David Ferstl

Improved 3D from Time-of-Flight Imaging

DOCTORAL THESIS

to achieve the university degree of
Doktor der technischen Wissenschaften

submitted to

Graz University of Technology

Supervisor

Prof. Dr. Horst Bischof
Institute for Computer Graphics and Vision
Graz University of Technology

Prof. Dr. Richard Bowden
Centre for Vision, Speech and Signal Processing
University of Surrey

Graz, Austria, November, 2016

Research is what I'm doing when I don't know what I'm doing.

Wernher Magnus Maximilian Freiherr von Braun
(1921 - 1977)

Abstract

Low cost depth imaging has the potential to revolutionize the way we interact with technical devices in 3D. The most promising device for depth acquisition is the Time-of-Flight technology. Thanks to an adoption by the gaming industry it is on its way into the mass market. Although the Time-of-Flight measurement principle is well understood and technical implementations are already available for several years, its measurement quality and range of possible applications reached a barrier due to limited packet size and energy consumption as well as its physical limitations. The sensors deliver dense depth maps from the scene with very high frame rates but the measurements are prune to high noise and low resolution.

In this thesis we meliorate these shortcomings by novel computer vision methods based on energy minimization and machine learning with the Time-of-Flight sensors. Furthermore, we present methods to fuse the sensor output with other sensor modalities and time-sequential sensing. These novel methods increase the quality of measured data of depth camera outputs in different ways. On the one side we propose a computer vision model to reduce the high acquisition noise (lateral resolution) and image size (spacial resolution) by a depth superresolution framework. For a successful superresolution we propose to fuse different sensor modalities in the optimization model. Furthermore, we show how learned models are included in the superresolution which aid the single image superresolution. With this novel approaches we achieve an image resolution of up to 16 times the original size while the quality is drastically increased. On the other side we propose a model to increase the depth information gain of acquisition sequences. We introduce an optimization framework to accurately measure the metric 3D motion between consecutive depth acquisitions. With the additional 3D motion, known as scene flow, we show how depth sensors can be used for the position and movement of the sensor in space, the 3D reconstruction of complex scenes as well as super-resolving depth measurements of dynamic scenes in time. The scene flow estimation is formulated as convex energy optimization problem by using the projective calibrated camera model. For a successful

motion estimation we define novel matching terms adapted from stereo imaging and 3D reconstruction.

In an extensive evaluation we show that with the proposed methods we can improve the quality and the quantity above of the traditional depth measurements. Consequently, these novel methods enable researches to increase the accuracy and robustness of all kinds of depth sensing applications such as object reconstruction, gesture recognition, robotic navigation and autonomous driver assistance.

This work was supported by *Infineon Technologies Austria AG*, the Austrian Research Promotion Agency (FFG) under the *FIT-IT Bridge* program, project #838513 (TOFUSION).

Kurzfassung

Die Bilderfassung mit Tiefenkameras hat das Potential den Umgang und die Möglichkeiten wie wir mit technischen Geräten interagieren von Grund auf zu verändern. Durch den großen Einsatz im Gaming-Bereich und die damit verbundene Verbreitung ist die vielversprechendste Technologie zur Aufnahme von Tiefendaten die Time-of-Flight Kamera. Das Messprinzip von Time-of-Flight Kameras wird schon seit längerer Zeit erforscht. Jedoch sind aufgrund von physikalischen Grenzen die Qualität und die Reichweite der Aufnahmen begrenzt. Die Sensoren liefern dichte Tiefenbilder der Szene mit sehr hohen Frame-Raten. Der Nachteil der Technologie ist ein sehr hoher Anteil an Rauschen und sehr geringer Bild-Auflösung.

In dieser Arbeit arbeiten wir an neuartigen Algorithmen um diese Nachteile der Time-of-Flight Technologie zu kompensieren und gleichzeitig nicht auf die Vorteile zu verzichten. Die Computer-Vision Methoden basieren hierbei auf Energie-Minimierungs-Modellen und Maschinellem Lernen. Darüber hinaus stellen wir eine Kombination mit anderen Sensoren vor um die Qualität weiter zu verbessern. Im ersten Teil dieser Arbeit stellen wir eine Methode vor um die Auflösung von Time-of-Flight Tiefenmessungen zu erhöhen. In einem mathematischen Optimierungsverfahren wird die räumliche Auflösung der Bilder drastisch erhöht während das Kamera-Rauschen um ein vielfaches verringert wird. Um die Qualität weiter zu verbessern zeigen wir wie das bestehende Optimierungsmodell mit anderen Sensoren kombiniert werden kann. Mit unserem Modell erreichen wir eine bis zu 16-fache Erhöhung der Bildauflösung während Rauschen und Messfehler um ein vielfaches verringert werden. In einem zweiten Teil zeigen wir ein Modell um die Genauigkeit der Messungen in Aufnahmesequenzen zu erhöhen. Hier präsentieren wir ein mathematisches Optimierungs-Modell um die exakte metrische Bewegung von beobachteten Punkten einer Szene zu messen, bekannt als Szenen-Fluss. Mit der zusätzlichen Information über die Bewegung in einer Szene ist es einfach die Auflösung in dynamischen Szenen zu erhöhen sowie komplexe Szenen zu rekonstruieren. Wir haben die Messung des Szenen-Flusses als

konvexes Optimierungsproblem formuliert wobei das Mathematische Bewegungsmodell als Projektion und Rückprojektion in einer Kalibrierten Tiefenkamera realisiert ist. In einer Umfangreichen Auswertung dieser Methoden zeigen wir, dass die Qualität gegenüber allen bisherigen Methoden verbessert werden kann. Mit der Verbesserung der Tiefenmessung und der Erhöhung des gemessenen Informationsgehalts werden auch alle darauf aufbauenden Methoden verbessert oder überhaupt ermöglicht. Diese Methoden reichen von Objektrekonstruktion/Erkennung, Gesichts und Gestenerkennung sowie Roboter Navigation und autonomer Fahrerassistenz in Autos.

Diese Arbeit wurde unterstützt von *Infineon Technologies Austria AG* und der Österreichischen Forschungsförderungsgesellschaft (FFG) im Rahmen des *FIT-IT Bridge* Programms, Projekt #838513 (TOFUSION).

Statutory Declaration

I declare that I have authored this thesis independently, that I have not used other than the declared sources/resources, and that I have explicitly marked all material which has been quoted either literally or by content from the used sources.

The text document uploaded to TUGRAZonline is identical to the presented doctoral thesis.

Place	Date	Signature

Eidesstattliche Erklärung

Ich erkläre an Eides statt, dass ich die vorliegende Arbeit selbstständig verfasst, andere als die angegebenen Quellen/Hilfsmittel nicht benutzt, und die den benutzten Quellen wörtlich und inhaltlich entnommene Stellen als solche kenntlich gemacht habe.

Das in TUGRAZonline hochgeladene Textdokument ist mit der vorliegenden Dissertation identisch.

Ort	Datum	Unterschrift

Acknowledgments

First and foremost I would like to thank my supervisor, Horst Bischof. In his great lectures, he first aroused my interest in computer vision in the early stages of my studies. After my master's thesis under his supervision he gave me the opportunity to work as a researcher at the Institute for Computer Graphics and Vision (ICG). I am also very grateful to Richard Bowden for being my secondary examiner.

Furthermore I want to thank all the colleagues at at the Institute for Computer Vision and Graphics (ICG). A special thanks to Chrisitan Reinbacher and Gernot Riegler with which I had the honor of working more closely together. We shared interesting ideas and great projects that I am exceptionally proud of.

I further want to thank the Robot Vision Lab where I could proudly say that I was a part of. Thanks Matthias Rüther for your support, guidance in all our projects as well as all the prove-reading work of all the papers we could publish.

A big hand also for the reading group where I not only have learned about all the different areas of computer vision but also to look at my own problems from a different perspective. With all their members we learned to work through and understand all kinds of scientific papers, and of course also to look behind the written. Special thanks to my colleagues Christian, Gernot, Samuel, Rene, Kö, Martin and Michael for the fruitful discussions and inputs not only at the institute but also at some drinks after work. Further, I was glad to be part of the coffee group and the ICG Soccer Team which were a perfect place for a little break and to regenerate during and after work.

