{1}{4}(\mathbf{z}_1 - \mathbf{z}_2)^T \hat{\mathbf{C}}_c^{-1}(\mathbf{z}_1 - \mathbf{z}_2)\right) \quad (3.19)$$

$$= \lim_{c \rightarrow \infty} \exp\left(-\frac{1}{4}(\mathbf{z}_1 - \mathbf{z}_2)^T \mathbf{Q} \left(\frac{\mathbf{\Lambda} + c\lambda_{\max}\mathbf{I}_n}{1 + c}\right)^{-1} \mathbf{Q}^{-1}(\mathbf{z}_1 - \mathbf{z}_2)^T\right) \quad (3.20)$$

$$= \exp\left(-\frac{1}{4}(\mathbf{z}_1 - \mathbf{z}_2)^T \mathbf{I}(\mathbf{z}_1 - \mathbf{z}_2)^T\right) \quad (3.21)$$

$$= \exp\left(-\frac{1}{4}\|\mathbf{z}_1 - \mathbf{z}_2\|^2\right). \quad (3.22)$$

From this result follows that we can compare the Mahalanobis distance to the Euclidean distance with the same likelihood function if the regularization is varied enough to reach convergence. The regularization parameter allows us to observe the stereo correspondence performance from a Euclidean distance metric to a proper Mahalanobis distance metric.

3.2 Method of evaluation

The theoretical background for the hypothesized improvement of the likelihood function is discussed in the previous section. It was proposed that an extension of the likelihood function with a Mahalanobis distance instead of a Euclidean distance as metric could increase matching performance for difficult regions. This section discusses the implementation and design decisions used to the the validity and usefulness of the approach.

We expect that for the occluded and distorted parts of the images — which remained difficult to find proper correspondance for — the matching performance will improve. The main goal is to compare the correct correspondances of Equation 3.10 (Mahalanobis) to the correct correspondances produced with Equation 3.3 (Euclidean).

Results for the experiment are obtained by implementation of Equation 3.18 where the covariance matrix is regularized with different constants. It was shown in Equation 3.19 that a large constant converges to a Euclidean distance. Therefore, a parameter sweep is applied for regularized covariance matrix, where the regularization factor ranges from 10^{-6} (near singular covariance matrix) to 10^3 (approximately the Euclidean distance).

Unfortunately, the addition of the trained covariance matrix to the likelihood function is not free in terms of computational power. The required matrix multiplications of the measurement differences ($\mathbf{z}_1 - \mathbf{z}_2$) with the precision matrix \mathbf{P} for every likelihood computation increases the computational complexity, and thereby decreases the degrees of freedom to explore in the available timeframe. Therefore, some presets have to be chosen carefully without compromising the results of the experiment. The experiment is run with a fixed window size on one image scale.

3.2.1 Data selection

To test for a possible improvement of correct matches with the Mahalanobis distance, a dataset is required with a precise and proper ground truth. Also, we prefer a varied set of scenes with a very large depth-of-view and good illumination.

The vision group from the Middlebury College provides several very good datasets that are used globally to compare common and state-of-the-art stereo-vision algorithms. Each subset from the dataset consists of seven high-resolution (1240-1396 by 1110 pixels) views taken under three different illuminations and three different exposures. Very accurate disparity maps are provided for two of the seven views to provide good benchmark possibilities. All images have been pre-processed to remove radial distortions and to rectify the images [11, 13]. A subset of eight datasets is selected from the entire dataset to benchmark for a varied spread of scenes. The optimal settings for the illumination (subset 1) and exposure (subset 2) are selected. An overview of the selected images and accompanying disparities is given in Appendix C on page 85.

The Middlebury datasets are used to generate the covariance matrices as well as for the benchmarking of the different algorithms. Three covariance matrices are generated for every dataset to benchmark for different cases:

- A generic matrix with the leave-one-out method: the other datasets are used as training datasets. This method should generate a realistic experiment for scenes and images not yet known.
- A self-trained matrix is generated from the dataset (to be tested) itself. If the self-trained matrix produces better stereo correspondance than a generic matrix, it tells us that there is room for improvement. Otherwise, the generic matrix appears to approach an optimum.

- Because we expect the performance to improve in difficult parts of the images with (partial) occlusions, we train a special matrix on pixel blocks with these effects. The model can be extended with segmentation if the gain in performance is significant.

3.2.2 Window size and image scaling

The computational requirements for the experiments with the Mahalanobis distance forces us to limit the degrees of freedom in the simulation. Large images imply more pixels to be processed, but contain more information. Also, larger windows increase the computational complexity significantly, because the number of pixels in the measurement vectors and the rank of the covariance matrix increases in a squared order with respect to the window size. Consequently, the required number of operations is of the second order with respect to the number of pixels. Therefore, the effect of the window size on the complexity is of the fourth order: $T(d) \in O(d^4)$, with d as the base of the window. However, the window should be large enough to support different weights for the pixel intensities within the window.

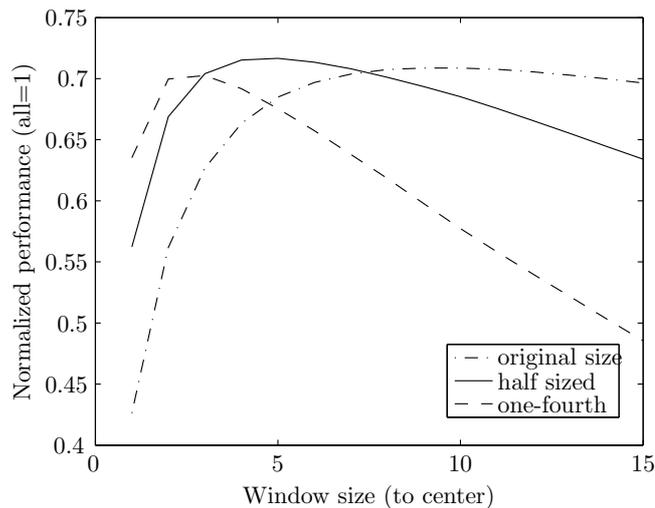


Figure 3.1: Stereo correspondance performance for different window sizes and image scales.

A quick comparison of window sizes and smaller image scales for the SSD is shown in Figure 3.1. Although the original images contain more information, a trade-off is made to reduce computation time. Half-sized images in combination with a window size of 5 provides good testing scores that can be used as a baseline. The half-sized images have only 620 to 698 rows with 555 pixels each that have to be tested for each case. The window size of 5 describes the size of the window in each direction measured from the center pixel, and implies a window size of $(5 \times 2 + 1)^2 = 121$ pixels with three primary colors each. The color images are chosen to take advantage of the correlations between

color values of the pixels. The dimension of the covariance matrix is therefore 363 by 363, and the measurements z_i contain 363 pixel intensities each.

3.2.3 Occluded regions

Most errors in the stereo correspondance seem to occur near occluded and distorted regions. For estimation of the regular covariance matrix, visible regions are used as well as the partially occluded regions. We suspect that a small improvement in performance is possible if we train a specific covariance matrix for these effects. In future models, special covariance matrices can be incorporated in the system with the use of image segmentation. However, segmentation is useful only if the number of correct correspondances does indeed increase with special covariance matrices.

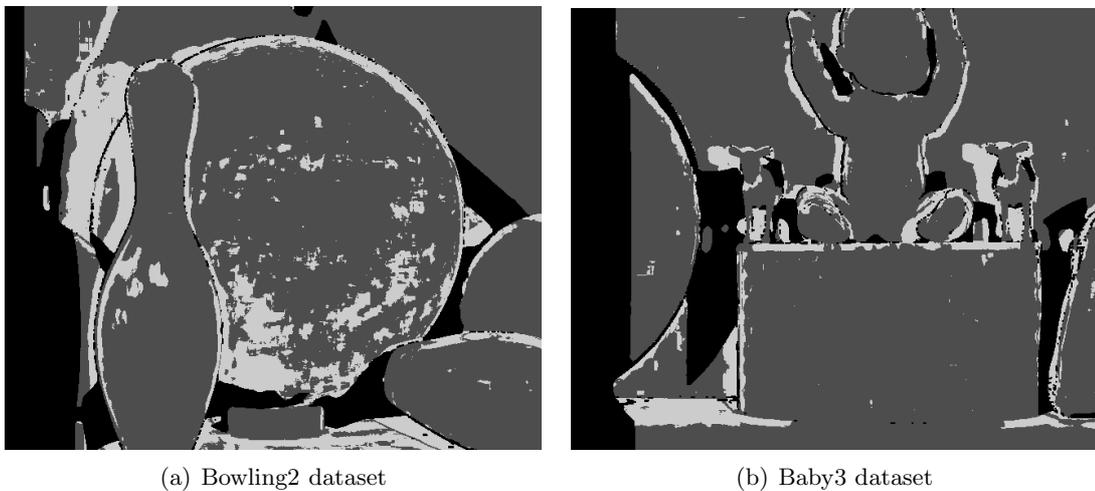


Figure 3.2: Correct and incorrect correspondances near occluded regions for the likelihood function with an Euclidean distance. Black pixels are occluded, grey pixels are correctly matched and light pixels indicate the room for improvement.

Figure 3.2 illustrates where the classic algorithm with the Euclidean distance has difficulties to obtain the correct correspondances. Likelihoods near occlusions appear to be less reliable than likelihoods of pixels in continuous surfaces of objects. The matching is difficult for windows that contain regions with different depths. The generic covariance matrix is mostly trained on smooth surfaces; therefore, we expect that it is worthwhile to research the possibilities of a covariance matrix that is trained on these different effects.

For this experiment, about 10% of the measurements contains 10 to 50 percent of occluded pixels for half sized images and a window size of 5 (11×11). Larger windows are more prone to include occlusions within the local measurement range; however, a few occluded pixels will have less impact on the final result. The scaling of the image affects the distribution, because windows of the same size cover a larger area of the scene.

The difference in performance is tested by using a special covariance matrix that is estimated in the same way as the regular covariance matrix for windows, but only on

windows that are moderately occluded. The additional experiment is run on the subset of windows that fit the category of 10% to 50% occlusions, both with the regular covariance matrix and the covariance matrix that is trained on occlusions only. Especially this subset of windows seems to suffer incorrect likelihoods in comparison to the ground truth. Accurate disparities near occlusions are important to define the boundaries of objects. Therefore, a small increase in local performance for the difficult regions benefits the total performance.

For this subset of 10% of the measurements, a special covariance matrix is generated and the results are compared to the generic simulations. The goal is to test whether special training for occluded regions improves performance for the likelihood function with the Mahalanobis distance.

3.2.4 Experiment

The planned simulations can be broken down in two important parts. First, the algorithm requires a trained covariance matrix, either from a training set or from the dataset itself. Second, the trained covariance matrix is used in the evaluation part of the simulations to obtain the most likely disparity map for the dataset to benchmark.

Both steps require a considerable number of computations. Therefore, the work is split in lots of chunks to distribute the work and save intermediate results. Because of its convenient prototyping environment and large set of high level instructions, the algorithms are implemented in Matlab. There are no additional requirements or toolboxes to run the simulations. A description of all scripts and implemented functions can be found in Appendix B.

Generation of the covariance matrix

The first step of the experiment consists of the generation of the covariance matrices. Equation 3.9 in the previous section stated the formal definition to estimate a covariance matrix.

The measurements that are used to estimate the covariance matrix contain the differences of the pixel intensities within the window between the left and the right image; as described in Equation 3.7. The ground truth of the dataset is used to determine which measurements will be taken into account, and which will be left out. It is also used to pair the window in the left image with the correct window in the right image. Measurements that contain too much occlusions, or are too close to the border of the images are not used. Also, windows without ground truth information cannot be used, because the disparity between the windows is not known.

The simulations for the datasets in combination with the respective covariance matrices incorporate an important parameter for the regularization of the covariance matrix. Because most estimated covariance matrices appeared unstable due to reasons described in Section 3.1, the regularization method of Equation 3.18 is implemented to provide robust matrices that satisfy the properties of the covariance matrix, and to make it invertible. The matrix is factorized with spectral decomposition in a square matrix with

all eigenvectors and a diagonal matrix whose diagonal elements are the corresponding eigenvalues. The regularization factor defines an addition to the value of all eigenvalues, based on the largest eigenvalue. This parameter controls the regularization and has significant impact on the covariance matrix and the performance of the algorithm; therefore, all experiments are run for a range of regularizations.

Disparity computation

In the second step, the performance of the algorithm is evaluated by implementation of Equation 3.12. Because input images from the Middlebury dataset are already pre-processed, rectification of the inputs can be omitted. However, different types of covariance matrices are to be tested on the datasets, and therefore have to be generated before stereo matching can commence.

The disparity map is built row-by-row and pixel-by-pixel. For every pixel, the maximum likelihood function is used to generate a set of probabilities that the local measurement in the left image matches a candidate set of shifted measurements in the right image. These probabilities are used to determine the most likely disparity map. The most likely pair of measurements indicates the most likely horizontal disparity. The *Winner Takes All* (WTA) method is implemented to measure and compare performance, but other algorithms can also be used in which the optimization of the maximum likelihood is applied to a larger range of pixels. However, this type of post-processing requires tuning and clouds the exact performance of the maximum likelihood function which would make it difficult to compare it objectively. Hence, the implementation with the WTA method is not optimal, but the goal is to explore the possibilities and the usefulness of a Mahalanobis distance.

The estimated disparities are used to compare the difference in correct matches for the selected range of regularization values. Appendix B summarizes the implemented algorithms and visualization scripts.

Preliminary experiments

The primary goal of the experiment is to explore the difference in the number of correct matches between different algorithms and methods. However, interesting information unveils itself during the initialization process in which the covariance matrix is generated. The structure of the precision matrix (the inverse of the covariance matrix) gives an impression of the weights allocated to the different elements. Therefore, the first step in the experiment is to present and discuss the structure of a generated precision matrix.

- We expect that the weights on the diagonal contribute the most. In other words, windows are mostly mapped one-to-one where distortions are not dominant.
- The distance of a pixel within a window is expected to influence the weight of the measurement pair. Pixels further away from the center are more vulnerable to distortions and occlusions.

- Mutual information between the primary colors of the pixels is expected.

Also, spectral analysis of the covariance matrix could give an impression for the results we can expect. The difference in the size of the eigenvalues gives an indication for the variation in variances and the effect of the applied regularization. The individual eigenvectors (ranked by eigenvalue) show which patterns contribute, and which do not.

Protocol

The purpose of the experiment is to test the following hypotheses:

1. The Mahalanobis distance produces better results than the Euclidean distance. (Equation 3.18 versus Equation 3.3.)
2. A covariance matrix trained on the dataset itself results in better correspondances between the left and the right image. (There is room for improvement for the generic covariance matrix generated on other datasets.)
3. A special covariance matrix for the partially occluded regions improves performance for these regions.

All three experiments require the Middlebury dataset[11] for the stereo images and depth groundtruth (Appendix C). The required Matlab scripts that contain the implemented algorithms are described in Appendix B.

Experiment 1: *Mahalanobis distance*

1. Select a dataset.
2. Generate a covariance matrix from the remaining datasets with Equation 3.9.
3. Apply different regularizations to the covariance matrix.
4. Compute the likelihood for every possible pixel pair with Equation 3.18.
5. Extract a disparity map from the likelihoods with the WTA method.

Experiment 2 (extension of experiment 1): *Covariance matrix generated from the same dataset*

2. Generate the covariance matrix only from the dataset that is selected for processing.

Experiment 3 (extension of experiment 1): *Occlusions*

- 2a. Generate a covariance matrix from the remaining datasets, but on windows with moderate occlusions only (10% to 50% occluded).
- 2b. Generate the covariance matrix from the same dataset (equal to experiment 2), but also on 10% to 50% occluded windows only.

4. Compute the likelihoods for windows where the window in the left window has 10% to 50% occlusions.
5. Extract the disparity map only for the pixels that correspond to the windows of (4.)