I am deeply indebted to my industrial project partner *Infineon Technologies Austria AG* and its staff. I want to thank all of them for providing know-how, industrial insights, hardware, software and finally a pleasant cooperation. Without this cooperation this theses would not have been possible.

Last but not least, I want to thank my family and friends for giving me the mental and social support that enabled my studies. They always believed in me and encouraged me in every phase of my work.

Contents

List of Figures

List of Tables

2 Motivation

Contents

The enhancements and quality improvements of intensity images by means of computer vision methods have a long history. Most current smartphones use built in systems for denoising, superresolution and motion estimation running in real-time. Along with these low-level quality improvements also the quality of algorithms building on the improved camera output increase such as face recognition and different applications for image retrieval.

Recently, depth sensing became more important in computer vision. Once prohibitively expensive range sensors reached their way to the mass market. Now, depth cameras range from expensive Lidar systems to very affordable consumer systems, such as the Microsoft Kinect, Asus Xtion Pro or the Creative Senz3D camera. These new depth cameras bring a new sense of metric distance into the field of low-cost computer vision. Many traditional computer vision problems have been simplified, such as 3D reconstruction, object recognition, gesture recognition or pose estimation. Currently, depth sensors are an essential part of many real-world applications such as human-computer interaction, robotic navigation or autonomous driving.

When it comes to implementing consumer products, there is always a tradeoff between package size, quality and price. Most of these cameras reliably deliver dense or semi-dense depth images of a scene, where every pixel gives the metric distance of the sensor pixel to a 3D scene point. On the downside, these sensors are limited in their spatial resolution and suffer from acquisition noise. Therefore, there is a huge need for cheap software solutions to improve the quality and information richness of the depth data.

In this thesis we propose several methods to enhance the depth output of these cameras. Our research mostly on two directions. First, we show different approaches for the *superresolution* of single depth images. We propose a formulation of the general superresolution as a convex optimization problem. The superresolution in our model covers an increase of the lateral resolution (noise in the image) and an extension of the spatial resolution (image size). In this context we further propose to combine the depth sensor with other sensing modalities to increase the superresolution quality. We show a method to drastically increase the resolution by a fusion of the low resolution depth image with a high resolution intensity image or a learned upsampling model.

Second, we show a method to increase the acquisition quality of image sequences by an estimation of the metric 3D motion in space, namely the estimation of *scene flow*. Measured depth and dense scene motion are essential to characterize a dynamic scene. Through the information of structure and motion in a scene both the movement of the camera as well as the movement of objects can be determined. Furthermore, also a temporal superresolution becomes possible.

The technology behind consumer depth cameras can be divided into active stereo and Time of Flight (ToF) technology. Throughout the thesis we focus on depth measurements based on the *ToF* principle due to its advantages and general applicability. In the following sections we will briefly discuss the advantages and disadvantages of this sensor technology in Section 2.1. Further we will give a short description of the problems we are facing in the following methods and give a short outline of the rest of the thesis in Section 2.2.

2.1 Why Time-of-Flight?

One of the most promising sensor principles for modern consumer depth cameras is the *ToF* sensor. It measures a per-pixel depth through the runtime of light. They deliver a dense depth map where every pixel delivers the metric distance from the sensor pixel to the observed point in space. This measurement is done by emitting and capturing a continuous-wave modulated light usually in the near infrared range (NIR). The distance of the camera pixel to the scene point is given by the phase difference between the emitted and the received signal, as shown in Figure 2.1. Due to this active measurement principle the measurements are independent of scene texture and largely independent of environmental lighting conditions. Compared to (active/passive) stereo sensors the *ToF* principle requires only one sensor. The emitting device is a Light-emitting Diode (LED). Therefore, no camera baseline is required which leads to small packet sizes and dense depth maps without occluded areas, even at very close ranges. Further, for the depth measurement no expensive calculations such as feature matching and triangulation are necessary. The depth information is readily available from the sensor pixel which results in depth acquisitions at very high frame rates (up to 160fps). Due to their small package size and low energy consumption these cameras are well suited for mobile devices as shown at the Google Tango smartphone.

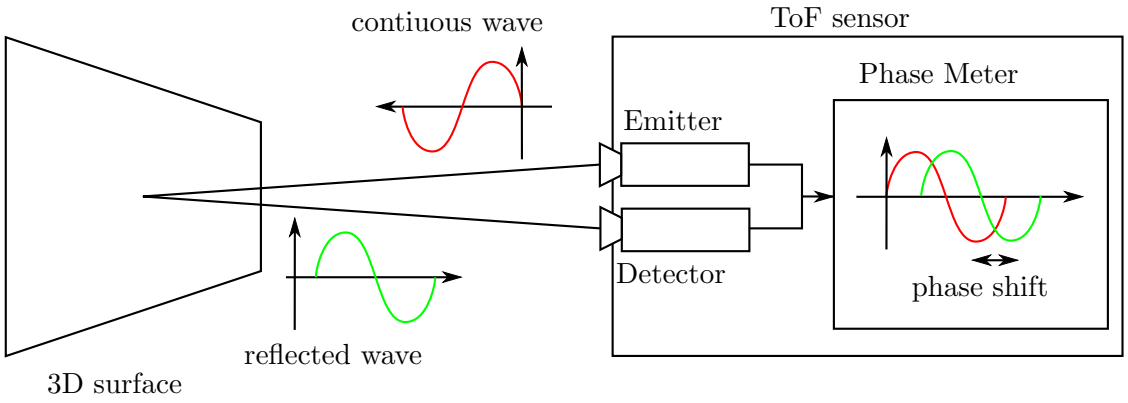


Figure 2.1: Time-of-Flight (ToF) measurement principle. Emitted and detected a continuous-wave modulated light. The distance of the 3D surface is measured by the phase shift between emitted and detected wave.

In contrast to the promising advantages of the *ToF* sensor, there exist several drawbacks of this technology. The *ToF* sensor captures the phase difference and therefore the resulting depth for each pixel. One pixel, called Photonic Mixing Device (PMD), directly measures the correlation between the emitted and received signal. Its high complexity is a limiting factor regarding the spatial resolution of the sensor. Although there has been a massive increase of pixels density, this development is reaching its physical limitations. Current state of the art sensors have a maximum of $100k$ pixels, compared to several megapixels which are reached for intensity imaging. Another limitation comes with the principle of an active illumination. The low dynamic range of these sensors results from the fact that the *PMD* pixels are linearly sensitive to incident light whereas the amplitude of an emitted light signal decreases with the square of the distance. The measurement quality is further dependent on the ratio between the signal strength of the reflected *NIR* signal and environmental light (Signal-to-Noise Ratio). This low dynamic range leads to high noise in the depth measurements, where the noise increases with the distance. Apart of these main problems there exist other drawbacks such as a limited working range due to the periodicity of the modulated signal and biased measurements due to multi-path reflections of the emitted light.

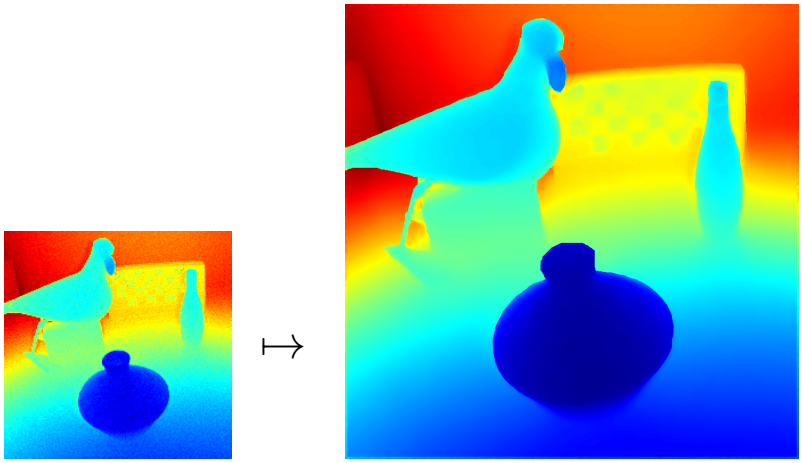
Overall, the *ToF* sensor seems to be the most promising choice for todays consumer depth cameras. In our work we focus on methods to increase the limited spatial resolution and reduce the high acquisition noise of this *ToF* sensors. Using the 3D measurements of *ToF* cameras we further show novel approaches to estimate dense 3D motion over time to increase the temporal resolution. Since most depth cameras come equipped with an additional intensity camera with a comparable higher resolution, we show several approaches to incorporate this information into our model to further increase the quality.

2.2 Problem Description and Outlook

When facing the quality improvements of *ToF* depth measurements we first have to state the underlying problem. In this section we give a rough overview of our definition of depth image superresolution and state the problem of scene flow estimation.

2.2.1 Increase the Depth Image Resolution

The goal of depth image superresolution is to find a high resolution (HR) estimate of a single low resolution (LR) depth measurement. In Figure 2.2 we have shown the superresolution of an exemplar *ToF* acquisition and a superresolved estimate.



(a) LR input **(b)** HR estimate

Figure 2.2: Depth superresolution problem on real *ToF* data from [66]. The goal is to calculate a high resolution estimate (b) of a given a low resolution depth measurement (a) (depth is color coded).

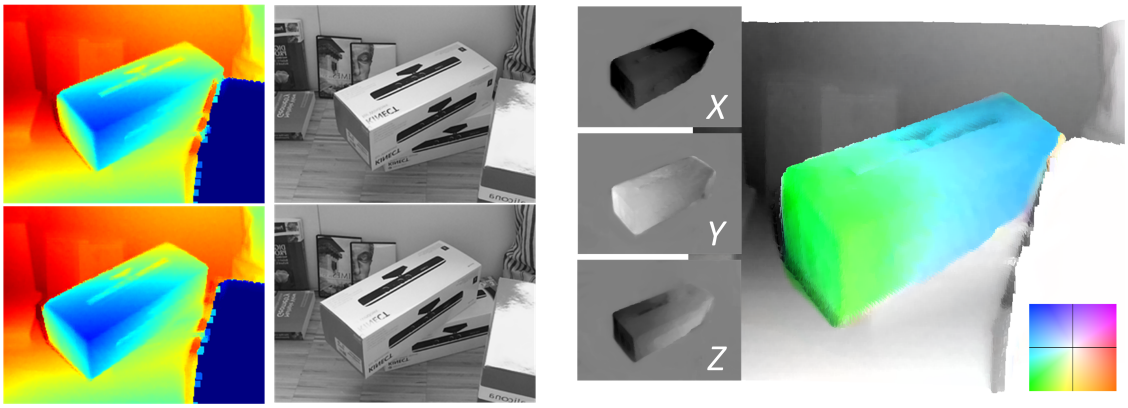
The formulation of an image superresolution is to model the image formation process from a high resolution to a low resolution exemplar. This image formation is defined by a fixed downsampling factor and a known point spread function added by an unknown amount of noise. The goal of superresolution is to estimate the inverse image formation which is extremely ill-posed since for a given low resolution input infinitely many high resolution images satisfy this model.

While the field of image superresolution is a widely researched area in computer vision, the superresolution of depth images is a quite new field. Although the main principle is the same, the underlying modalities and their internal problems are quite different. While intensity acquisitions include a lot of texture and comparably low acquisition noise the depth map of a scene mostly consists of piece-wise planar surfaces with a comparably high amount of acquisition noise. Hence, also the estimation procedure has to be chosen

appropriately. In our work we formulate the superresolution problem as a convex energy minimization and define specific regularization terms to reconstruct depth images. Further, we show different approaches where additional data from an intensity camera or a learned model is used to support the optimization.

2.2.2 Scene Flow Estimation

In contrast to depth superresolution where the measurement quality is increased for a single acquisition, the field of scene flow estimation aims to increase the amount of information over time. Most of todays consumer depth sensors come equipped with an additional intensity camera or deliver an additional infrared image of the scene. Hence, the fundamental goal for a successful scene flow estimation is to calculate a motion field of an observed scene using consecutive depth and intensity acquisitions. Given a set of two consecutive acquisitions the scene flow is defined by the metric 3D movement of every observed scene point from the first frame into the second frame, as shown in Figure 2.3.



(a) Depth Input **(b)** Intensity Input **(c)** Scene Flow in X, Y and Z

Figure 2.3: Scene Flow Problem. Given two consecutive depth and intensity acquisitions (a,b) the goal of scene flow estimation is to accurately estimate the 3D motion in X, Y and Z direction (c). The left side of (c) shows the intensity coded flow in each direction (black negative, white positive) and the right side shows the color coded 2D motion rendered on the depth map (2D color key in the bottom right).

Scene flow estimation builds on the assumption that the intensity and depth value of an object at a given object position does not change over time. Hence, the main idea of flow estimation is to find a motion vector for each scene point such that the difference of the pixel intensity and depth between the acquisitions is minimized according to the motion. By using the depth information this motion vector is defined as the metric 3D motion in space. Even if this assumption holds, this estimation is extremely ill-posed. Real camera acquisitions are affected by noise and illumination changes. Hence, in a successful

estimation additional constraints on noise, illumination changes and constancy have to be defined in the model.

2.2.3 Outlook

In the following chapters we present methods for depth image superresolution and scene flow estimation to solve the above problems and give a deep insight of their properties and principles.

In Chapter 3 we start with a definition of the mathematical background which is used throughout the thesis. We define the mathematical notation and give an overview of the geometric camera models as well as the mapping between different cameras. Further, we give a short introduction to convex analysis and optimization given a variational model.

Next, we propose different methods for depth image superresolution in Chapter 4. We first define the basic superresolution model and how it is solved using convex optimization. Further, we show how an additional high resolution intensity image or a learned model is used to guide the optimization process to increase the accuracy. In an evaluation based on different real and synthetic datasets we show the performance of the proposed superresolution models compared to the State-of-the-Art (SoA) in this field.

Third, in Chapter 5, we propose a basic variational model to estimate dense scene flow from consecutive acquisitions. We extend the basic model to increase the accuracy and robustness to noise and illumination changes during the acquisitions. The chapter is concluded by an extensive evaluation of each part of our scene flow model with a comparison to current *SoA* methods in this field.

Finally, we give a conclusion of our work and propose different ideas for extensions and future work, in Chapter 6.

3 Mathematical Background

Contents

In this thesis we show novel approaches to improve the quality and robustness of current depth sensors using optimization models. This chapter is devoted to give a brief overview on the mathematical framework for the presented models. We first introduce the common basic notations, norms and operations used throughout the thesis. Further, we will introduce the image formation process when a camera acquires an image, the projective camera model between 3D world and 2D image space as well as the mapping between different 2D and 2.5D camera models. We conclude this chapter with a short review on variational methods commonly used for imaging problems and present some basics on convex analysis and optimization used as foundation in our methods.

3.1 Notation

In this section we give an overview of the mathematical notations and definitions. This includes the spaces, norms and operations that are used throughout the thesis.

Entity	Notation
Scalar	$a,\ b_i$
Position in 2D space	$\mathbf{x} = (x, y)^{\mathrm{T}}$
Position in 3D space	$\mathbf{X} = (X, Y, Z)^{\mathrm{T}}$
N-dimensional vector	$\mathbf{a} = (a_1, \ldots, a_N)^{\mathrm{T}}$
Matrix	$\mathbf{A} = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$

Table 3.1: Basic notations. Points in space, scalar, vector and matrix.

The most basic notation of scalars, vectors and matrices shown in Table 3.1. According to this definition the set of real N-dimensional vectors is devoted as $\mathbb{R}^N$ and the set of matrices of size $M \times N$ is $\mathbb{R}^{M \times N}$. In this context we use the vectorization operator $\mathrm{vec}(\mathbf{A}) \in \mathbb{R}^{MN}$ to indicate a row-major vector representation of the matrix $\mathbf{A} \in \mathbb{R}^{M \times N}$.