Controls:

- Window size is chosen at 5 pixels from the center pixels: 11×11 windows.
- The images are scaled by a factor two to limit the required computing power.
- The regularization constant is swepted from very small (10^{-6}) to very large (10^3): convergence to euclidean distance.

Data interpretation:

The covariance matrix is generated in the first stage of the experiment. Because the inverse of the covariance matrix (the precision matrix) can be seen as a matrix of weights for the measurement residuals, it is inspected before running the second stage of the experiment. The diagonal of the precision matrix holds the primary weights and is expected to have the highest energy. The diadonal elements have the same constant values for the euclidean distance. For the mahalanobis distance, however, we expect that the diagonal is dominant, but models the importance of the pixels within the windows. The elements on the diagonal for pixels in the center are expected to have more energy than pixels near the border (of the window). Also, we expect that correlation between the primary colors becomes visible on the diagonals between the different colors. The content of the precision matrix confirms the chosen model if it satisfies our expectations. Spectral analysis shows the importance of the different eigenvectors by the value of their accompanying eigenvalue.

The second stage of the experiment returns likelihoods for all possible window combinations. We compare the performance of the different settings and experiments by extracting the most likely window pair (best matches; WTA). These window pairs combined form the estimated disparity map, which is inversely proportional to the depth. The total number of mismatches is obtained by comparison of the disparity map with the ground truth of the dataset. The raw number of mismatches is given in Appendix D.

In order to compare the difference in performance properly, we normalize the number of mismatches with the euclidean distance as baseline. The euclidean distance is indicated in the graphs by the very large regularization constant of 10^3 and has a relative performance of 1. A relative score larger than one indicates that there are more incorrect matches, and is therefore worse than the euclidean distance. A lower score equals less incorrect matches, and therefore represents better performance. For example, a score of 0.9 equals 10% less incorrect matches

The first experiment contains the results for the individual datasets, as well as the average performance over all datasets. For the additional experiments with self-training and training on occlusions, only the mean performance is used, and compared to the

mean performance of the first experiment (baseline). Again, Lower scores indicate less incorrect matches.

3.3 Results

This section presents the results obtained by the experiments described in the protocol of Section 3.2.4. The raw data of the experiments (the number of incorrect matches per dataset) is bundled in several tables in Appendix D on page 89. Also, additional images that hold the individual gains and losses of correct pixels are given in the supplemental chapter.

3.3.1 Preliminary: covariance matrix

The precision matrix for the maximum likelihood function is generated according to the recipe of Section 3.2.4. The disparities from the ground truths of the remaining datasets are used to extract a large set of samples from which the covariance matrix can be estimated. Consequently, the precision matrix is obtained from the covariance matrix after regularization of its eigenvalues. In the previous section, a few expectations were given in the protocol of the experiment. We expect that the assumed model has some noticeable consequences for the shape of the elements in the precision matrix.

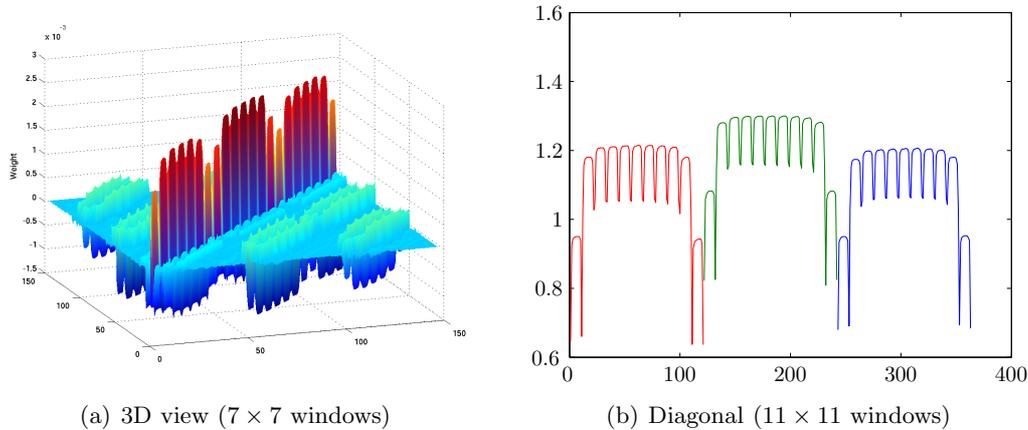


Figure 3.3: The precision matrix built from residuals of the Bowling2 dataset.

Figure 3.3(a) shows an example of a generated precision matrix for 7-by-7 sized windows. The protocol of the experiment specified 11-by-11 windows for all experiments, but for a proper 3D-representation of the weights this window size appeared to be produce an indistinct graph. With 7-by-7 windows, the effects present in the precision matrix are exactly the same; however, it is much clearer the recognise the seperate rows, columns and colors. The diagonal elements of the matrix represent the weights for the residuals between the same colors and pixels. It consists of three parts for red, green,

and blue respectively. Every column in the window is recognizable as a small slope of seven pixels. It is clear that the outer pixels have considerable lower weights than the center pixels, which confirms our first prediction. Also, it is clear that there is significant correlation between the color values of the same pixels. Weights for pixels in the left window that are shifted one position in the right window even appear to have a negative weight. Most other elements are near-zero, and can therefore be left out in simplifications of the algorithm. The simplification will have a very small impact on the performance, but reduces the computational complexity significantly.

The diagonal elements for a precision matrix generated with an 11-by-11 window are given in Figure 3.3(b). Similar to the 3D-representation, the columns are stacked together ordered from left to right for each color. The weight of the residuals falls off from the center to the border of the window. Also, it appears that for every dataset more weight is allocated to the color green. It is possible that this effect is caused by properties of the cameras.

Eigendecomposition

With spectral decomposition, the covariance matrix can be factorized and represented by its eigenvalues and eigenvectors. The eigenvalues determine the scaling of the accompanying eigenvector. A properly estimated covariance matrix is invertible and positive definite (Appendix A.4). This constraint implies that all eigenvalues positive and sufficiently large. The eigenvalues for the covariance matrices that belong to all datasets used in this experiment are given in Figure 3.4. (The covariance matrix for the `Bowling2` dataset was given in Figure 3.3.)

Unfortunately, one of the eigenvalues is very small and approaches zero. Hence, it is clipped in the graph to keep the range of the logarithmic scale useful. This indicates that regularization is required to obtain the proper precision matrix from the covariance matrix. Also, note that the range of the eigenvalues is quite large and indicates that there is a significant difference in the scaling of the eigenvectors. This implies that the estimated covariance matrix for the mahalanobis distance differs from the euclidean distance, and that we should expect differences in performance.

The eigenvectors for the `Bowling2` dataset are shown in Figure 3.5. The image contains all eigenvectors of the covariance matrix, with every block as an individual eigenvector transformed back to two dimensions. The dominant eigenvector is placed in the top-left corner, and the accompanying eigenvalue decreases column-wise to the bottom, and then column-by-column to the right. The smallest eigenvalue of the *covariance matrix* belongs to the top-left eigenvector; however, because the precision matrix is the inverse of the covariance matrix, the top-left block is the most important and the bottom-right block is the least important for the similarity measure.

The constant eigenvector has the largest contribution to the precision matrix. Furthermore, it appears that high-frequency eigenvectors contribute the most to the precision matrix weights. Also, the (large) eigenvectors show a clear drop-off in importance from the center of the window to the outside of the window. Hence, it takes the shape of a two-dimensional Gaussian. The tilted planes and low-order eigenvectors (on the right

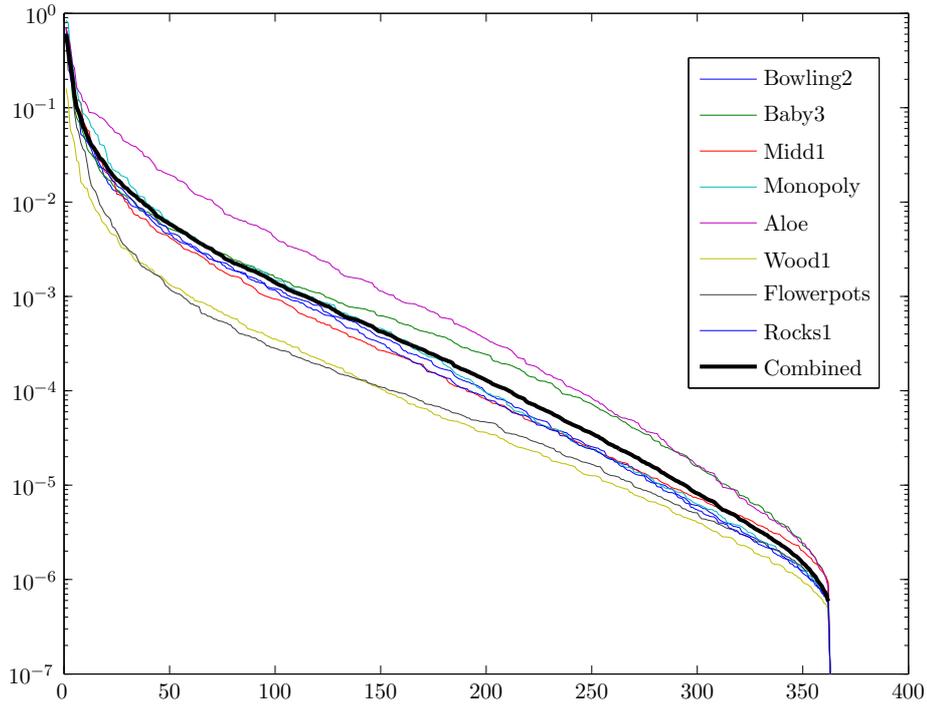


Figure 3.4: Eigenvalues of the covariance matrix built from the available datasets. The covariances are generated from the half-size images with 11 by 11 windows.

side of the image) contribute most to the covariance matrix, and therefore implies that these are the least important eigenvectors for the precision matrix.

In conclusion, it appears that the trained covariance matrix confirms the predictions from the protocol. Therefore, the Mahalanobis distance is still expected to improve the matching performance for the selected datasets.

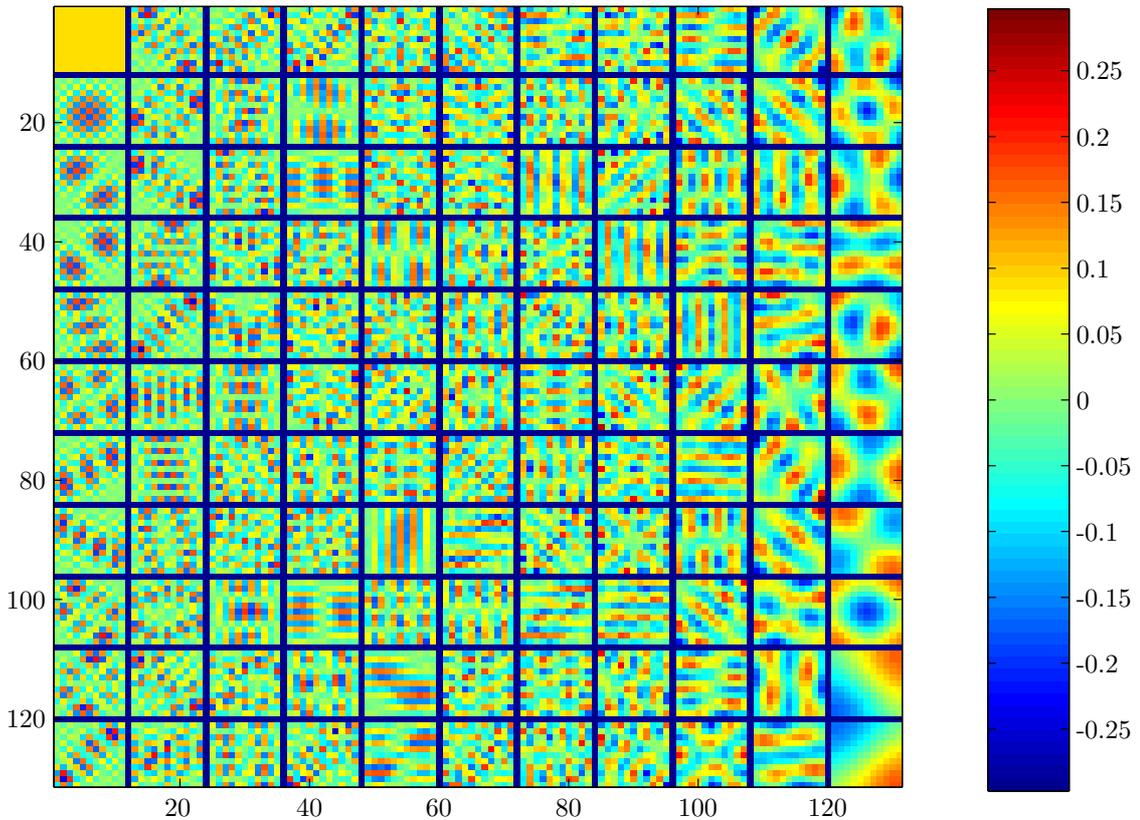


Figure 3.5: Eigenvectors of the covariance matrix for the `Bowling2` dataset. The covariance matrix is generated from gray-scale images with window size = 11×11 .

3.3.2 Mahalanobis distance vs. Euclidean distance

The number of correctly found disparities is tested with the algorithms described in Section 3.2.4 and compared to the ground truths of the selected datasets from the Middlebury dataset.

The first experiment of the protocol supplied a recipe to compare the performance of the Mahalanobis distance versus the Euclidean distance. Figure 3.6 shows the results for this experiment. For a regularization constant of 10^3 , the likelihood function has converged to the Euclidean distance. The number of mismatches is shown relative to the Euclidean distance. A lower score indicates less incorrect matches, and therefore a better stereo correspondence performance.

In Figure 3.6, the experiment produces varying results for the individual datasets in terms of relative decrease of incorrect correspondences. However, the optimal regularization constant that leads to the least number of incorrect matches is quite stable about 10^{-2} . For this regularization, the mean of the relative performance shows a clear optimum with an average increase in performance of almost 4%. Unfortunately, it ap-

pears that it remains difficult to provide a good correspondence for every position in the scene.

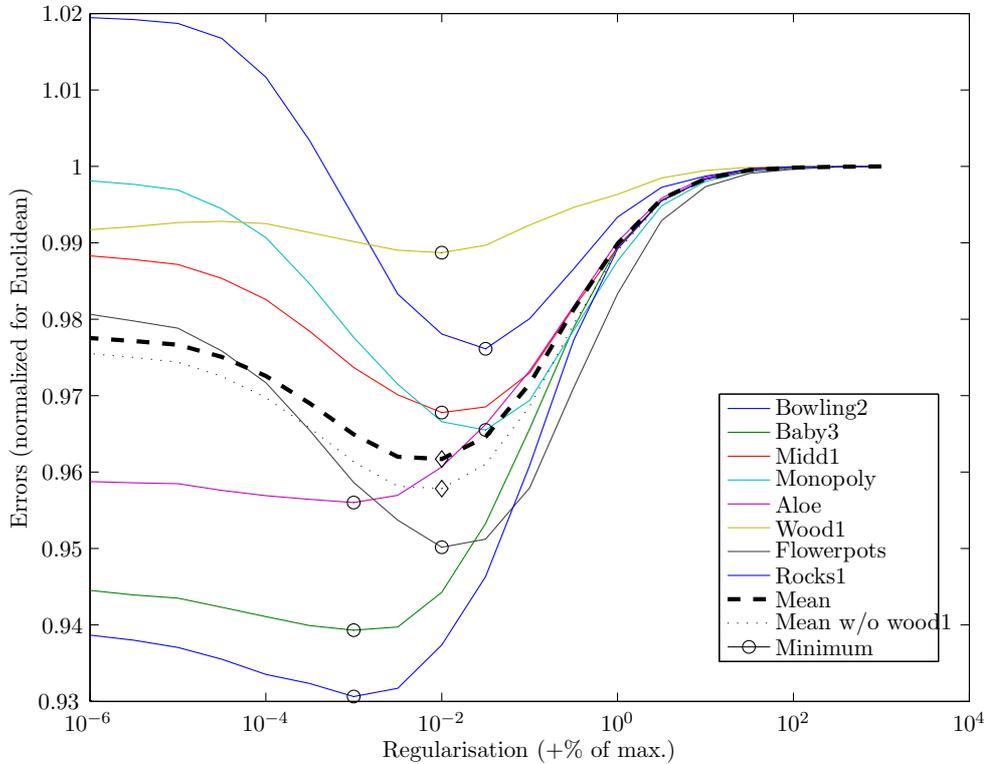


Figure 3.6: Number of errors for a regularized covariance matrix normalized to the Euclidean distance.