Definition: The Inner Product Given the vectors $\mathbf{a}, \mathbf{b}, \mathbf{c} \in X$, where X is a finite dimensional vector space $X = \mathbb{R}^M$, the inner product is a function $\langle . \rangle_X \colon \mapsto X \times X \mapsto \mathbb{R}$ defined by the following conditions

$$\langle \mathbf{a}, \mathbf{b} \rangle_X = \langle \mathbf{b}, \mathbf{a} \rangle_X \tag{3.1}$$

$$\langle \mathbf{a}, \mathbf{a} \rangle_X \geq 0 \tag{3.2}$$

$$\langle \mathbf{a}, \mathbf{b} + \mathbf{c} \rangle_X = \langle \mathbf{a}, \mathbf{b} \rangle_X + \langle \mathbf{a}, \mathbf{c} \rangle_X \tag{3.3}$$

Further, two vectors $\mathbf{a}$ and $\mathbf{b}$ are called orthogonal if $\langle \mathbf{a}, \mathbf{b} \rangle_X = 0$.

Definition: Normed Vector Spaces Let X be a finite dimensional vector space. Given a vector $\mathbf{a} \in X$ and a vector $\mathbf{b} \in X$, a semi-norm on X is a function $f \colon X \mapsto \mathbb{R}_{\geq 0}$ if

$$\begin{aligned} f(\lambda \mathbf{a}) &= |\lambda| f(\mathbf{a}) \qquad \forall\ \lambda \in \mathbb{R} \\ f(\mathbf{a} + \mathbf{b}) &\leq f(\mathbf{a}) + f(\mathbf{b}). \end{aligned} \tag{3.4}$$

If the function f additionally satisfies

$$f(\mathbf{a}) = 0 \qquad \text{iff} \quad \mathbf{a} = \mathbf{0} \tag{3.5}$$

then f is called norm $\|.\|$. This can be seen as a generalization of $|.|$ on $\mathbb{R}$. A vector space together with a norm is called a normed vector space $(X, \|.\|)$.

The ℓ_p-norm of $\mathbf{a}$ is defined as

$$\|\mathbf{a}\|_p = \left(\sum_{i=1}^{M} |a_i|^p \right)^{1/p} \qquad \forall\, 1 \leq p < \infty. \tag{3.6}$$

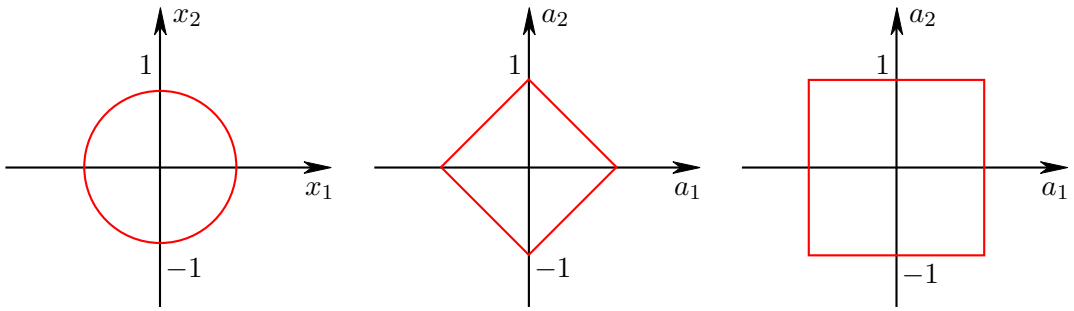
Common cases of the ℓ_p-norm are the ℓ_2-norm (Euclidean) and the ℓ_1-norm (Manhattan) which are defined as

$$\|\mathbf{a}\|_2 = \sqrt{\mathbf{a}^T \mathbf{a}} \qquad \text{and} \qquad \|\mathbf{a}\|_1 = \sum_{i=1}^{M} |a_i| \tag{3.7}$$

The case for $p = \infty$ is called the Chebyshev- or ℓ_∞-norm is given by

$$\|\mathbf{a}\|_\infty = \max\left(|a_1|, |a_2|, \ldots, |a_M| \right) \tag{3.8}$$

An illustration of unit cycles of the different norms in $\mathbb{R}^2$ is shown in Figure 3.1.



(a) $\|\mathbf{a}\|_2 \leq 1$ **(b)** $\|\mathbf{a}\|_1 \leq 1$ **(c)** $\|\mathbf{a}\|_\infty \leq 1$

Figure 3.1: Illustration of the unit norm ball for different norms in $\mathbb{R}^2$.

It is also possible to define a mixture of different norms. A popular example is the Huber-norm [51] which combines the ℓ_2-norm and the ℓ_1-norm and is defined as

$$|q|_\varepsilon = \begin{cases} \frac{|q|^2}{2\varepsilon} & \text{if} \quad |q| \leq \varepsilon \\ |q| - \frac{\varepsilon}{2} & \text{if} \quad |q| > \varepsilon \end{cases}, \tag{3.9}$$

where $\varepsilon \in \mathbb{R}_{\geq 0}$ defines the Huber parameter that combines the properties of the ℓ_2 at values q smaller than ε and the ℓ_1 norm at larger values. In the sense of signal processing the Huber-norm introduces a Gaussian smoothing for small values and linear smoothing at higher values.

A special case is the ℓ_0-norm

$$\|\mathbf{a}\|_0 = \lim_{p\to 0} \left(\sum_{i=1}^{M} |a_i|^p \right)^{1/p} = \#\left\{a_i \neq 0\right\}_{i=1}^{M}, \tag{3.10}$$

where the number of non-zero entries (sparsity) in a vector is calculated. Note that the ℓ_0-norm does not fulfill the norm definition since it is not a semi-norm, according to Eqn. (3.4) and therefore not also not a norm (Eqn. (3.5)).

Definition: Linear Continuous Operators Let X and Y be linear spaces. Given the vectors $\mathbf{a}, \mathbf{b} \in X$, an operator $L\colon X \mapsto Y$ is called linear if

$$L(\mathbf{a}+\mathbf{b}) = L\mathbf{a} + L\mathbf{b} \tag{3.11}$$

$$L(\lambda \mathbf{x}) = \lambda L(\mathbf{x}) \quad \forall \lambda \in \mathbb{R} \tag{3.12}$$

$\mathcal{L}(X,Y)$ is the space of linear continuous operators for X and Y. If X and Y are normed vector spaces, then the norm on $\mathcal{L}(X,Y)$ is defined by

$$\|L\|_{\mathcal{L}(X,Y)} = \sup_{\|\mathbf{a}\|_X \leq 1} \|L(\mathbf{a})\|_Y = \sup_{\|\mathbf{a}\|_X \neq 0} \frac{\|L(\mathbf{a})\|_Y}{\|\mathbf{a}\|_X}. \tag{3.13}$$

This is called the operator norm.

Definition: Derivative Operator Let $F\colon X \mapsto Y$ be a map. It is called differentiable in $\mathbf{a} \in X$ if there exists an operator $DF(\mathbf{a}) \in \mathcal{L}(X,Y)$ s.t.

$$\forall \varepsilon > 0 \quad \exists \delta > 0, \quad 0 < \|\mathbf{h}\|_X < \delta, \qquad \mathbf{a}+\mathbf{h} \in X \quad \text{and} \tag{3.14}$$

$$\frac{\|F(\mathbf{a}+\mathbf{h}) - F(\mathbf{a}) - DF(\mathbf{a})\mathbf{h}\|_Y}{\|\mathbf{h}\|_X} < \varepsilon \tag{3.15}$$

The derivative operator for $F\colon \mathbb{R}^M \mapsto \mathbb{R}$ defines the gradient operator

$$DF(\mathbf{a}) = \nabla F(\mathbf{a}) = \left(\frac{\partial F}{\partial a_1}, \frac{\partial F}{\partial a_2}, \ldots, \frac{\partial F}{\partial a_M} \right) \tag{3.16}$$

for which the squared derivative (second derivative) given by

$$D^2F(\mathbf{a}) = \nabla^2 F(\mathbf{a}) = \begin{pmatrix} \frac{\partial^2 F}{\partial^2 a_1} & \cdots & \frac{\partial^2 F}{\partial a_1 \partial a_M} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 F}{\partial a_M \partial a_1} & \cdots & \frac{\partial^2 F}{\partial^2 a_M} \end{pmatrix}, \tag{3.17}$$

defines the Hessian Matrix. Similar, the Jacobian Matrix defines a matrix of all partial derivatives of a function $F\colon \mathbb{R}^M \mapsto \mathbb{R}^N$ usually defined as

$$DF(\mathbf{a}) = \nabla F(\mathbf{a}) = \begin{pmatrix} \frac{\partial F_1}{\partial a_1} & \cdots & \frac{\partial F_1}{\partial a_M} \\ \vdots & \ddots & \vdots \\ \frac{\partial F_N}{\partial a_1} & \cdots & \frac{\partial F_N}{\partial a_M} \end{pmatrix}. \tag{3.18}$$

Furthermore, we define the adjoint gradient operator as $\nabla^T\colon Y \mapsto X$ also known as the negative divergence operator given by

$$\langle \nabla \mathbf{a}, \mathbf{b} \rangle \equiv \left\langle \mathbf{a}, \nabla^T \mathbf{b} \right\rangle_X = \langle \mathbf{a}, -\operatorname{div} \mathbf{b} \rangle_X . \tag{3.19}$$

Definition: Image Space An image containing either scalar values (either intensity or depth) is defined as the spatial real valued function $\mathbf{I}(\mathbf{x})$. The 2D vector $\mathbf{x} \in \mathbb{R}^2$ denotes the spatial image coordinate defined as $\mathbf{x} = (x, y)^{\mathrm{T}}$ which defines the location on a regular Cartesian grid of size $M \times N$ given by

$$\Omega = \{(x, y)\colon 1 \leq x \leq M, 1 \leq y \leq N\}. \tag{3.20}$$

From this it follows that images are defined as $\mathbf{I}\colon (\Omega = \mathbb{R}^2) \mapsto \mathbb{R}$. Hence, the discrete pixel position can also be described as (x, y). Using the previously defined gradient operator and discrete gradient of an image becomes

$$\nabla \mathbf{I} = \begin{pmatrix} \nabla_x \mathbf{I} \\ \nabla_y \mathbf{I} \end{pmatrix}, \tag{3.21}$$

which represents the gradient in both directions of the defined Cartesian grid s.t. $\nabla\colon \Omega \mapsto \mathbb{R}^{2\times M\times N}$. In a numerical scheme the gradient operators ∇_x, ∇_y are calculated by standard forward differences on the regular grid with Neumann boundary conditions given by

$$\begin{aligned} \nabla_x \mathbf{I}(x, y) &= \begin{cases} \mathbf{I}(x+1, y) - \mathbf{I}(x, y) & \text{if} \quad x < M \\ 0 & \text{else} \end{cases} \quad \text{and} \\ \nabla_y \mathbf{I}(x, y) &= \begin{cases} \mathbf{I}(x, y+1) - \mathbf{I}(x, y) & \text{if} \quad y < N \\ 0 & \text{else} \end{cases}. \end{aligned} \tag{3.22}$$

Given the gradient result $\mathbf{P} \in \mathbb{R}^{2\times M\times N}$ the inverse to the gradient is given by the adjoint gradient operator as $\nabla^T\colon \mathbb{R}^{2\times M\times N} \mapsto \Omega$ calculated as standard backward differences

$$\nabla^T \mathbf{P} = \nabla_x^T \mathbf{P}^1 + \nabla_y^T \mathbf{P}^2. \tag{3.23}$$

The backward differences with Dirichlet boundary condition are defined as

$$\nabla_x^T P^1(x,y) = \begin{cases} P^1(x,y) & \text{if} \quad x = 1 \\ P^1(x,y) - P^1(x-1,y) & \text{if} \quad 1 < x < M \\ -P^1(x-1,y) & \text{if} \quad x = M \end{cases} \quad \text{and}$$
$$\nabla_y^T P^2(x,y) = \begin{cases} P^2(x,y) & \text{if} \quad y = 1 \\ P^2(x,y) - P^2(x,y-1) & \text{if} \quad 1 < y < N \\ -P^2(x-1,y) & \text{if} \quad y = N \end{cases}. \tag{3.24}$$

3.2 Camera Model

In our work we are dealing with one or more cameras which image the scene. In this section we introduce the standard camera model and image formation process used throughout all our methods called the *finite projective camera*. In our work we distinguish between two types of camera which differ in the sort of information they acquire. An intensity camera acquires intensity values for each point in the field of view while a depth camera acquires information about the absolute or relative distance between the camera and each point in the field of view. In principle a standard pinhole camera model as shown in the seminal work of Hartley and Zisserman [42] applies to both of our camera types. The pinhole camera model is depicted in Figure 3.2.

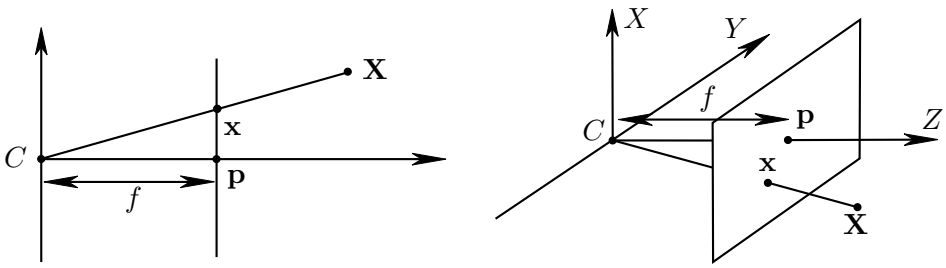


Figure 3.2: Pinhole camera model. The camera model with projection center **C** at the coordinate origin.

Definition: Central Perspective Projection The pinhole model defines the mapping of points in space onto a plane. Let the center of the our model be the origin of a Euclidean coordinate system we define this plane at depth $Z = f$. This plane is called the *image plane*. The central projection of a point $\mathbf{X}^h = (X, Y, Z, W)^{\mathrm{T}}$ in space onto the image point $\mathbf{x}^h = (x, y, w)^{\mathrm{T}}$ represented as a homogeneous 3-vector. This projection onto the *image*

plane can be described by

$$\mathbf{x}^h = \begin{pmatrix} x \\ y \\ w \end{pmatrix} = \begin{pmatrix} f_x & 0 & p_x & 0 \\ 0 & f_y & p_y & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} = \mathbf{K} \begin{pmatrix} \mathbf{I} & \mathbf{0} \end{pmatrix} \mathbf{X}^h, \tag{3.25}$$

where the vector $\mathbf{f} = (f_x, f_y)^{\mathrm{T}}$ is the *focal length* and $\mathbf{p} = (p_x, p_y)^{\mathrm{T}}$ is the *principle point offset* (both in pixels) defining the *camera calibration matrix* $\mathbf{K}$.

Definition: Projection in Space To describe the central perspective projection it is assumed that the *camera coordinate system* is the coordinate center of all observed points. In general this does not hold since points are often in a *world coordinate system* (especially when multiple cameras observe a scene). Hence, the *camera* and the *world coordinate system* are related by a rotation and a translation, as shown in Figure 3.3. Together with Eqn. (3.25) a projection of a sample point $\mathbf{X}$ in space is described as

$$\mathbf{x}^h = \mathbf{K}\mathbf{R} \begin{pmatrix} \mathbf{I} & -\mathbf{C} \end{pmatrix} \mathbf{X}^h = \mathbf{P}\mathbf{X}^h, \tag{3.26}$$

where $\mathbf{I}$ is the identity matrix, $\mathbf{R} \in \mathrm{SO}(3)$ defines the rotation and $\mathbf{C} \in \mathbb{R}^3$ the translation of the camera with reference to the world coordinate system. The parameters contained in $\mathbf{K}$ are called the *intrinsic parameters* of the camera and the parameters $\mathbf{R}$ and $\mathbf{C}$ are called the *extrinsic parameters* which relate the camera orientation and position in the world coordinate system. The *camera calibration matrix* together with the rotation and translation forms the *camera matrix* $\mathbf{P}$. Note: to get the Euclidean image point $\mathbf{x}$ the result has to be normalized by its homogeneous coordinate w.