However, if we take a closer look at the gains and losses, it becomes clear that some segments are very difficult to process. Regions with lots of occlusions or regions with very little information are amongst the most difficult. Figure D.1 in Appendix D shows images for all processed datasets in which the exact gain and loss in performance occurs. Figure 3.7 is a larger version of one of these datasets and shows the increase in performance for the **Bowling2** dataset. Correct matches for both the Mahalanobis distance and the Euclidean distance are colored gray. Correct disparities gained with the Mahalanobis distance are colored green, and the red color indicates disparities that are computed correctly with the Euclidean distance, but produce erroneous disparities with the Mahalanobis distance. Furthermore, the blue color indicates regions for which no ground truth is known (due to occlusions or out-of-range disparities). Finally, the pixels for which the correspondences are computed incorrectly, but are geometrically possible to compute correctly, are colored black. Yellow pixels are the same as black pixels, but with the complication that there is almost no texture available in the scene to work with (e.g. a white background). Hence, black and yellow indicate the room for improvement.

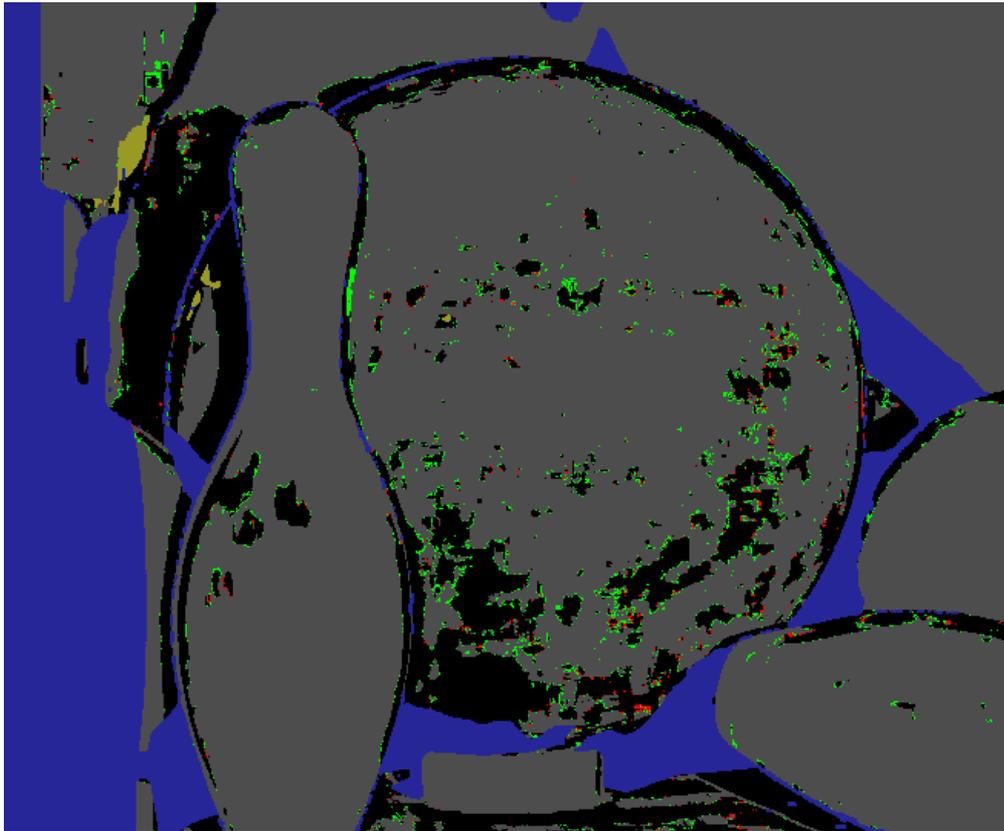


Figure 3.7: Difference for the Bowling2 dataset between the Mahalanobis distance for optimal regularization and the Euclidean distance.

Although there are still a lot of pixels with incorrect disparities, it appears that the Mahalanobis distance does lead to a lot of new important correct correspondences. Especially the gains near occlusions might prove valuable with additional post-processing. With the WTA method, a new correct match does not influence its region. However, if we use the information of neighboring correspondences, it is possible to produce an improved disparity map in which very unlikely jumps in depth are discarded. Combined with object segmentation, the small percentage of newly gained correspondences could lead to much more accurate disparity map.

3.3.3 Results for self-training

The second experiment of the protocol describes a re-run of the experiment with a covariance matrix generated on the dataset and the ground truth itself. This self-training allows us to explore the leeway for further performance improvement if the covariance matrix is optimal for the testcase at hand. All simulations are repeated for precision matrices that are self-trained on the ground-truths. In Figure 3.8 the combined results of

the eight datasets is presented. For optimal regularization, the difference in performance appears to be negligible, but it is present for the entire range of regularizations. Therefore, it can be concluded that the generated precision matrices used for the simulations presented in Figure 3.6 approach the maximum attainable disparity performance. Table 3.1 gives the exact differences of (in)correct matches.

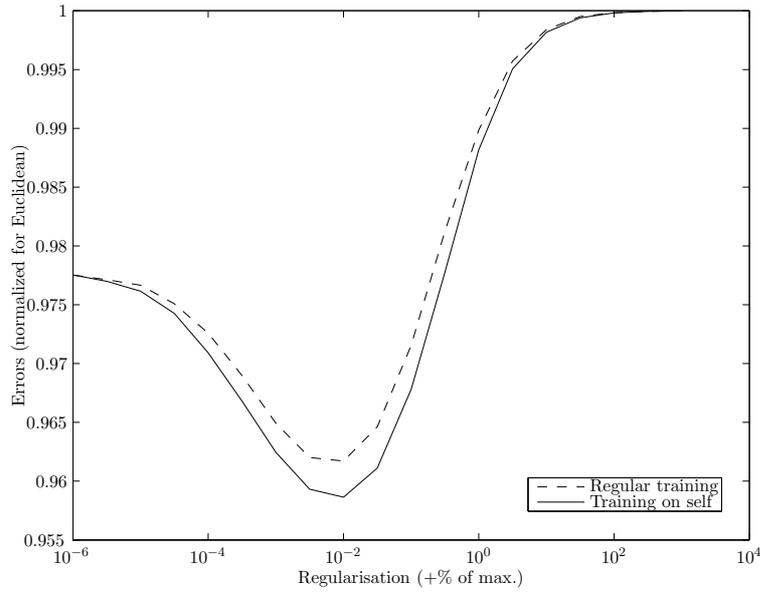


Figure 3.8: Number of errors for the covariance matrix trained on the test image itself. The window size is 11-by-11 applied on the half size images.

Dataset	Regular	Self	Net. gain	Samples	Incorrect
Bowling2	158	209	+51	299698	54832
Baby3	149	253	+104	281712	29448
Midd1	602	708	+106	320494	185852
Monopoly	423	662	+239	337470	178075
Aloe	474	508	+34	306092	61026
Wood1	176	180	+4	332762	43469
Flowerpots	248	567	+319	259313	47236
Rocks1	134	113	-21	289511	17918

Table 3.1: Difference in performance for the Mahalanobis distance with a covariance matrix trained with the leave-one-out and on the dataset itself method.

3.3.4 Results for occluded regions

Preliminary experiments indicated that the likelihood function contained a lot of incorrect correspondences near occlusions. The occlusions and discontinuities result in windows in the left and the right image where pixels cannot be mapped one-to-one. Therefore, those pixels will result in high contributions to lower the likelihood.

The third experiment in the protocol proposed to test if covariance matrices can be trained properly to include occlusion effects. Contrary to a generic covariance matrix, it is trained on windows that fit the category of partially occluded windows. It was expected that the prediction of correct disparities is especially difficult for pixels bordering occlusions. Unfortunately, the hypothesis that special training and segmentation improves performance appears to be false. The results of the experiment are shown in Figure 3.9. The relative performance of incorrect likelihoods is given for covariance matrices trained on other datasets, as well as on the dataset itself. Both cases result in negligible differences in performance between training on all windows and on partially occluded windows only.

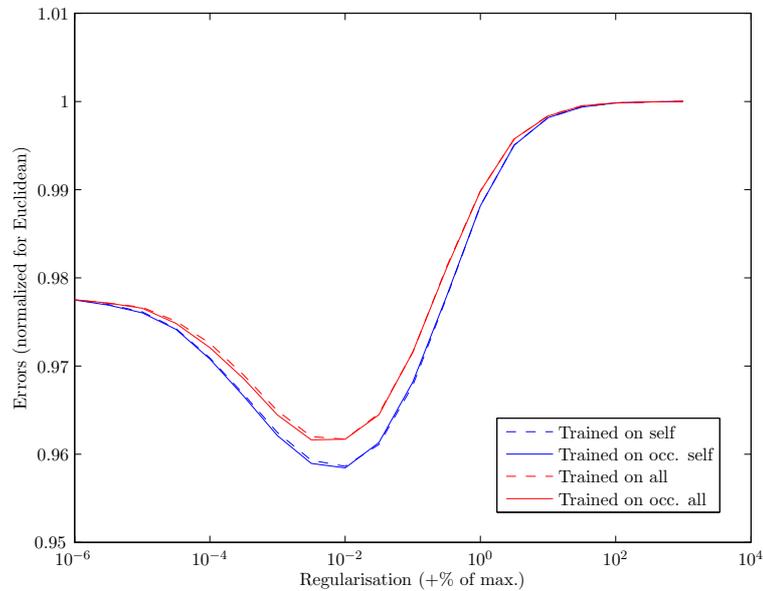


Figure 3.9: Comparison of different covariance matrices for 11-by-11 windows evaluated on half size images of the Middlebury dataset.

3.4 Discussion

This chapter is dedicated to research the usefulness and practical implications of a maximum likelihood function that uses Mahalanobis distance instead of a Euclidean distance. The Mahalanobis metric introduces a covariance matrix to the likelihood function that

has to be generated from training datasets. The proposed model of Section 3.1 is tested with the protocol of Section 3.2.4.

The first part of the experiment was implemented to answer the primary research question for this chapter by researching whether the Mahalanobis distance produces less incorrect correspondences than the classic Euclidean distance. The estimation of high-dimensional covariance matrices often leads to unstable solutions that are not invertible to the required precision matrix. This problem is overcome by regularization of the covariance matrix; however, the intensity of the regularization has a direct effect on the performance of the likelihood function. For this experiment, a regularization that adds about the order of one percent of the maximum eigenvalue to all eigenvalues appeared to give the best results. Eight datasets from the Middlebury dataset were processed and the average reduction of incorrect matches appeared to be about 4%. Unfortunately, this reduction of incorrect matches is relatively quite small; although it generally implies several tens of thousands of extra correct correspondences in the disparity map. Also, the extra correct correspondences occur in difficult regions that contain partially occluded windows.

In the second experiment, generation of the covariance matrix on the dataset itself resulted in slightly better performance than performing the experiments with a generic matrix. Therefore, the generic covariance matrix does not appear to be an optimal solution for the likelihood function. In the third experiment, a simulation targeted to explore whether the performance for partially occluded windows can be increased resulted in no observable difference in performance.

The hypothesis that the Mahalanobis distance improves the performance compared to the Euclidean distance is validated. The initial assumption that pixels in both windows have an additive white noise to the unknown texture indeed seems to be valid. However, the small reduction of incorrect matches comes at the cost of a severe increase in computational complexity. Because the addition of the covariance matrix requires a full matrix multiplication with high-dimensional matrices, the number of required multiplications is a few orders higher than for the Euclidean distance.

Suggestions

In the preliminary analysis of the precision matrix it became clear that most elements contribute very little to the likelihood function. A selective use of a several diagonal elements decreases computational complexity, but probably doesn't lower the matching performance of the likelihood function. Also, large matrix multiplications can be accelerated quite a lot by exploiting the highly parallelized structure of modern graphics cards.

Lastly, a variable window size that adapts to borders of the images might be able to cover a larger surface of both images, and thereby producing a larger disparity map for the available pixels. In the current implementation, the entire window has to fit the images taken by both cameras.

Conclusions and Discussion

The goal of this thesis was to examine the likelihood function that was introduced in the paper of Damjanović et al.[4]. Several research questions defined the framework of the research project. The contents of the thesis are largely divided in two parts: the first part focused on the solution for and improvement of the new likelihood function. The second part describes the experiment that was designed to test the difference between the Mahanalobis distance and the Euclidean distance as metric for the likelihood function. In the first section of this chapter, the research questions are answered. The final section gives recommendations for future research.

4.1 Research questions

Q1 *How can the algorithm as introduced by Damjanović et al. be improved, taking into account the complication of the analytical derivation?*

Halfway the derivation, it became clear that an error was made in the marginalization of the unknown texture. The correct solution of the marginalization of the unknown texture introduced a very high order pole in the expression. The final solution is obtained by marginalization of the gains. Unfortunately, the high order pole prevents a proper final solution. Although we did not succeed in improving the likelihood function, it is an important result that the error in the derivation has been corrected. We expect that the integral should be solvable with some transformations and Cauchy's integral formula.

(Q1a) *Are approximations sensible to obtain a solution for the likelihood function?*

No, it appears that the correction complicates the term that was approximated in the paper. It is an essential part of the likelihood function, and care should be taken with respect to equating the gains to constants in certain situations. However, the scaling constant of the likelihood function can be discarded without consequence, because only the difference in likelihoods is used.

In our quest for a solution, we have tried to approximate certain parts of the expression with a Taylor expansion. The Taylor expansion is very complex for even low order expansions, and it appeared that even high order expansions introduce severe approximation errors. Therefore, we conclude that it is not a useful method to obtain a solution for the likelihood function.

(Q1b) *How is the new likelihood function generalized for more than two camera views?*

The n -view generalization of the likelihood function is an extension of the two-view likelihood function. Because we were unable to find a proper complete solution for the two-view likelihood function, we were also unable to provide a generalized likelihood function. However, it is shown in Section 2.5 that the derivation of the generalized likelihood function is more abstract, but not significantly harder to solve. Therefore, we conclude that an n -view variant of the likelihood function can be obtained once the two-camera solution is found.

Q2 *Does a simplified version of the likelihood function improve performance?*

It was concluded in Chapter 2 that a simplified form of the new likelihood function equals a monotonically decreasing function of the SSD. In Chapter 3, the difference in performance between the Mahalanobis distance and the Euclidean distance is evaluated. Literature research revealed that it — the likelihood function with a Mahalanobis distance as its metric — is an extension of the similarity measure that was used in both the paper of Cox et al.[3] and the paper of Belhumeur[1].

The similarity measure with the Mahalanobis distance produces better results for all eight tests in the experiment of Section 3.3. However, it appeared to be necessary to regularize the estimated covariance matrices to avoid negative eigenvalues and keep it invertible.

(Q2a) *How significant is the reduction of errors in the stereo correspondence?*

The performance of the Mahalanobis distance is dependant on the amount of regularization. For good regularization, the number of incorrect matches compared to the Euclidean distance was about 4% lower. Although 4% does not seem much at first, a closer inspection revealed that most the the new correct matches lie close to discontinuities and occlusions. This gain in performance is important, because these pixels will have a significant contribution to the post-processing algorithms, such as the one-dimensional Viterbi or the two-dimensional Belief Propagation.