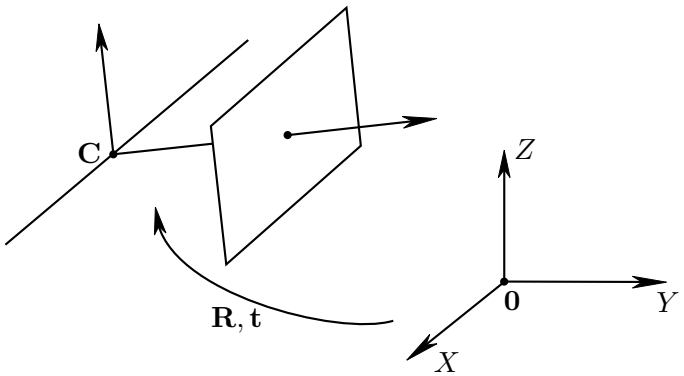


Figure 3.3: Transformation between the *camera* and the *world coordinate system* through rotation and translation.

Definition: Back-Projection into Space Similar to the projection of 3D points into the 2D image space one can back-project an image point to a ray in space. The ray in 3D space may be specified in several representations, e.g. *Plücker* representation. In this thesis a 3D ray is represented as the joint of a known point and a direction vector. The point in this representation is the camera center $\mathbf{C}$. Assume a central perspective projection, where the camera center lies in the origin of the Euclidean coordinate system, then the ray into space is defined by

$$\mathbf{X}(\lambda) = \lambda \mathbf{K}^{-1} x^h, \tag{3.27}$$

where λ defines the position on the ray. Given either the rotation matrix $\mathbf{R}$ and the camera center $\mathbf{C}$ the general back-projection of points to rays is defined by

$$\mathbf{X}(\lambda) = \lambda (\mathbf{RK})^{-1} \mathbf{x}^h + \mathbf{C}. \tag{3.28}$$

Alternatively if only the *camera matrix* $\mathbf{P}$ is known it can be decomposed into its left hand 3×3 submatrix and its last column such that $\mathbf{P} = \begin{pmatrix} \mathbf{M} & \mathbf{p}^4 \end{pmatrix}$. Hence, we can easily substitute the unknown parts of Eqn. (3.28) such that $\mathbf{M} = \mathbf{KR}$ and $\mathbf{M}^{-1}\mathbf{p}^4 = \mathbf{C}$. Note that by this definition the rotation matrix and the camera projection matrix can be decomposed by RQ matrix decomposition of $\mathbf{M}$. Furthermore, from the definition $\mathbf{PC} = \mathbf{0}$ it follows that the camera center is defined as the right null space of $\mathbf{P}$ found by Singular Value Decomposition (SVD).

If the depth d at pixel $\mathbf{x}$ is known from e.g. the pixel value of a depth camera the corresponding 3D point is calculated with Eqn. (3.28). Hence, the exact 3D point in space is defined by

$$\mathbf{X} = \frac{(\mathbf{RK})^{-1}\mathbf{x}^h}{\|(\mathbf{RK})^{-1}\mathbf{x}^h\|_2} d(\mathbf{x}) + \mathbf{C}, \tag{3.29}$$

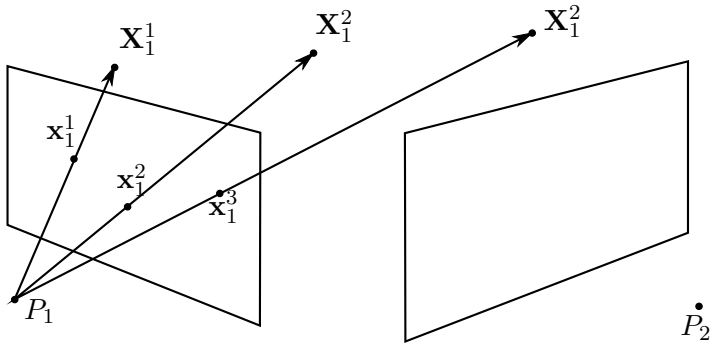
where the viewing ray $(\mathbf{RK})^{-1}\mathbf{x}^h$ is normalized to unit length.

Mapping between Images In our work we often use the projection and back-projection together to map depth or intensity values from one camera coordinate system to the other. Mathematically this is done by combining Eqn. (3.29) to project into 3D space and Eqn. (3.26) to project back into the next image space.

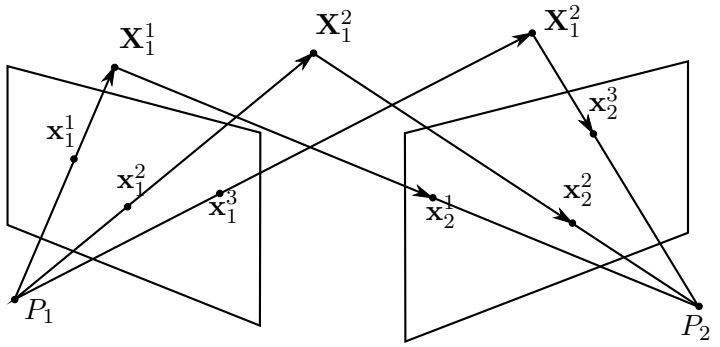
Given two calibrated cameras $1, 2$ defined by a subscript on the intrinsic and extrinsic parameters $\mathbf{P}$, $\mathbf{K}$, $\mathbf{R}$ and $\mathbf{C}$ the mapping from image space Ω_1 to Ω_2 is given by

$$\mathbf{x}_2^h = \mathbf{P}_2 \left(\frac{(\mathbf{R}_1\mathbf{K}_1)^{-1}\mathbf{x}_1^h}{\|(\mathbf{R}_1\mathbf{K}_1)^{-1}\mathbf{x}^h\|_2} d(\mathbf{x}_1)_1 + \mathbf{C}_1 \right), \tag{3.30}$$

given all depth measurements $d_1 \in \Omega_1$. This mapping is shown in Figure 3.4 for some sample points. In the case of intensity mapping the pixels in Ω_2 take over the intensity



(a) Back-projection from 2D into 3D space



(b) Projection from 3D into the second image system

Figure 3.4: Mapping between images. The mapping between two image system 1 and 2 is done through projection into 3D space followed by a back-projection into the second image.

values from Ω_1. In case of a depth mapping the new values in Ω_2 are given by the Euclidean distance $\|\mathbf{X} - \mathbf{C}_2\|_2$.

Definition: Distortion Until now we assumed a linear projection model of the imaging process. However, for real (non-pinhole) lenses this assumption does not hold. The most common distortion errors are the radial and the tangential distortions. While the radial distortion arises from the imperfection of a photographic lens and becomes more significant as the focal length decreases the tangential distortion comes from imperfect alignment between lenses and sensor. The cure for these distortions is to calibrate a mapping function which maps the distorted image positions to the image positions of a perfect linear camera model. In this thesis we use a distortion model defined by

$$\hat{x} = (x - p_x)\left(1 + \kappa_1 r + \kappa_2 r^2\right) + \left(2\xi_1 xy + \xi_2(r^2 + 2x^2)\right) + p_x \tag{3.31}$$

$$\hat{y} = (y - p_y)\left(1 + \kappa_1 r + \kappa_2 r^2\right) + \left(2\xi_1 xy + \xi_2(r^2 + 2y^2)\right) + p_y, \tag{3.32}$$

where $\hat{\mathbf{x}} = (\hat{x}, \hat{y})^{\mathrm{T}}$ is the distorted image position and $r = \sqrt{(x - p_x)^2 + (y - p_y)^2}$ is the pixel distance to the principal point. The distortion coefficients are κ_1, κ_2 for radial

distortion and ξ_1, ξ_2 to model the tangential distortion.

3.3 Introduction in Convex Analysis

In this section we show how convexity is defined and give an overview of convex optimization, focusing on the special role of variational energy minimization. The concepts which are introduced here are used throughout our work.

The field of convex optimization refers to the minimization of convex functions (or maximization of concave functions). It finds numerous applications in scientific fields such as signal processing, images processing, computational geometry or machine learning. Specifically variational methods that are based on continuous energy functionals are formulated to model a convex energy functional. There exists a number of optimization methods which are used to obtain the minimizer of such a convex functional. The very active field convex optimization has been studied for multiple centuries. Hence, we refer the interested readers to more detailed works such as Boyd and Vadenberghe [11], Rockafellar [97, 98], Nesterov [74].

A convex optimization problem is a mathematical optimization problem of the generic form

$$\min_{\mathbf{x}\in\mathbb{R}^n} f_0(\mathbf{x}) \qquad f_i(\mathbf{x}) \leq b_i,\ i = 1 \dots m, \tag{3.33}$$

where $f_0, \dots, f_m \colon \mathbb{R}^n \mapsto \mathbb{R}$ are convex functions. In this formulation f_0 is called the objective function and $f_1, \dots, f_m$ are called the constraint functions. The optimization variable is denoted as $\mathbf{x} \in C$, where C is a convex set. If a minimum of a convex optimization problem exists it is guaranteed that every local minimum is also a global minimum. In the following we will define the special properties of convex functions and give details on general convexity and convex sets.

3.3.1 Convex Sets

Given two points in space $\mathbf{x}_1 \neq \mathbf{x}_2 \in \mathbb{R}^m$ a line including these points is defined as

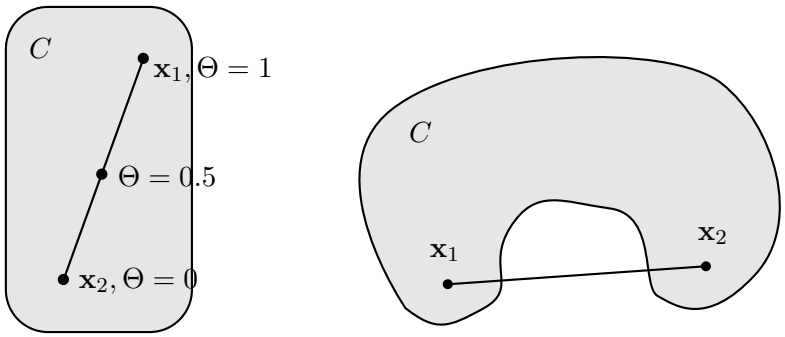
$$\mathbf{y} = \Theta\mathbf{x}_1 + (1 - \Theta)\mathbf{x}_2. \tag{3.34}$$

If the parameter value Θ is bounded in the interval $[0, 1]$ then the line function defines the (closed) line segment between $\mathbf{x}_1$ and $\mathbf{x}_2$.

Definition: Convex Set A subset $C \in \mathbb{R}^m$ is called convex set if

$$\Theta\mathbf{x}_1 + (1 - \Theta)\mathbf{x}_2 \in C \qquad \forall \mathbf{x}_1, \mathbf{x}_2 \in C,\ \Theta \in [0, 1]. \tag{3.35}$$

Meaning that if the line segment between any two points in C lies in C and $\Theta \in [0,1]$ then C is called a convex set. An example of convex and non-convex sets in $\mathbb{R}^2$ is shown in Figure 3.5. Generally, a set is called convex if every point in the set can be seen by every other point along a straight path.



(a) Convex set **(b)** Non-convex set

Figure 3.5: Example for convex and non-convex sets.

Definition: Convex Hull The convex hull of a set (convex or non-convex) C is defined as the set of all convex combinations of points in C, defined by

$$\left\{ \sum_{i=1}^{\infty} \Theta_i \mathbf{x}_i \right\}, \; with \Theta_i \geq 0, \; \sum_{i=1}^{\infty} \Theta_i = 1. \tag{3.36}$$

Consequently the convex hull of a set is always convex.

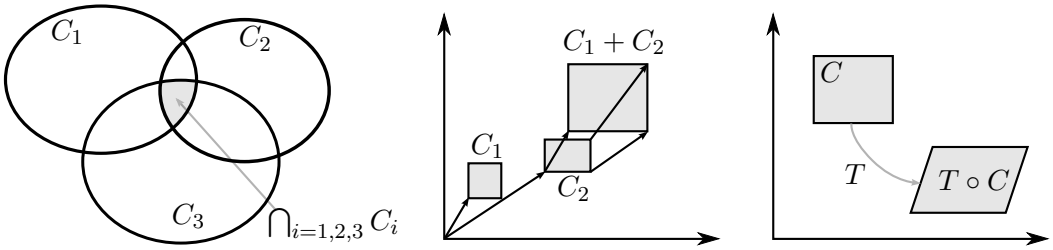
Definition: Convex Cone A set C is called convex cone if for any two points in the set $\mathbf{x}_1, \mathbf{x}_2 \in C$ we have

$$\Theta_1 \mathbf{x}_1 + \Theta_2 \mathbf{x}_2 \in C \; \forall \Theta_1, \Theta_2 \geq 0. \tag{3.37}$$

Similar to a convex hull the set of all possible conic combinations of points in C called the conic hull.

Convex set verification Operations on convex convex sets that preserve convexity:

- For any collection of convex sets the intersection $\bigcap_i C_i$ is again convex (Figure 3.6a).
- The vector sum $C_1 + C_2 = \{\mathbf{x}_1 + \mathbf{x}_2 \mid \mathbf{x}_1 \in C_1, \mathbf{x}_2 \in C_2\}$ of two convex sets C_1, C_2 (Minkowiski-Sum) is again convex (Figure 3.6b).
- The image of a convex set under affine transformation $T \circ C$ is again convex (Figure 3.6c).



(a) Intersection **(b)** Vector sum **(c)** Affine transform

Figure 3.6: Convex set verification.

3.3.2 Convex Functions

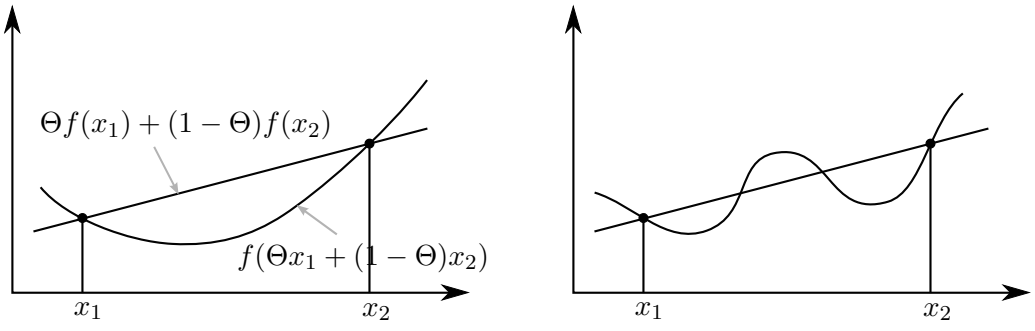
Given a convex set C, a function $f(\mathbf{x})\colon \mathbb{R}^n \mapsto \mathbb{R}$ is convex if the assumption

$$f(\Theta\mathbf{x}_1 + (1-\Theta)\mathbf{x}_2) \leq \Theta f(\mathbf{x}_1) + (1-\Theta)f(\mathbf{x}_2) \tag{3.38}$$

holds for any $\mathbf{x}_1, \mathbf{x}_2 \in C$ and $0 \leq \Theta \leq 1$. Eqn. (3.38) is also known as the Jensen's Inequality. speaking the above inequality means that the line connecting two function points $(\mathbf{x}_1, f(\mathbf{x}_1))$ and $(\mathbf{x}_2, f(\mathbf{x}_2)$ is entirely above the graph, as shown in Figure 3.7. Further, a function is called strictly convex if

$$f(\Theta\mathbf{x}_1 + (1-\Theta)\mathbf{x}_2) < \Theta f(\mathbf{x}_1) + (1-\Theta)f(\mathbf{x}_2), \tag{3.39}$$

whenever $\mathbf{x}_1 \neq \mathbf{x}_2$ and $0 < \Theta < 1$. Consequently a function is called *concave* if -f is convex and *strictly concave* if -f is strictly convex.



(a) Convex function **(b)** Non-convex function

Figure 3.7: Graph example for convex and non-convex functions. The function is convex is the line segments between two points on the graph is above the graph.