The similarity measure with the Mahalanobis distance, however, is far more computationally complex than the Euclidean distance.

(Q2b) *Is it possible to improve the matching performance of the simplified likelihood function?*

The matching performance does depend on the specific covariance matrix. Figure 3.8 shows the results of the experiment where the covariance matrix

is built on the dataset itself. It appears that the generic matrix is not yet optimal for every image; however, the difference is very small.

It was also hypothesized that the use of different covariance matrices for different segments of the scene could improve performance. A lot of mismatches occur near occlusions. Therefore, we have decided to run extra experiments on the occluded parts of the datasets. The difference in performance between a covariance matrix generated on all data and on the occluded regions only appeared to be negligible, as shown in Figure 3.9. We conclude that it is not useful to segment the image and use specific covariance matrices for every segment type.

4.2 Recommendations

The main recommendation is to find a proper and complete solution for the likelihood function. However, it has become clear during this project that this will not be an easy task. It might be useful to approximate the expression that was obtained by the marginalization of the unknown texture. Special care should be taken to reduce the approximation errors as much as possible. We expect that it is possible to obtain an analytic solution by transforming the expression to polar coordinates. After the transformation, it seems that the difficult part of the integral can be solved with Cauchy's integral formula.

Once a solution is found for the likelihood function, it is highly recommended to investigate the performance of a generalized likelihood function for more than two cameras. Current similarity measures for n -view vision compute measures for camera pairs and combine the results afterwards. Our method allows incorporation of all information in one model. In other words: the generalized likelihood function then depends on all available cameras directly.

In Chapter 3, it was concluded that the Mahalanobis distance improves the stereo correspondence slightly; however, it was noted that this comes at the cost of significantly more processing power. If we take a closer look at Figure 3.3(a), it appears that a lot of weights in the precision matrix are very close to zero. These weights contribute little to the likelihood. It might be possible that an approximation of the Mahalanobis distance with only a few coefficients performs almost the same as the complete Mahalanobis distance in terms of matching performance. This could reduce the computational complexity of the simplified likelihood function without sacrificing much matching performance.

It was concluded that the number of errors in the matching process decreases with a good Mahalanobis distance instead of a Euclidean distance. The gained correct matches appeared to lie in difficult regions (e.g. near occlusions and discontinuities). It might be interesting to investigate whether these new matches contribute to better performance when post-processing methods are used.

Proofs and Reference

A.1 Integral of a Gaussian function

Theorem

$$\int_{-\infty}^{\infty} a \exp(-bx^2 + cx + f) dx = a \exp\left(\frac{c^2}{4b} + f\right) \sqrt{\frac{\pi}{b}} \quad (\text{A.1})$$

Proof

$$F = \int_{-\infty}^{\infty} a \exp\left(-\frac{(x+b)^2}{c^2}\right) dx \quad (\text{A.2})$$

Factor out a ; change the variable of integration to $y = x + b$:

$$F = a \int_{-\infty}^{\infty} \exp\left(-\frac{y^2}{c^2}\right) dy \quad (\text{A.3})$$

then change the variable of integration to $z = \frac{y}{|c|}$:

$$F = a|c| \int_{-\infty}^{\infty} \exp(-z^2) dz = a|c|\sqrt{\pi} \quad (\text{A.4})$$

However, for ease of use this can be rewritten to an alternative form in which the refactored form is used to separate the constant part of the exponent from the integrand.

Thereafter, the integral can be evaluated as a regular Gaussian integral:

$$\begin{aligned}
& \int_{-\infty}^{\infty} a \exp(-bx^2 + cx + f) dx \\
&= \int_{-\infty}^{\infty} a \exp\left(-b\left(x - \frac{c}{2b}\right)^2 + \frac{c^2}{4b} + f\right) dx \\
&= \int_{-\infty}^{\infty} a \exp\left(-b\left(x - \frac{c}{2b}\right)^2\right) \exp\left(\frac{c^2}{4b} + f\right) dx \\
&= a \exp\left(\frac{c^2}{4b} + f\right) \int_{-\infty}^{\infty} \exp\left(-b\left(x - \frac{c}{2b}\right)^2\right) dx \\
&= a \exp\left(\frac{c^2}{4b} + f\right) \sqrt{\frac{\pi}{b}} \tag{A.5}
\end{aligned}$$

A.2 Multidimensional Gaussian integral

Let $\mathbf{x} = [x_1 \ x_2 \ \dots \ x_n]^T$ and its variables of integration $d^n \mathbf{x} \equiv \prod_{i=1}^n dx_i$.

Theorem

One of the basic properties of the covariance matrix states that the covariance matrix Σ is symmetric positive-semidefinite (See Appendix A.4), thus let Σ be a symmetric positive definite matrix and $f : \mathbb{R}^n \rightarrow \mathbb{R}$, where $f(\mathbf{x}) = \exp(-\frac{1}{2}\mathbf{x}^T \Sigma^{-1} \mathbf{x})$.

Then

$$\int_{\infty}^{\infty} \exp\left(-\frac{1}{2}\mathbf{x}^T \Sigma^{-1} \mathbf{x}\right) d^n \mathbf{x} = \sqrt{(2\pi)^n |\Sigma|}, \tag{A.6}$$

where $|\Sigma| = \det \Sigma$.

Proof

Σ^{-1} is real and symmetric ($(\Sigma^{-1})^T = (\Sigma^T)^{-1} = \Sigma^{-1}$). For convenience, let $\mathbf{A} = \Sigma^{-1}$. We can decompose \mathbf{A} into $\mathbf{A} = \mathbf{T} \mathbf{\Lambda} \mathbf{T}^{-1}$, where \mathbf{T} is an orthonormal ($\mathbf{T}^T \mathbf{T} = \mathbf{I}$) matrix of the eigenvectors of \mathbf{A} and $\mathbf{\Lambda}$ is a diagonal matrix with the eigenvalues of \mathbf{A} . Then

$$\int_{\infty}^{\infty} \exp\left(-\frac{1}{2}\mathbf{x}^T \mathbf{A} \mathbf{x}\right) d^n \mathbf{x} = \int_{\infty}^{\infty} \exp\left(-\frac{1}{2}\mathbf{x}^T \mathbf{T} \mathbf{\Lambda} \mathbf{T}^{-1} \mathbf{x}\right) d^n \mathbf{x}. \tag{A.7}$$

Because \mathbf{T} is orthonormal, we have $\mathbf{T}^{-1} = \mathbf{T}^T$. Now define a new vector variable $\mathbf{y} \equiv \mathbf{T}^T \mathbf{x}$, and substitute:

$$\int_{\infty}^{\infty} \exp\left(-\frac{1}{2}\mathbf{x}^T \mathbf{T} \mathbf{\Lambda} \mathbf{T}^{-1} \mathbf{x}\right) d^n \mathbf{x} = \int_{\infty}^{\infty} \exp\left(-\frac{1}{2}\mathbf{x}^T \mathbf{T} \mathbf{\Lambda} \mathbf{T}^T \mathbf{x}\right) d^n \mathbf{x} \tag{A.8}$$

$$= \int_{\infty}^{\infty} \exp\left(-\frac{1}{2}\mathbf{y}^T \mathbf{\Lambda} \mathbf{y}\right) |\mathbf{J}| d^n \mathbf{y} \tag{A.9}$$

In this equation $|\mathbf{J}|$ is the determinant of the Jacobian matrix $J_{mn} = \frac{\partial x_m}{\partial y_n}$, and in this case, $\mathbf{J} = \mathbf{T}$. The n -by- n -matrix \mathbf{T} is of the special orthogonal group, a subgroup of the general orthogonal group, for which the determinant is always equal to 1. Therefore, the integration by substitution is free as $|\mathbf{J}| = 1$. (The special orthogonal group is the kernel of the Dickson invariant [19].)

Since $\mathbf{\Lambda}$ is diagonal, the integral may be separated into the product of n independent Gaussian distributions. Each independent part can be integrated separately using the well-known formula (equation A.5):

$$\int_{-\infty}^{\infty} \exp\left(-\frac{1}{2}at^2\right) dt = \sqrt{\frac{2\pi}{a}}. \quad (\text{A.10})$$

Solving the integral for every independent part gives:

$$\int_{-\infty}^{\infty} \exp\left(-\frac{1}{2}\mathbf{y}^T \mathbf{\Lambda} \mathbf{y}\right) d^n \mathbf{y} = \prod_{k=1}^n \int_{-\infty}^{\infty} \exp\left(-\frac{1}{2}\lambda_k y_k^2\right) dy_k \quad (\text{A.11})$$

$$= \prod_{k=1}^n \sqrt{\frac{2\pi}{\lambda_k}} \quad (\text{A.12})$$

$$= \sqrt{\frac{(2\pi)^n}{\prod_{k=1}^n \lambda_k}} \quad (\text{A.13})$$

$$= \sqrt{\frac{(2\pi)^n}{|\mathbf{\Lambda}|}}. \quad (\text{A.14})$$

Now, we have $|\mathbf{A}| = |\mathbf{T}\mathbf{\Lambda}\mathbf{T}^{-1}| = |\mathbf{T}||\mathbf{\Lambda}||\mathbf{T}^{-1}| = |\mathbf{\Lambda}|$, so this becomes

$$\int_{-\infty}^{\infty} \exp\left(-\frac{1}{2}\mathbf{x}^T \mathbf{A} \mathbf{x}\right) d^n \mathbf{x} = \sqrt{\frac{(2\pi)^n}{|\mathbf{A}|}}. \quad (\text{A.15})$$

To finalize the proof, \mathbf{A} has to be substituted back in for $\mathbf{\Sigma}^{-1}$ which results in the theorem to be proved:

$$\int_{-\infty}^{\infty} \exp\left(-\frac{1}{2}\mathbf{x}^T \mathbf{\Sigma}^{-1} \mathbf{x}\right) d^n \mathbf{x} = \sqrt{\frac{(2\pi)^n}{|\mathbf{\Sigma}^{-1}|}} = \sqrt{(2\pi)^n |\mathbf{\Sigma}|}, \quad (\text{A.16})$$

A.3 Integral of a multi-dimensional Gaussian function

First, to simplify the integral notation, we introduce:

$$\int_{-\infty}^{\infty} \dots \int_{-\infty}^{\infty} f(\mathbf{x}) dx_1 \dots dx_n = \int_{-\infty}^{\infty} f(\mathbf{x}) d^n \mathbf{x} \quad (\text{A.17})$$

with $f : \mathbb{R}^n \rightarrow \mathbb{R}$, where $f(\mathbf{x}) = a \exp(-b\mathbf{x}^T \mathbf{P} \mathbf{x} + c\mathbf{d}^T \mathbf{x} + f)$.

Theorem

Let \mathbf{C} be a symmetric positive definite matrix and $f : \mathbb{R}^n \rightarrow \mathbb{R}$. Then:

$$a \int_{-\infty}^{\infty} \exp(-b\mathbf{x}^T \mathbf{C}^{-1} \mathbf{x} + c\mathbf{d}^T \mathbf{x} + f) d^n \mathbf{x} = a \exp\left(f + \frac{c^2}{4b} \mathbf{d}^T \mathbf{C} \mathbf{d}\right) \sqrt{|\mathbf{C}|} \left(\frac{\pi}{b}\right)^n, \quad (\text{A.18})$$

where $|\mathbf{C}| = \det \mathbf{C}$.

Proof

The exponent contains a shift vector $\mathbf{d} = [d_1, \dots, d_n]$ and a constant f . As is the case with the basic multivariate integral in Appendix A.2, the matrix \mathbf{P} is constrained to be symmetric and positive semidefinite. These constraints imply that \mathbf{P} is a precision matrix of covariance matrix $\mathbf{C} = \mathbf{P}^{-1}$. (See Appendix A.4.)

This multivariate integral of Equation A.18 is difficult to evaluate; however, it is possible to use the properties of the precision matrix \mathbf{P} to represent this matrix in terms of eigenvalues and eigenvectors that can be integrated separately. With eigendecomposition it is possible to factorize a matrix into a canonical form such that $\mathbf{P} = \mathbf{T} \mathbf{\Lambda} \mathbf{T}^{-1}$. Replacing \mathbf{P} with a diagonal matrix $\mathbf{\Lambda}$ of eigenvalues $\Lambda_{ii} = \lambda_i$ and accompanying matrix \mathbf{T} with corresponding eigenvectors t_i on its columns gives:

$$F = a \int_{-\infty}^{\infty} \exp(-b\mathbf{x}^T \mathbf{T} \mathbf{\Lambda} \mathbf{T}^{-1} \mathbf{x} + c\mathbf{d}^T \mathbf{x} + f) d^n \mathbf{x} \quad (\text{A.19})$$

$$= a \int_{-\infty}^{\infty} d^n \mathbf{x} \exp(-b\mathbf{x}^T \mathbf{T} \mathbf{\Lambda} \mathbf{T}^T \mathbf{x}) \exp(c\mathbf{d}^T \mathbf{x}) \exp(f) \quad (\text{A.20})$$

Because the multivariate integral results in a scalar, it is possible to rewrite the matrix notation to summations as presented in Equation A.54:

$$F = a \int_{-\infty}^{\infty} d^n \mathbf{x} \exp\left(-b \sum_{i=1}^n \sum_{j=1}^n \sum_{k=1}^n x_j t_{ji} \lambda_{ii} t_{ki} x_k\right) \exp\left(c \sum_{l=1}^n d_l x_l\right) \exp(f) \quad (\text{A.21})$$

In the proof of Section A.1, the expression in the exponent is brought into a vertex form to separate the constant part from the integral. The same method can be applied to the current expression while keeping in mind that we would like to bring the eigenvector matrix into the variable of integration. The vertex form becomes:

$$F = a \int_{-\infty}^{\infty} d^n \mathbf{x} \exp\left(-b \sum_{i=1}^n \sum_{j=1}^n \sum_{k=1}^n \lambda_i (t_{ji} x_j - \alpha_{ijk})(t_{ki} x_k - \beta_{ijk}) + \gamma_{ijk}\right) \exp(f). \quad (\text{A.22})$$

In the current state of the expression a term is evaluated n^3 times over all three summations; however, it is possible to rewrite the summations to a vertex form such that

the multiplication of every eigenvector with each input variable is evaluated within the factor. This brings us one step closer to the regular Gaussian integral expressed as:

$$F = a \int_{-\infty}^{\infty} d^n \mathbf{x} \exp \left(-b \sum_{i=1}^n \lambda_i \left(\sum_{j=1}^n t_{ji} x_j - \alpha_i \right)^2 + \gamma_i \right) \exp(f) \quad (\text{A.23})$$

$$= \exp \left(f - b \sum_{i=1}^n \gamma_i \right) \int_{-\infty}^{\infty} d^n \mathbf{x} \exp \left(-b \sum_{i=1}^n \lambda_i \left(\sum_{j=1}^n t_{ji} x_j - \alpha_i \right)^2 \right). \quad (\text{A.24})$$

The constant contributions γ_i and f within the exponent of the integral are extracted and removed from the integrand in equation A.24. The constant summation is temporarily compacted to $g = -b \sum_{i=1}^n \gamma_i$. The quadratic function is factored to bring the expression in vertex or standard form, for which the solution for α_i and γ_i is:

$$\alpha_i = \frac{c}{2b\lambda_i} \mathbf{d}^T \mathbf{T}^{-1} \mathbf{S}^{ii} \quad (\text{A.25})$$

$$\gamma_i = -\lambda_i \alpha_i^2, \quad (\text{A.26})$$

where $\mathbf{S}^{ij} \in \mathbb{R}^{n \times n}$ is the single-entry matrix which is zero everywhere except for the (i, j) -th entry, in which case the value of the entry equals one. The constant part is then grouped in g , with:

$$g = -b \sum_{i=1}^n \gamma_i = b \sum_{i=1}^n \lambda_i \alpha_i^2. \quad (\text{A.27})$$

With substitution of α_i from equation A.25, g can be written back to matrix form, where the eigendecomposition is reversed to regain an expression with the precision matrix \mathbf{P} . The inversion property of the eigenvalue matrix is used in equation A.30 to bring \mathbf{P} to the numerator.

$$g = b \sum_{i=1}^n \lambda_i \left(\frac{c}{2b\lambda_i} \right)^2 \mathbf{d}^T \mathbf{T} (\mathbf{S}^{ii})^2 \mathbf{T}^T \mathbf{d} \quad (\text{A.28})$$

$$= b \sum_{i=1}^n \frac{c^2}{4b^2} \mathbf{d}^T \mathbf{T} \frac{1}{\lambda_i} \mathbf{S}^{ii} \mathbf{T}^T \mathbf{d} \quad (\text{A.29})$$

$$= \frac{c^2}{4b} \mathbf{d}^T \mathbf{T} \mathbf{\Lambda}^{-1} \mathbf{T}^{-1} \mathbf{d} = \frac{c^2}{4b} \mathbf{d}^T \mathbf{P}^{-1} \mathbf{d} = \frac{c^2}{4b} \mathbf{d}^T \mathbf{C} \mathbf{d} \quad (\text{A.30})$$

The next step is reached by substitution of the new integration variable $y_i = \sum_{j=1}^n t_{ji} x_j$. Unfortunately, the substitution introduces a derivative in the expression as described in Appendix A.2 so that the new integral to be evaluated transforms to:

$$F = a \exp(f + g) \int_{-\infty}^{\infty} d^n \mathbf{y} \exp \left(-b \sum_{i=1}^n \lambda_i (y_i - \alpha_i)^2 \right) |\mathbf{J}| \quad (\text{A.31})$$

In equation A.31, $|\mathbf{J}|$ is the absolute determinant of the Jacobian matrix $J_{mn} = \frac{\partial x_m}{\partial y_n}$, and in this case, $\mathbf{J} = \mathbf{T}$ and therefore equal to one (see Appendix A.1 and [19]). Hence, the change of integration variable further simplifies the integral. Because the integration is applied from minus infinity to infinity, the shift variables α_i of the integrand have no contribution to the final result of the integral, and can therefore be omitted:

$$F = a \exp(f + g) \int_{-\infty}^{\infty} d^n \mathbf{y} \exp \left(-b \sum_{i=1}^n \lambda_i (y_i)^2 \right). \quad (\text{A.32})$$

The integral of the Gaussian function is now brought in the same form as the integral in equation A.11. The summation of the multivariate integrand-part inside the exponent is equal to a multiplication of the same number of exponents.

Because the integration of the exponents is independent for every y , the integral of products can also be written as a product of independent integrals:

$$F = a \exp(f + g) \prod_{i=1}^n \int_{-\infty}^{\infty} dy_i \exp(-b\lambda_i y_i^2) \quad (\text{A.33})$$

This product of integrals can be solved with the same recipe as Appendix A.2, which results in:

$$F = a \exp(f + g) \prod_{i=1}^n \sqrt{\frac{\pi}{b\lambda_i}} \quad (\text{A.34})$$

$$= a \exp(f + g) \sqrt{\left(\frac{\pi}{b}\right)^n \frac{1}{\prod_{i=1}^n \lambda_i}} \quad (\text{A.35})$$

$$= a \exp(f + g) \sqrt{\left(\frac{\pi}{b}\right)^n \frac{1}{|\mathbf{\Lambda}|}} \quad (\text{A.36})$$

Again, by substitution of $|\mathbf{\Lambda}|$ with $|\mathbf{P}| = |\mathbf{C}^{-1}| = |\mathbf{C}|^{-1}$ and g with its respective value, we obtain:

$$F = \int_{-\infty}^{\infty} a \exp(-b\mathbf{x}^T \mathbf{P} \mathbf{x} + c\mathbf{d}^T \mathbf{x} + f) d^n \mathbf{x} \quad (\text{A.37})$$

$$= a \exp \left(f + \frac{c^2}{4b} \mathbf{d}^T \mathbf{P}^{-1} \mathbf{d} \right) \sqrt{\left(\frac{\pi}{b}\right)^n \frac{1}{|\mathbf{P}|}} \quad (\text{A.38})$$

$$= a \exp \left(f + \frac{c^2}{4b} \mathbf{d}^T \mathbf{C} \mathbf{d} \right) \sqrt{|\mathbf{C}| \left(\frac{\pi}{b}\right)^n} \quad (\text{A.39})$$

This completes the proof for the improper integral of a multivariate Gaussian function. This result can be used to solve the marginalization of probability density functions that contain an expression with a Gaussian function.

A.4 Proof symmetric positive-semidefinite covariance

The symmetry for the covariance matrix Σ corresponding to a random vector \mathbf{X} follows from its definition:

$$\begin{aligned} \text{cov}(\mathbf{X}, \mathbf{Y}) &= \mathbb{E} \left[(\mathbf{X} - \mathbb{E}[\mathbf{X}]) (\mathbf{Y} - \mathbb{E}[\mathbf{Y}])^T \right] \\ &= \mathbb{E} \left[(\mathbf{Y} - \mathbb{E}[\mathbf{Y}]) (\mathbf{X} - \mathbb{E}[\mathbf{X}])^T \right] \\ &= \text{cov}(\mathbf{Y}, \mathbf{X}) \end{aligned} \quad (\text{A.40})$$

The positive semi-definiteness property of the covariance matrix is obtained by starting with the definition that X_{ik} is the k -th observation of the i -th pixel of m by n length respectively. This implies an observation matrix $\mathbf{X} \in \mathbb{R}^{n \times m}$ where every row has a number of observations for a certain window value. The resulting covariance matrix then is $\Sigma \in \mathbb{R}^{n \times n}$. For any vector $z \in \mathbb{R}^n$:

$$\mathbf{z}^T \Sigma \mathbf{z} = \sum_{i=1}^n \sum_{j=1}^n (z_i \Sigma_{ij} z_j) \quad (\text{A.41})$$

$$= \sum_{i=1}^n \sum_{j=1}^n (\text{cov}(\mathbf{X}_i, \mathbf{X}_j) z_i z_j) \quad (\text{A.42})$$

$$= \sum_{i=1}^n \sum_{j=1}^n (\mathbb{E}[(\mathbf{X}_i - \mathbb{E}[\mathbf{X}_i]) (\mathbf{X}_j - \mathbb{E}[\mathbf{X}_j])]) z_i z_j \quad (\text{A.43})$$

$$= \mathbb{E} \left[\sum_{i=1}^n \sum_{j=1}^n (\mathbf{X}_i - \mathbb{E}[\mathbf{X}_i]) (\mathbf{X}_j - \mathbb{E}[\mathbf{X}_j]) z_i z_j \right] \quad (\text{A.44})$$

$$= \frac{1}{m} \sum_{k=1}^m \sum_{i=1}^n \sum_{j=1}^n (x_{ik} - \mathbb{E}[\mathbf{X}_i]) (x_{jk} - \mathbb{E}[\mathbf{X}_j]) z_i z_j \quad (\text{A.45})$$

$$(\text{A.46})$$

In order to simplify the equation we can rewrite x_{ik} to vectors with zero mean and divide it by the sampling size to include the encapsulating expected value:

$$u_{ik} = \frac{x_{ik} - \mathbb{E}[\mathbf{x}_i]}{m} = \frac{x_{ik} - \sum_{k=1}^m \frac{x_{ik}}{m}}{m} = \frac{x_{ik}}{m} - \sum_{k=1}^m \frac{x_{ik}}{m^2} \quad (\text{A.47})$$

The equation then simplifies to:

$$\mathbf{z}^T \Sigma \mathbf{z} = \sum_{k=1}^m \sum_{i=1}^n \sum_{j=1}^n z_i u_{ik} u_{jk} z_j \quad (\text{A.48})$$

$$= \sum_{k=1}^m \mathbf{z}^T \mathbf{u}_k \mathbf{u}_k^T \mathbf{z} \quad (\text{A.49})$$

$$= \sum_{k=1}^m (\mathbf{u}_k^T \mathbf{z})^T (\mathbf{u}_k^T \mathbf{z}) \quad (\text{A.50})$$

$$= (\mathbf{U}^T \mathbf{z})^T (\mathbf{U}^T \mathbf{z}) \quad (\text{A.51})$$

Let $\mathbf{w} = \mathbf{U}^T \mathbf{z}$:

$$\mathbf{z}^T \Sigma \mathbf{z} = \mathbf{w}^T \mathbf{w} = \sum_{i=1}^n w_i w_i = \sum_{i=1}^n w_i^2 \geq 0 \quad (\text{A.52})$$

Hence, w_i^2 is never negative as long as z and Σ satisfy the condition of being real. Thus results the conclusion that every covariance matrix must be a symmetric positive semidefnite matrix.

A.5 Vector and matrix properties

In this report, several multivariate functions are used with a scalar output: $f : \mathbb{R}^n \rightarrow \mathbb{R}$. To proof some theorems it is practical to switch to a summation notation, where $\mathbf{x} = [x_1 \dots x_n]^T$, $\mathbf{y} = [y_1 \dots y_n]^T$ and $\mathbf{A} \in \mathbb{R}^{n \times n}$:

$$\mathbf{x}^T \mathbf{A} \mathbf{y} = \sum_{i=1}^n \sum_{j=1}^n x_i a_{ij} y_j. \quad (\text{A.53})$$

For a decomposed case, an extra diagonal matrix is introduced with eigenvalues on the diagonal of the matrix and zeros everywhere else:

$$\mathbf{x}^T \mathbf{A} \mathbf{D} \mathbf{B}^T \mathbf{y} = \sum_{i=1}^n \sum_{j=1}^n \sum_{k=1}^n x_j a_{ji} d_i b_{ki} y_k, \quad (\text{A.54})$$

with $\mathbf{B} \in \mathbb{R}^{n \times n}$ and $\mathbf{D} \in \mathbb{R}^{n \times n}$.

Also, the determinant of a the diagonal matrix has a useful alternate notation:

$$\det(\mathbf{D}) = \prod_{i=1}^n d_{ii}. \quad (\text{A.55})$$

A.6 Fubini theorem

Fubini's theorem¹ establishes a connection between a multiple integral and a repeated integral if $f(x, y)$ is measurable on the rectangular region $A \times B$ and if

$$\int_{A \times B} |f(x, y)| \, d(x, y) < \infty, \quad (\text{A.56})$$

then:

$$\int_A \left(\int_B f(x, y) \, dy \right) dx = \int_B \left(\int_A f(x, y) \, dx \right) dy = \int_{A \times B} f(x, y) \, d(x, y). \quad (\text{A.57})$$

Consequently, for $f(x, y) = g(x)h(y)$:

$$\int_A g(x) \, dx \int_B h(y) \, dy = \int_{A \times B} f(x, y) \, d(x, y) \quad (\text{A.58})$$

A.7 Integration by substitution for multiple variables

Sometimes, integrals to may look hard to perform at first hand can be transformed to easier ones through algebraic substitution. The counterpart to the chain rule of differentiation is for a continuous function:

$$\int_a^b f(g(t))g'(t)dt = \int_{g(a)}^{g(b)} f(x) \, dx, \quad (\text{A.59})$$

with the substitution $x = g(t)$. Hence, this yields $\frac{dx}{dt} = g'(t)$, or $dx = g'(t)dt$.

When integrating functions with multiple variables it is possible to use the same method, a change of variables, to simplify the integral. For the substitution function $(v_1, \dots, v_n) = \varphi(u_1, \dots, u_n)$ the integration variables have to be changed to²:

$$dv_1 \cdots dv_n = |\det(D\varphi)(u_1, \dots, u_n)| du_1 \cdots du_n, \quad (\text{A.60})$$

with $\det(D\varphi)(u_1, \dots, u_n)$ holding the partial derivatives of φ . For any real-valued, compactly supported, continuous function f and a real valued set of substitution variables φ , the integral can be rewritten to a new set of integration variables:

$$\int_{\varphi(U)} f(\mathbf{v}) \, d\mathbf{v} = \int_U f(\varphi(\mathbf{u})) |\det(D\varphi)(\mathbf{u})| \, d\mathbf{u}, \quad (\text{A.61})$$

with the absolute value of the determinant as the spanned volume of the given vectors.

¹http://en.wikipedia.org/wiki/Fubini's_theorem

²http://en.wikipedia.org/wiki/Integration_by_substitution

A.8 Cauchy's Residue Theorem

$$\oint_{\Gamma} f(z) dz = 2\pi i \sum_{k=1}^n \text{Res}(f, z_k) \quad (\text{A.62})$$

$$\text{Res}(f, z_0) = \frac{1}{2\pi i} \int_{\gamma} f(z) dz = \lim_{z \rightarrow z_0} \frac{1}{(m-1)!} \frac{d^{m-1}}{dz^{m-1}} [(z - z_0)^m f(z)] \quad (\text{A.63})$$

Cauchy's principal value:

$$\text{p.v.} \int_{-\infty}^{\infty} f(x) dx := \lim_{\rho \rightarrow \infty} \int_{-\rho}^{\rho} f(x) dx \quad (\text{A.64})$$

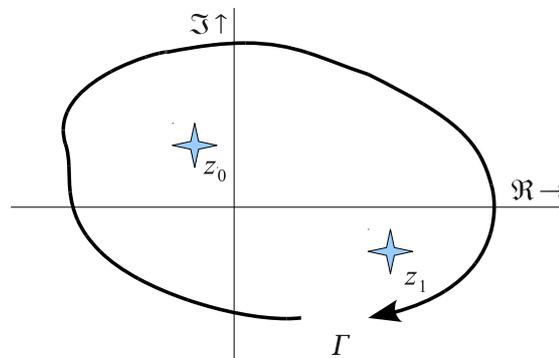


Figure A.1: Contour integral around poles z_i

B

List of project files

B.1 Data and file structure

```
/
├── matlab/ ..... Matlab workspace, see Table B.1
├── data/
│   ├── delta/ ..... Window size benchmarks
│   ├── energy/ ..... Benchmarks for low energy regions
│   ├── full/ ..... Leave-one-out benchmarks (high resolution)
│   ├── full_self/ ..... Self-trained benchmarks (high resolution)
│   ├── scaled2/ ..... Leave-one-out benchmarks (half resolution)
│   │   ├── self/ ..... Self-trained benchmarks
│   │   ├── occmed/ ..... Benchmarks trained and applied on occluded regions
│   │   └── occmed_self/ ..... Self-trained benchmarks on occluded regions
│   ├── scaled4/ ..... Leave-one-out benchmarks (one-fourth resolution)
│   │   ├── self/ ..... Self-trained benchmarks
│   │   ├── occmed/ ..... Benchmarks trained and applied on occluded regions
│   │   └── occmed_self/ ..... Self-trained benchmarks on occluded regions
├── results/
├── images/
│   ├── Aloe/ ..... Figure C.1(a)
│   ├── Baby3/ ..... Figure C.1(c)
│   ├── Bowling2/ ..... Figure C.1(e)
│   ├── Flowerpots/ ..... Figure C.1(g)
│   ├── Midd1/ ..... Figure C.1(i)
│   ├── Monopoly/ ..... Figure C.1(k)
│   ├── Rocks1/ ..... Figure C.1(m)
│   └── Wood1/ ..... Figure C.1(o)
```

B.2 Scripts and programs

Script	Description
funcgui.m	Visual exploration of the likelihood function
cell_filter.m	Select a subset from the dataset (leave-one-out)
collect_cov.m	Collect covariance matrices required for a sweep
compute_cov.m	Compute a covariance matrix for applied datasets
compute_cov_occ.m	Compute a covariance matrix for occluded parts only
compute_energy.m	Compute the energy of windows for stereo images
covmat_reg_test.m	Explore the effects of regularization of covariance matrices
dataset_info.m	Generate information descriptor for the results datasets
diff_mask.m	Visualize image regions based disparity errors
disparity_diff.m	Compute difference between ground-truth and results
row_fw.m	Optimal path for a disparity row with the forward algorithm
row_fwbw.m	Optimal path for a disparity row with forward-backward
row_viterbi.m	Viterbi optimal path for a disparity row
gen_covmat.m	Generate all required covariance matrices for the benchmarks
gen_covmat_occ.m	Generate covariance matrices for occluded regions
genreport_covplot.m	Function to plot the 2D-eigenvectors of a covariance matrix
genreport_imerrors.m	Visualization of the gains and losses between two methods
genres_eigenvalues.m	Extract the eigenvalues and eigenvectors from results datasets
gt_occlusion.m	Detect occlusions between the left and the right ground-truth
occlusion_mask.m	Generate a mask based on occlusion detection
occlusion_range.m	Generates map for how occluded every window is
scale_disparity.m	Scale a disparity map or ground-truth for downsampled benchmarks
stereo_compare.m	Computation of the actual results (calls measure, returns disparity)
stereo_image_slice.m	Extract a selection from an image or ground-truth
stereo_row_ncc.m	Compute the normalized cross correlation for an image row
stereo_row_ssd.m	Compute the similarity measure with the Mahalanobis distance
stereo_row_ssdu.m	Compute the similarity measure with the Euclidean distance
sweep_const.m	Initialization parameters for the sweeps (names, paths, and config)
sweep.m	Generate the primary result files (distributed computing)
sweep_cov.m	Generate results for variations of the covariance matrix
sweep_delta.m	Generate results for a changing window size
sweep_eye.m	Generate results for the Euclidean distance
sweep_occ.m	Mahalanobis distance results trained on occlusions
sweep_ssd_noise.m	Measure the effect of different noises (Euclidean)
sweep_<TYPE>_collect.m	Collect the partial results and combine in a single dataset

Table B.1: Matlab scripts used to run the experiments

B.3 Image dataset

The image dataset structure:

/data/images/<DATASET>/

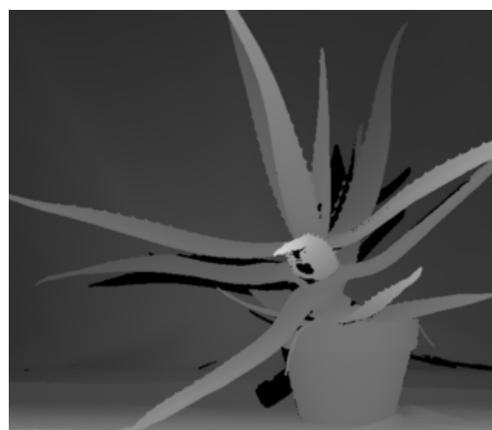
```
|_ disp1.png.....Disparity grond-truth for the left camera
|_ disp5.png.....Disparity grond-truth for the right camera
  |_ Illum{1,2,3}/... 1: normal illumination
    | 2: alternative illumination
    | 3: alternative illumination
    |_ Exp{0,1,2}/ ... 0: very short exposure time → dark image
      | 1: short exposure time
      | 2: optimal exposure time for illumination 1
      |_ view1.png..... Image taken with left camera
      |_ view5.png..... Image taken with right camera
      |_ view{0,2,3,4,6,7}.png ..... Images > 2-view vision
```


C

Dataset



(a) Aloe left view



(b) Aloe disparity



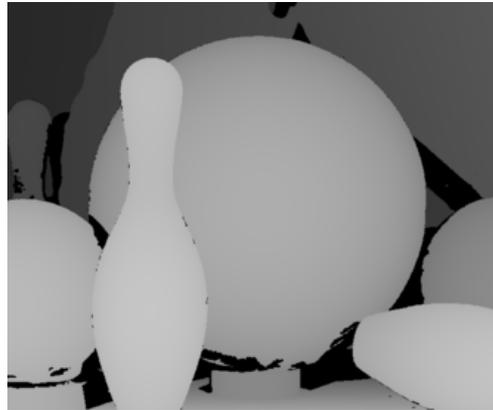
(c) Baby3 left view



(d) Baby3 disparity



(e) Bowling2 left view



(f) Bowling2 disparity



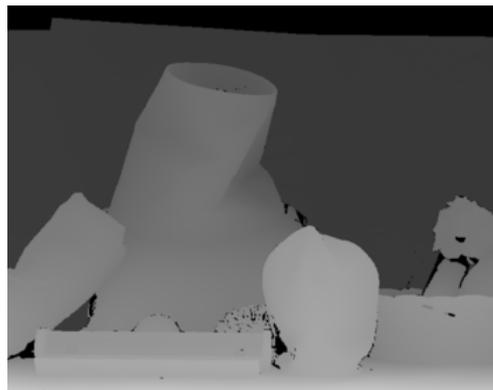
(g) Flowerpots left view



(h) Flowerpots disparity



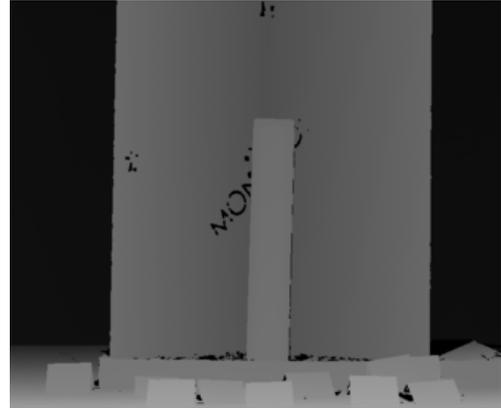
(i) Midd1 left view



(j) Midd1 disparity



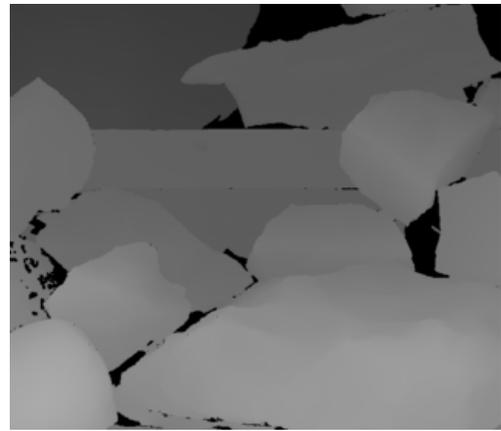
(k) Monopoly left view



(l) Monopoly disparity



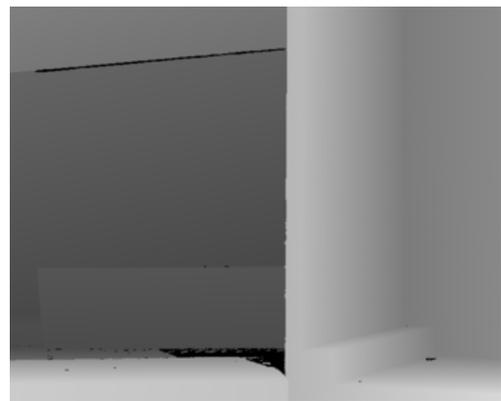
(m) Rocks1 left view



(n) Rocks1 disparity



(o) Wood1 left view



(p) Wood1 disparity

Figure C.1: Images selected from the Middlebury dataset

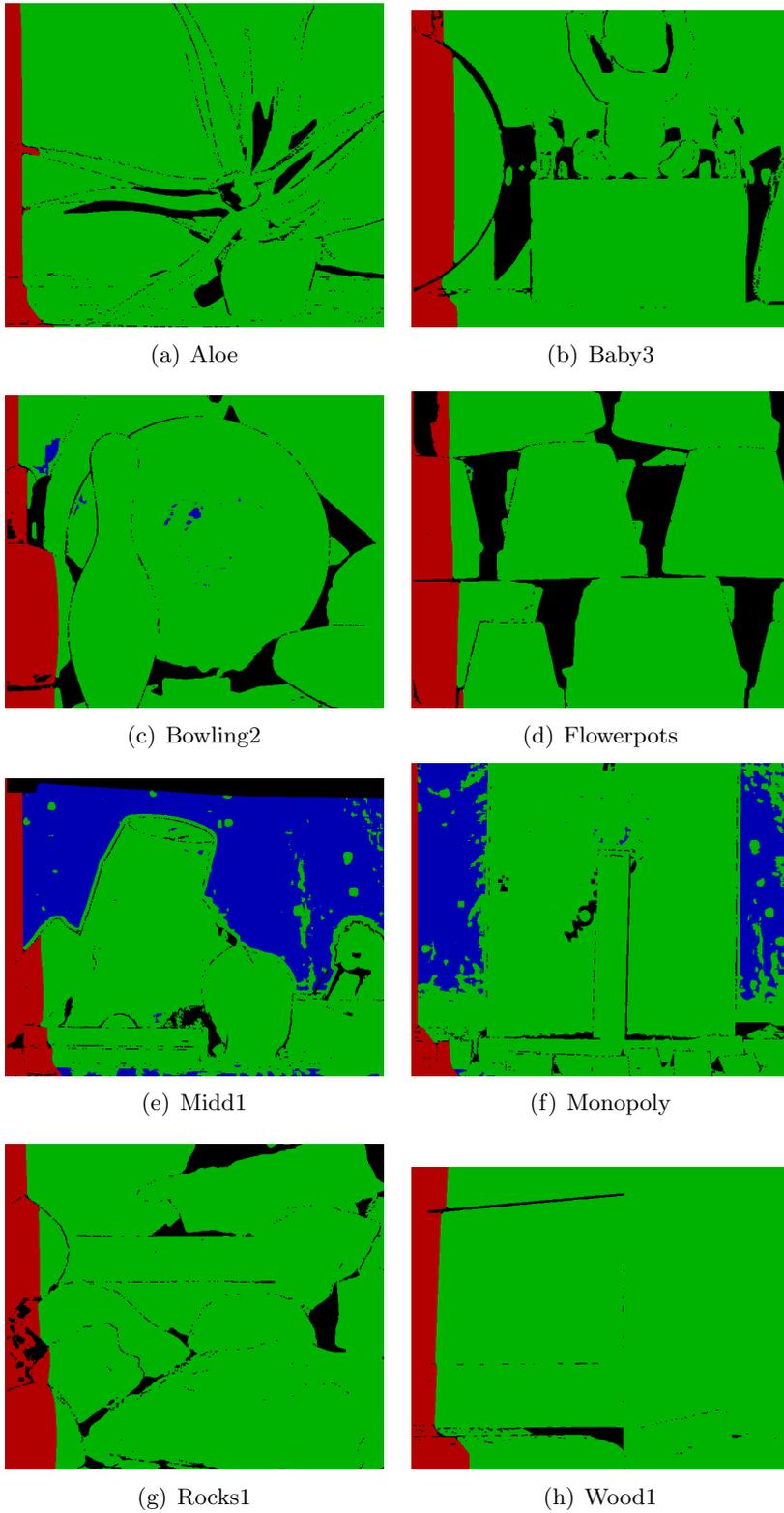


Figure C.2: Overview of the image regions of the selected datasets. Green: ground-truth is known. Red: impossible to estimate disparity due to the offset in camera location. Black: ground-truth is undefined. Blue: region with very low energy, and therefore difficult to match between images.

D

Supplemental results

Reg.	Bowling2	Baby3	Midd1	Monopoly	Aloe	Wood1	Flowerpots	Rocks1
10^{-6}	57314	29713	189914	184095	61203	43633	49071	18079
$10^{-5.5}$	57300	29694	189823	184007	61192	43651	49027	18066
10^{-5}	57271	29681	189697	183874	61185	43674	48979	18048
$10^{-4.5}$	57161	29643	189350	183417	61129	43682	48831	18018
10^{-4}	56876	29605	188817	182725	61085	43668	48623	17980
$10^{-3.5}$	56408	29568	188015	181602	61054	43617	48310	17957
10^{-3}	55848	29549	187107	180318	61027	43565	47970	17924
$10^{-2.5}$	55281	29562	186424	179185	61088	43515	47723	17945
10^{-2}	54987	29705	185975	178280	61325	43501	47545	18054
$10^{-1.5}$	54878	29988	186117	178084	61679	43544	47599	18226
10^{-1}	55099	30377	186973	178794	62127	43660	47934	18506
$10^{-0.5}$	55466	30795	188610	180487	62669	43762	48590	18822
10^0	55847	31127	190121	182162	63206	43837	49204	19052
$10^{0.5}$	56066	31319	191302	183491	63568	43931	49684	19173
10^1	56149	31404	191838	184082	63745	43974	49906	19229
$10^{1.5}$	56200	31441	192048	184318	63809	43993	49993	19254
10^2	56216	31453	192118	184393	63827	43997	50020	19259
$10^{2.5}$	56220	31455	192156	184432	63830	43997	50035	19259
10^3	56220	31458	192165	184443	63835	43998	50039	19260
Samples	299698	281712	320494	337470	306092	332762	259313	289511

Table D.1: Incorrect pixel disparities

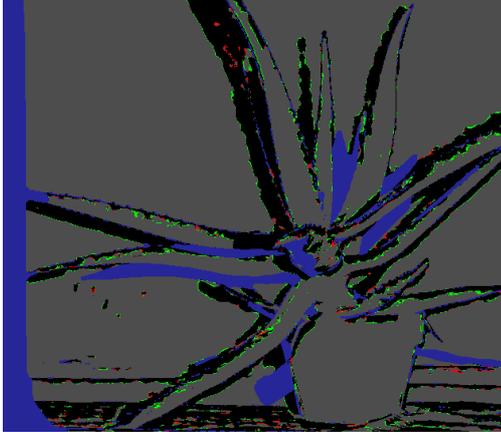
Reg.	Bowling2	Baby3	Midd1	Monopoly	Aloe	Wood1	Flowerpots	Rocks1
10^{-6}	56553	29713	49793	104729	61203	43633	49044	18079
$10^{-5.5}$	56540	29694	49780	104667	61192	43651	49000	18066
10^{-5}	56511	29681	49759	104568	61185	43674	48952	18048
$10^{-4.5}$	56401	29643	49682	104248	61129	43682	48804	18018
10^{-4}	56113	29605	49550	103774	61085	43668	48596	17980
$10^{-3.5}$	55644	29568	49349	103043	61054	43617	48283	17957
10^{-3}	55092	29549	49122	102170	61027	43565	47943	17924
$10^{-2.5}$	54524	29562	48958	101398	61088	43515	47696	17945
10^{-2}	54228	29705	48897	100767	61325	43501	47518	18054
$10^{-1.5}$	54116	29988	48988	100602	61679	43544	47573	18226
10^{-1}	54338	30377	49296	101003	62127	43660	47908	18506
$10^{-0.5}$	54707	30795	49775	101991	62669	43762	48564	18822
10^0	55088	31127	50221	103017	63206	43837	49177	19052
$10^{0.5}$	55305	31319	50562	103818	63568	43931	49658	19173
10^1	55388	31404	50710	104173	63745	43974	49880	19229
$10^{1.5}$	55439	31441	50771	104317	63809	43993	49967	19254
10^2	55455	31453	50789	104362	63827	43997	49994	19259
$10^{2.5}$	55458	31455	50800	104385	63830	43997	50009	19259
10^3	55458	31458	50803	104392	63835	43998	50013	19260
Samples	297987	281712	193098	271628	306092	332762	259257	289511

Table D.2: Incorrect pixel disparities with low energy windows discarded

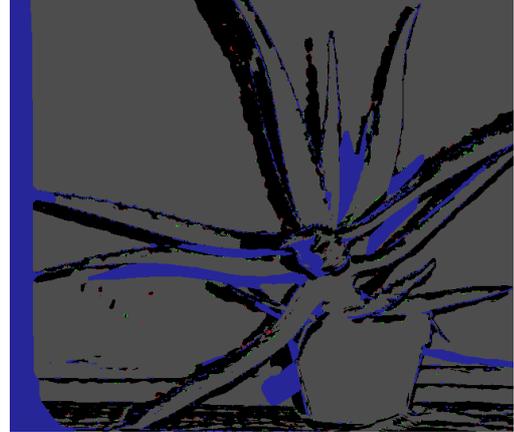
Reg.	Bowling2	Baby3	Midd1	Monopoly	Aloe	Wood1	Flowerpots	Rocks1
10^{-6}	57320	29716	190558	184803	61210	43633	49067	18076
$10^{-5.5}$	57304	29708	190418	184664	61186	43640	49006	18060
10^{-5}	57285	29685	190150	184384	61188	43662	48899	18037
$10^{-4.5}$	57183	29662	189576	183720	61170	43682	48658	18011
10^{-4}	56898	29620	188708	182639	61131	43686	48295	17968
$10^{-3.5}$	56545	29593	187485	181162	61087	43668	47820	17951
10^{-3}	56032	29486	186571	179855	61046	43608	47486	17918
$10^{-2.5}$	55420	29448	186009	178753	61026	43518	47283	17960
10^{-2}	55018	29571	185852	178142	61084	43469	47236	18020
$10^{-1.5}$	54832	29858	186127	178075	61334	43480	47346	18145
10^{-1}	54933	30300	187007	178788	61794	43567	47693	18399
$10^{-0.5}$	55275	30756	188713	180478	62345	43656	48376	18728
10^0	55724	31104	190623	182592	63018	43787	49153	18991
$10^{0.5}$	56014	31316	191922	184084	63463	43887	49677	19159
10^1	56132	31404	192515	184788	63699	43957	49913	19218
$10^{1.5}$	56190	31443	192723	185029	63787	43989	49996	19245
10^2	56212	31451	192794	185109	63822	43996	50022	19258
$10^{2.5}$	56217	31454	192827	185146	63828	43998	50035	19259
10^3	56219	31458	192836	185156	63832	43997	50039	19260
Samples	299698	281712	320494	337470	306092	332762	259313	289511

Table D.3: Incorrect pixel disparities for images processed with covariances matrices trained on the dataset itself

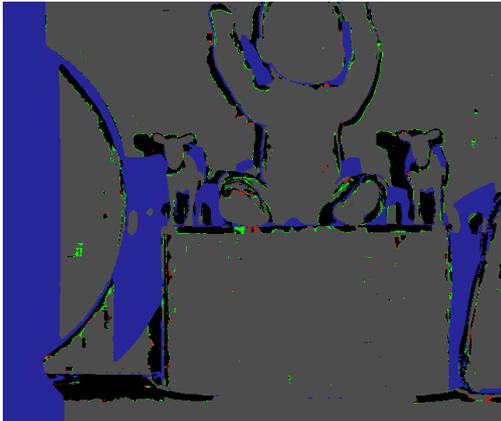
D.1 Performance differences



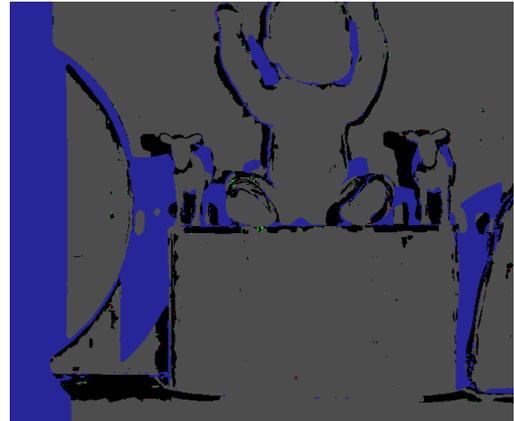
(a) Aloe: Mahalanobis vs. Euclidean



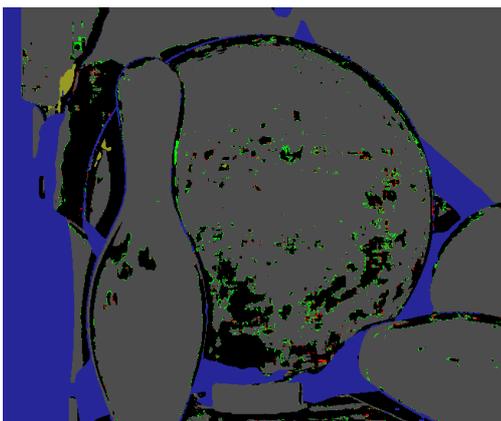
(b) Aloe: trained on self vs. general



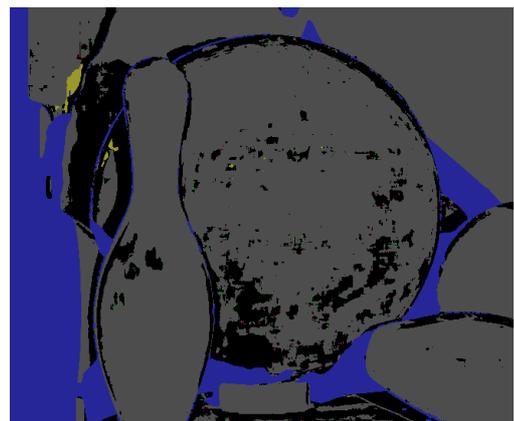
(c) Baby3: Mahalanobis vs. Euclidean



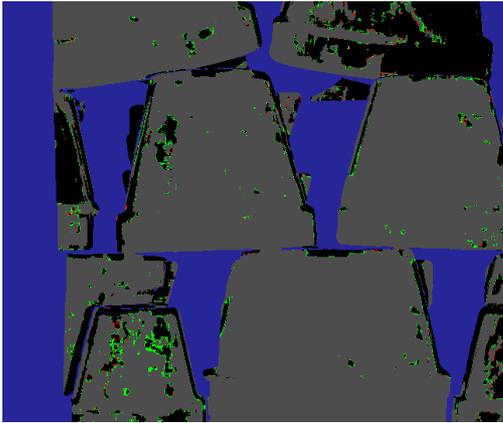
(d) Baby3: trained on self vs. general



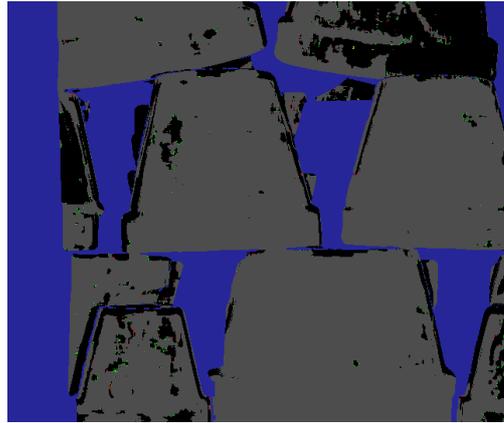
(e) Bowling2: Mahalanobis vs. Euclidean



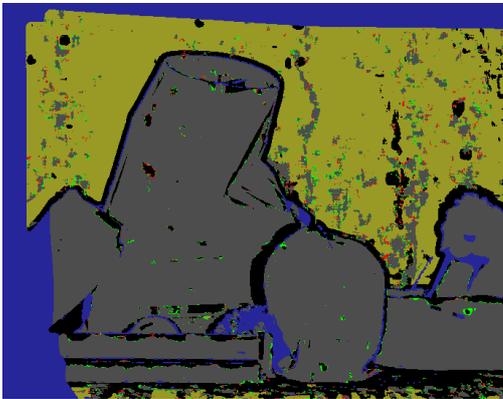
(f) Bowling2: trained on self vs. general



(g) Flowerpots: Mahalanobis vs. Euclidean



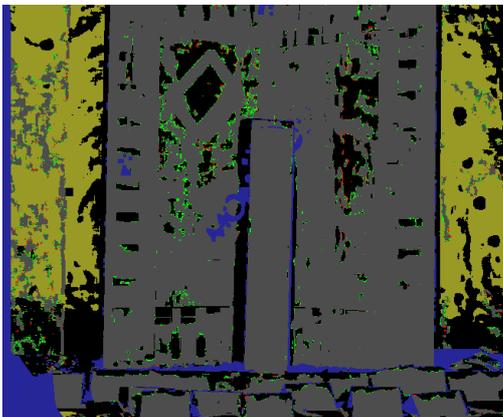
(h) Flowerpots: trained on self vs. general



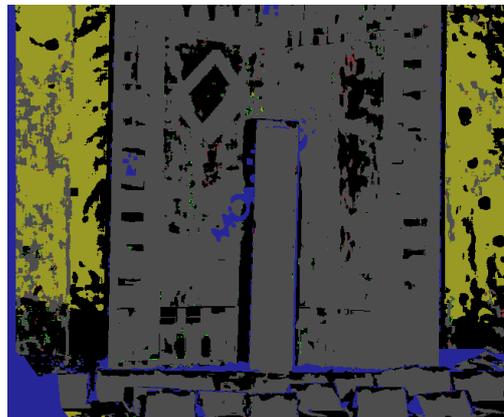
(i) Midd1: Mahalanobis vs. Euclidean



(j) Midd1: trained on self vs. general



(k) Monopoly: Mahalanobis vs. Euclidean



(l) Monopoly: trained on self vs. general

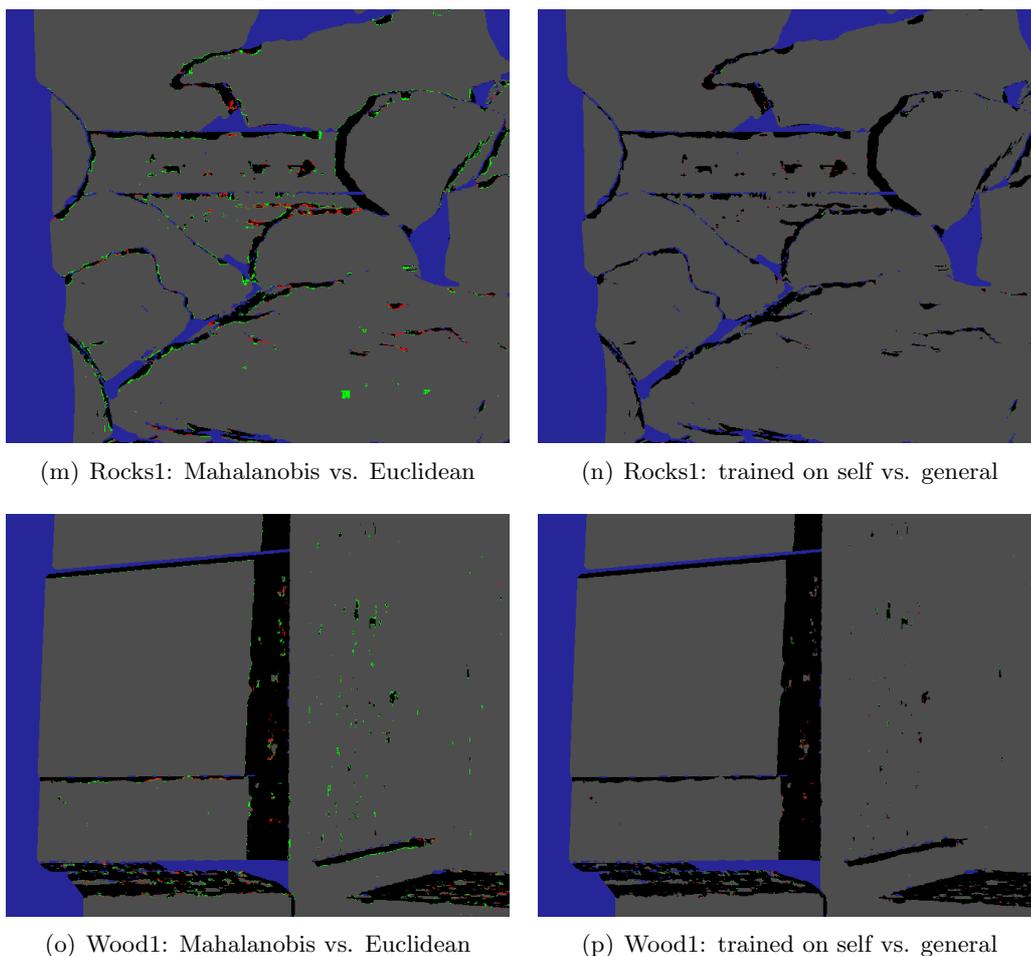


Figure D.1: On the left side: the difference in performance between the Mahalanobis method versus the Euclidean method. On the right side: the difference between covariance matrices trained with the leave-one-out method compared to training on the dataset itself. The green pixels depict the gain of correct pixels, whereas the red pixels show the loss of correct results. The netto gain in performance is the difference of the gain and loss of correct disparities. Furthermore, the gray pixels denote common successes for both methods. The black pixels represent regions for which depth cannot be extracted or compared to the reference images. Low energy regions are highlighted with yellow.

D.2 Eigenvectors

The images contain all eigenvectors of the covariance matrix, with every block as an individual eigenvector transformed back to two dimensions. The dominant eigenvector is placed in the top-left corner, and the accompanying eigenvalue decreases column-wise to the bottom and to the right. The smallest eigenvalue of the covariance matrix belongs

to the top-left eigenvector. Because the precision matrix is the inverse of the covariance matrix, the top-left block is the most important and the bottom-right block is the least important for the similarity measure.

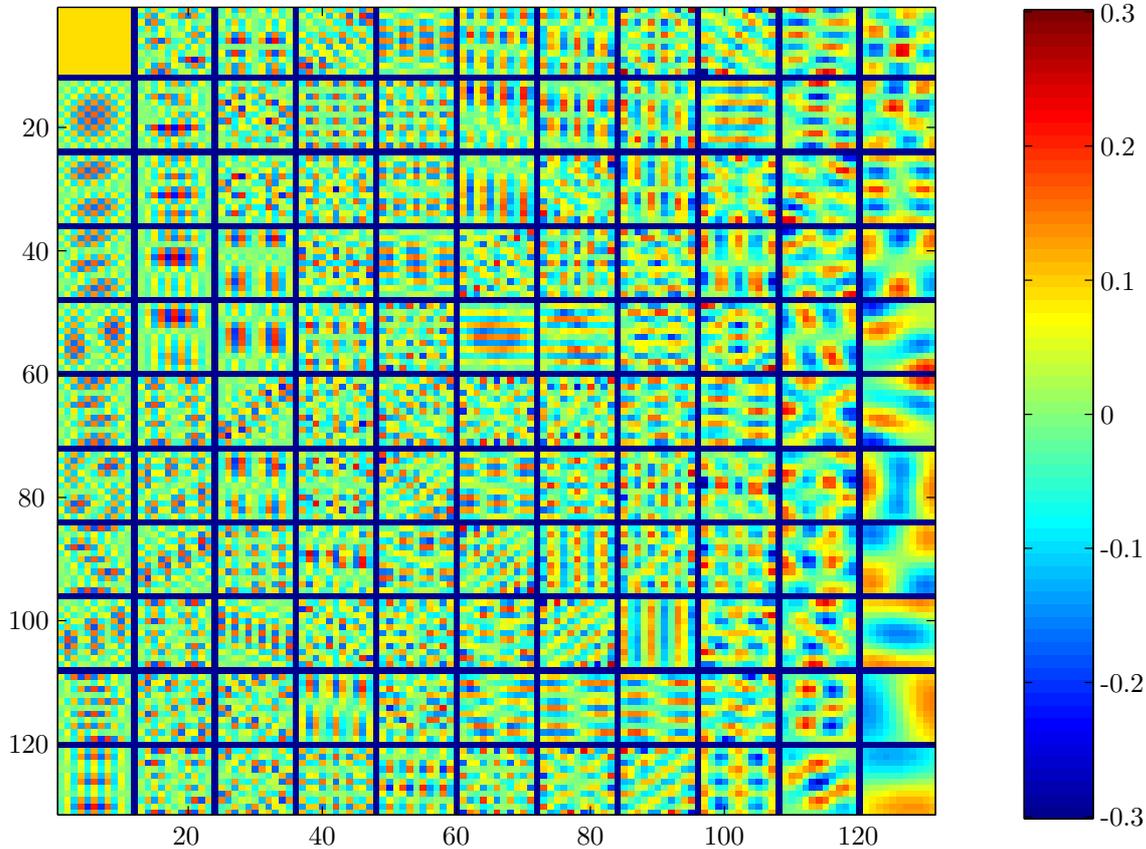


Figure D.2: Eigenvectors of the covariance matrix for the Aloe image; gray; window size = 11×11

The eigenvectors for the covariance matrix generated with the Aloe image dataset is shown in Figure D.2. It was generated for grayscale high resolution images and a window size of 11 by 11. It appears that the distribution of the eigenvectors is similar to Figure 3.5. The constant eigenvector has the largest contribution to the precision matrix. Then, the high-frequency eigenvectors contribute the most to the precision matrix weights. Also, the (large) eigenvectors show a clear drop-off in importance from the center of the window to the outside of the window. Hence, it takes the shape of a two-dimensional Gaussian. The tilted planes and low-order eigenvectors (on the right side of the image) contribute most to the covariance matrix, and therefore implies that these are the least important eigenvectors for the precision matrix.

The covariance matrix used to test the performance of the Mahalanobis distance is generated from color images. The dimensions of the covariance and precision matrix

are therefore a factor three larger, and every eigenvector consists consequently of three primary color blocks. The eigenvectors for a covariance matrix generated from the color version of the **Bowling2** dataset is given in Figure D.3. The importance of the eigenvectors is sorted in the same way as Figure D.2. However, the introduction of separate primary color values increases the number of eigenvectors. The shape and order of high and low frequency is very similar to the gray scale versions.



Figure D.3: Eigenvectors of the covariance matrix for the **Bowling2** image; color; window size = 11×11

Taylor expansions and marginalizations in Maple

```

1 > p := (a1, a2) -> K/(a1^2+a2^2)^(n/2)
2   * exp(-1/2/(a1^2+a2^2) * (a2^2*r11+a1^2*r22-2*a1*a2*r12))
3   * exp(-(a1-1)^2/s1^2)
4   * exp(-(a2-1)^2/s2^2);
5 > # expand about (1,1)
6 > a1_0 := 1; a2_0 := 1;
7 > tmp:= (x^2+y^2)^(-(1/2)*n);
8
9 > p_taylor[0] := simplify(p(a1_0, a2_0));
10 > p_taylor[1] := collect(eval((
11   simplify(diff(p(x, y), x)/tmp) * (a1 - a1_0)
12   + simplify(diff(p(x, y), y)/tmp) * (a2 - a2_0)
13 ))*tmp, [x=a1_0, y=a2_0]), [exp, K]);
14 > p_taylor[2] := collect(eval( 1/2!*(
15   simplify(diff(p(x, y), x$2)/tmp) * (a1 - a1_0)^2
16   + 2*simplify(diff(p(x, y), x, y)/tmp) * (a1 - a1_0) * (a2 - a2_0)
17   + simplify(diff(p(x, y), y$2)/tmp) * (a2 - a2_0)^2
18 ))*tmp, [x=a1_0, y=a2_0]), [exp, K]);
19 > p_taylor[3] := collect(eval( 1/3!*(
20   simplify(diff(p(x, y), x$3)/tmp) * (a1 - a1_0)^3
21   + 3*simplify(diff(p(x, y), x$2, y$1)/tmp) * (a1 - a1_0)^2 * (a2 - a2_0)
22   + 3*simplify(diff(p(x, y), x$1, y$2)/tmp) * (a1 - a1_0) * (a2 - a2_0)^2
23   + simplify(diff(p(x, y), y$3)/tmp) * (a2 - a2_0)^3
24 ))*tmp, [x=a1_0, y=a2_0]), [exp, K]);
25 > p_taylor[4] := collect(eval( 1/4!*(
26   simplify(diff(p(x, y), x$4)/tmp) * (a1 - a1_0)^4
27   + 4*simplify(diff(p(x, y), x$3, y$1)/tmp) * (a1 - a1_0)^3 * (a2 - a2_0)
28   + 6*simplify(diff(p(x, y), x$2, y$2)/tmp) * (a1 - a1_0)^2 * (a2 - a2_0)^2
29   + 4*simplify(diff(p(x, y), x$1, y$3)/tmp) * (a1 - a1_0) * (a2 - a2_0)^3
30   + simplify(diff(p(x, y), y$4)/tmp) * (a2 - a2_0)^4
31 ))*tmp, [x=a1_0, y=a2_0]), [exp, K]);
32 > p_taylor[5] := collect(eval( 1/5!*(
33   simplify(diff(p(x, y), x$5)/tmp) * (a1 - a1_0)^5
34   + 5*simplify(diff(p(x, y), x$4, y$1)/tmp) * (a1 - a1_0)^4 * (a2 - a2_0)
35   + 10*simplify(diff(p(x, y), x$3, y$2)/tmp) * (a1 - a1_0)^3 * (a2 - a2_0)^2
36   + 10*simplify(diff(p(x, y), x$2, y$3)/tmp) * (a1 - a1_0)^2 * (a2 - a2_0)^3
37   + 5*simplify(diff(p(x, y), x$1, y$4)/tmp) * (a1 - a1_0) * (a2 - a2_0)^4
38   + simplify(diff(p(x, y), y$5)/tmp) * (a2 - a2_0)^5
39 ))*tmp, [x=a1_0, y=a2_0]), [exp, K]);

```

E.1 Third order approximation

```

1 > pt3:= unapply(
2 >   simplify(collect(
3 >     int(int(p_taylor[0]+p_taylor[1]+p_taylor[2], a1=a1b..a1e), a2=a2b..a2e),
4 >     [exp, K]
5 >   )/2^(-(1/2)*n)), r11, r12, r22, n, s1, s2, a1b, a1e, a2b, a2e);
6 > pt3s:= simplify(pt3(r11, r12, r22, n, s, t, 1-a, 1+a, 1-b, 1+b));

```

E.2 Three cameras

```

1 >p3:= (a1, a2, a3) -> K / (a1^2+a2^2+a3^2)^(n/2) * exp((
2   - r11 * (a2^2+a3^2)
3   + r12 * (2*a1*a2)
4   + r13 * (2*a1*a3)
5   - r22 * (a1^2+a3^2)
6   + r23 * (2*a2*a3)
7   - r33 * (a1^2+a2^2)
8   ) / (2*(a1^2+a2^2+a3^2)))
9 * exp(-(a1-1)^2/s1^2)
10 * exp(-(a2-1)^2/s2^2)
11 * exp(-(a3-1)^2/s3^2);
12
13 > a1_0 := 1; a2_0 := 1; a3_0 := 1;
14 > tmp:= (x^2+y^2+z^2)^(-(1/2)*n);
15 > s:= [x=a1_0, y=a2_0, z=a3_0];
16
17 > p3_taylor[0] := simplify(p3(a1_0, a2_0, a3_0));
18 > p3_taylor[1] := collect(eval((
19   simplify(eval(diff(p3(x, y, z), x)/tmp, s)) * (a1 - a1_0)
20   + simplify(eval(diff(p3(x, y, z), y)/tmp, s)) * (a2 - a2_0)
21   + simplify(eval(diff(p3(x, y, z), z)/tmp, s)) * (a3 - a3_0)
22   )*tmp, s), [exp, K]) / 1!;
23 > p3_taylor[2] := collect(eval((
24   simplify(eval(diff(p3(x, y, z), x$2)/tmp, s)) * (a1 - a1_0)^2
25   + simplify(eval(diff(p3(x, y, z), y$2)/tmp, s)) * (a2 - a2_0)^2
26   + simplify(eval(diff(p3(x, y, z), z$2)/tmp, s)) * (a3 - a3_0)^2
27   +2*simplify(eval(diff(p3(x, y, z), x,y)/tmp, s)) * (a1 - a1_0) * (a2 - a2_0)
28   +2*simplify(eval(diff(p3(x, y, z), x,z)/tmp, s)) * (a1 - a1_0) * (a3 - a3_0)
29   +2*simplify(eval(diff(p3(x, y, z), y,z)/tmp, s)) * (a2 - a2_0) * (a3 - a3_0)
30   )*tmp, s), [exp, K]) / 2!;
31 > p3_taylor[3] := collect(eval((
32   simplify(eval(diff(p3(x, y, z), x$3)/tmp, s)) * (a1 - a1_0)^3
33   + simplify(eval(diff(p3(x, y, z), y$3)/tmp, s)) * (a2 - a2_0)^3
34   + simplify(eval(diff(p3(x, y, z), z$3)/tmp, s)) * (a3 - a3_0)^3
35   +3!/2!/1!/0!*simplify(eval(diff(p3(x, y, z), x$2,y)/tmp, s)) * (a1 - a1_0)^2 * (a2 - a2_0)
36   +3!/2!/1!/0!*simplify(eval(diff(p3(x, y, z), x$2,z)/tmp, s)) * (a1 - a1_0)^2 * (a3 - a3_0)
37   +3!/2!/1!/0!*simplify(eval(diff(p3(x, y, z), y$2,x)/tmp, s)) * (a2 - a2_0)^2 * (a1 - a1_0)
38   +3!/2!/1!/0!*simplify(eval(diff(p3(x, y, z), y$2,z)/tmp, s)) * (a2 - a2_0)^2 * (a3 - a3_0)
39   +3!/2!/1!/0!*simplify(eval(diff(p3(x, y, z), z$2,x)/tmp, s)) * (a3 - a3_0)^2 * (a1 - a1_0)
40   +3!/2!/1!/0!*simplify(eval(diff(p3(x, y, z), z$2,y)/tmp, s)) * (a3 - a3_0)^2 * (a2 - a2_0)
41   +3!/1!/1!/1!*simplify(eval(diff(p3(x, y, z), x,y,z)/tmp, s)) * (a1 - a1_0) * (a2 - a2_0) * (
42   a3_0 - a3_0)
43   )*tmp, s), [exp, K]) / 3!;

```

Marginalizations:

```

1 > p3t3:= unapply(
2   simplify(collect(
3     int(int(int(p3_taylor[0]+p3_taylor[1]+p3_taylor[2], a1=a1b..a1e), a2=a2b..a2e), a3=a3b..a3e
4     ),
5     [exp, K]
6   )/3^(-(1/2)*n)), r11, r12, r13, r22, r23, r33, n, s1, s2, s3, a1b, a1e, a2b, a2e, a3b, a3e);
> p3t3s:= simplify(p3t3(r11, r12, r13, r22, r23, r33, n, s, t, u, 1-a, 1+a, 1-b, 1+b, 1-c, 1+c)
);

```

Fifth order coefficient extractor

```

1 > extract_coeff35 := proc(f, i11, i22, i33, i12, i13, i23)
2   local res, tmp, i, j, k;
3   # extract coefficient
4   res:= coeff(f * r11, r11^(i11+1));
5   res:= coeff(res * r22, r22^(i22+1));
6   res:= coeff(res * r33, r33^(i33+1));
7   res:= coeff(res * r12, r12^(i12+1));
8   res:= coeff(res * r13, r13^(i13+1));
9   res:= coeff(res * r23, r23^(i23+1));
10
11   # collect s, t, u and n
12   tmp:= res;
13   res:= 0;
14   for i from 0 by 2 to 4 do
15     for j from 0 by 2 to 4 do
16       for k from 0 by 2 to 4 do
17         res:= res + collect(simplify(coeff(coeff(coeff(tmp/s^i/t^j/u^k, 1/s^4), 1/t^4), 1/u
18           ^4) / s^(4-i) / t^(4-j) / u^(4-k)), n);
19       end do;
16     end do;
17   end do;

```

```
20     end do;
21
22     # return result
23     res;
24 end proc;
25
26 > max_order:= 4;
27 > cnt:= 0;
28 > for i11 from 0 to max_order do
29 >   for i22 from 0 to max_order do
30 >     for i33 from 0 to max_order do
31 >       for i12 from 0 to max_order do
32 >         for i13 from 0 to max_order do
33 >           for i23 from 0 to max_order do
34 >             if (i11+i22+i33+i12+i13+i23<=max_order) then
35 >               c35[i11,i22,i33,i12,i13,i23]:= extract_coeff35(p3t5_poly, i11, i22, i33, i12,
36 >                 i13, i23);
37 >               cnt:= cnt + 1;
38 >             end if;
39 >           end do;
40 >         end do;
41 >       end do;
42 >     end do;
43 >   end do;
44 >   printf("Number of coefficients found: %d", cnt);
```

Bibliography

- [1] P.N. Belhumeur. A bayesian approach to binocular stereopsis. *International Journal of Computer Vision*, 19(3):237–260, 1996.
- [2] I. J. Cox. A maximum likelihood N-camera stereo algorithm. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pages 733–739, Los Alamitos, CA, USA, June 1994. IEEE Computer Society Press.
- [3] I.J. Cox, S.L. Hingorani, S.B. Rao, and B.M. Maggs. A maximum likelihood stereo algorithm. *Computer vision and image understanding*, 63(3):542–567, 1996.
- [4] S. Damjanović, F. van der Heijden, and L.J. Spreeuwers. A new likelihood function for stereo matching: how to achieve invariance to unknown texture, gains and offsets? University of Twente, 2009.
- [5] Olivier Faugeras. *Three-Dimensional Computer Vision: A Geometric Viewpoint*. MIT Press, Cambridge, Massachusetts, 1993.
- [6] M.A. Fischler and R.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.
- [7] M. J. Hannah. *Computer matching of areas in stereo images*. PhD thesis, Comput. Sci. Dept., Stanford Univ, Stanford, CA, July 1974. also Tech. Rep. STAN-CS-74-438.
- [8] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition, 2004.
- [9] H. Hirschmüller. Accurate and efficient stereo processing by semi-global matching and mutual information. In *CVPR*, pages II: 807–814, 2005.
- [10] H. Hirschmüller, P.R. Innocent, and J. Garibaldi. Real-time correlation-based stereo vision with reduced border errors. *International Journal of Computer Vision*, 47(1):229–246, 2002.

-
- [11] H. Hirschmüller and D. Scharstein. Middlebury stereo datasets. <http://vision.middlebury.edu/stereo/data/>, 2006.
 - [12] H. Hirschmüller and D. Scharstein. Evaluation of cost functions for stereo matching. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2007*, June 2007.
 - [13] H. Hirschmüller and D. Scharstein. Evaluation of stereo matching costs on images with radiometric differences. *IEEE transactions on pattern analysis and machine intelligence*, pages 1582–1599, 2008.
 - [14] D.G. Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.
 - [15] D. Marr and T. A. Poggio. A computational theory of human stereo vision. *Proceedings of Royal Society of London*, B-204:301–328, 1979.
 - [16] K.B. Petersen and M.S. Pedersen. The matrix cookbook. *Technical University of Denmark*, 2008.
 - [17] Edward B. Saff, Arthur David Snider, Lloyd N. (Lloyd Nicholas) Trefethen, and Tobin A. (Tobin Allen) Driscoll. *Fundamentals of complex analysis: with applications to engineering and science*. Prentice-Hall, pub-PH:adr, third edition, 2003.
 - [18] D. Scharstein and R. S. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International Journal of Computer Vision*, 47(1-3):7–42, April 2002.
 - [19] D.E. Taylor. *The geometry of the classical groups*, volume 9. Heldermann Verlag, 1992.
 - [20] F. Van der Heijden, R.P.W. Duin, D. De Ridder, and DMJ Tax. *Classification, parameter estimation and state estimation*. Wiley Online Library, 2004.