The *epigraph* of a function (epi(f)) is defined as the set

$$\text{epi}(f) = \{(\mathbf{x}, \varepsilon) \mid \mathbf{x} \in C, f(\mathbf{x}) \leq \varepsilon\}. \tag{3.40}$$

Geometrically this means that the epigraph of a function defines the area above the function, as shown in Figure 3.8. The epigraph epi(f) is a convex subset of $\mathbb{R}^{n+1}$ if $f(\mathbf{x})$ is convex on $C \subset \mathbb{R}^n$.

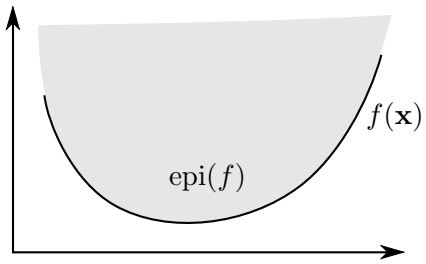


Figure 3.8: Example epigraph of a convex function $f(\mathbf{x})$.

Characterization of differentiable convex functions For the class of differentiable functions there exists an alternative characterization of function convexity based on the first and second order conditions.

Suppose $f\colon \mathbb{R}^n \mapsto \mathbb{R}$ that is differentiable over $\mathbb{R}^n$ and the domain C is a convex set. According to the first order convexity definition the function f is convex if and only if

$$f(\mathbf{x}_1) \geq f(\mathbf{x}_2) + (\mathbf{x}_1 - \mathbf{x}_2)^T \nabla f(\mathbf{x}_2), \qquad \forall \mathbf{x}_1, \mathbf{x}_2 \in C. \tag{3.41}$$

Further, f is strictly convex if and only if

$$f(\mathbf{x}_1) > f(\mathbf{x}_2) + (\mathbf{x}_1 - \mathbf{x}_2)^T \nabla f(\mathbf{x}_2), \qquad \forall \mathbf{x}_1, \mathbf{x}_2 \in C,\, \mathbf{x}_1 \neq \mathbf{x}_2. \tag{3.42}$$

The affine function $f(\mathbf{x}_2) + (\mathbf{x}_1 - \mathbf{x}_2)^T \nabla f(x_2)$ is the first order Taylor approximation (linearization) at point x_2. For convex functions it is the *global underestimatior* of the function, which is an important property of convex functions since it shows that we can derive global information from the local derivative. Hence, for convex optimization problems Eqn. (3.41) shows that if $\nabla f(\mathbf{x}_2) = 0$, then for all $\mathbf{x}_1 \in C$, $f(\mathbf{x}_1) \leq f(\mathbf{x}_2)$. global minimizer of the function f. This is a necessary and sufficient condition for $\mathbf{x}_2$ to be a global minimizer of the function f over C.

Consequently, assume that $f\colon \mathbb{R}^n \mapsto \mathbb{R}$ is twice continuously differentiable over $\mathbb{R}^n$ and the domain C is a convex set then f is convex if and only if

$$\nabla^2 f(\mathbf{x}) \succeq 0, \qquad \forall \mathbf{x} \in C \tag{3.43}$$

and strictly convex if

$$\nabla^2 f(\mathbf{x}) \succ 0, \qquad \forall \mathbf{x} \in C. \tag{3.44}$$

This means that the derivative of f is non-decreasing and hence, the function f has a

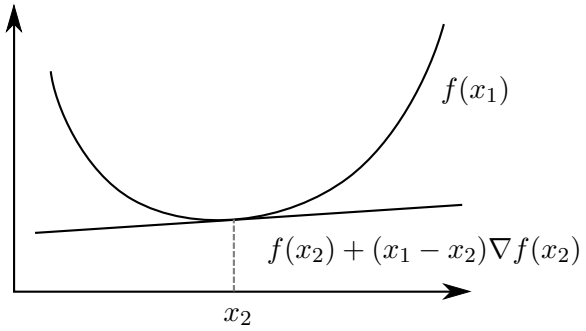


Figure 3.9: First-order Taylor approximation $f(x_2)+(x_1-x_2)^T\nabla f(x_2)$ is a *global underestimatior* of f.

non-negative curvature at any point $\mathbf{x}$.

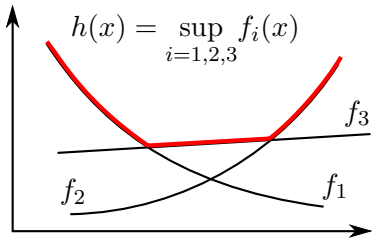


Figure 3.10: Supremum of convex function is again convex.

Properties of convex functions and operations that maintain convexity

- If a minimum of a convex function exists any local minimum is a global minimum.
- If the function is *strictly convex* the global minimum is unique.
- The weighted sum of convex functions is again convex.
- Any vector norm is convex (Section 3.1).
- The scaling of a convex function $\lambda f(\mathbf{x})$, where $\lambda \geq 0$, is convex.
- If $\mathcal{I}$ is an index set and $C \subset \mathbb{R}^n$ and $f_i \colon C \mapsto \mathbb{R} \cup \{\infty\}$ are convex functions, then the function $h \colon C \mapsto \mathbb{R} \cup \{\infty\}$ defined by the pointwise maximum $h(\mathbf{x}) = \sup_{i \in \mathcal{I}} f_i(\mathbf{x})$ is convex (see Figure 3.10).

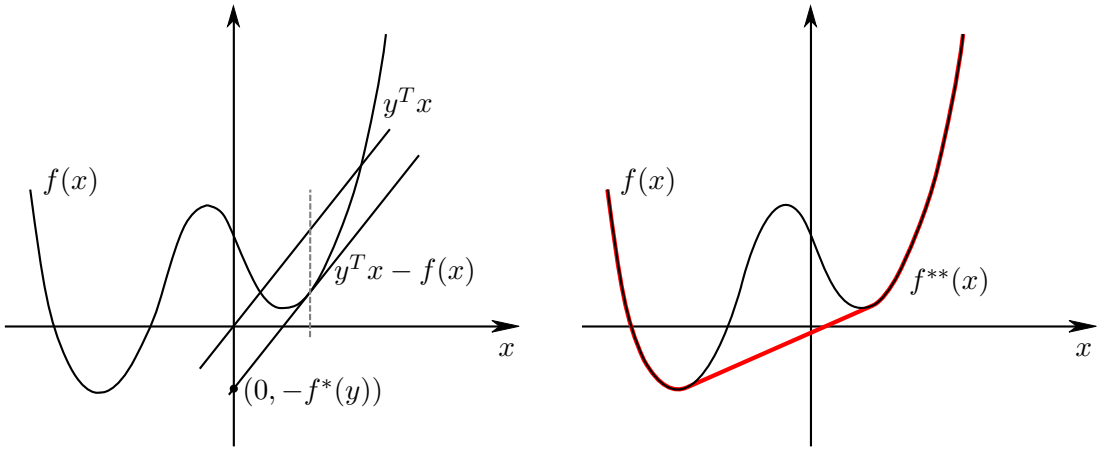
3.3.3 Convex Conjugate

In this section we introduce the convex conjugate function which is a generalization of the Legendre-Fenchel transform (LF) transform and plays an important role in the convex optimization problems in later chapters where we will stick to the terminus conjugate transform or simply dual.

Given a function $f\colon \mathbb{R}^n \mapsto \mathbb{R}$ then the function $f^*\colon \mathbb{R}^n \mapsto \mathbb{R}$ is defined as

$$f^*(\mathbf{y}) = \sup_{\mathbf{x}\in \operatorname{dom} f} \left(\mathbf{y}^T\mathbf{x} - f(\mathbf{x})\right) \tag{3.45}$$

is called the *conjugate* of f. As the name suggests, the convex conjugate is always convex and closed. As depicted in Figure 3.11a, the convex conjugate is convex by definition since it is the point-wise supremum of a set of convex (affine) functions of $\mathbf{y}$, independent on the convexity of f.



(a) Convex Conjugate **(b)** Conjugate Transform

Figure 3.11: Illustration of the conjugate transform. (a) the conjugate function f^* is the maximum between the linear function $\mathbf{y}^T\mathbf{x}$ and f. (b) the bi-conjugate f^{**} is the largest convex function below f (convex envelope).

From the definition in Eqn. (3.45) we obtain the so-called Fenchel Inequality defined by

$$f(\mathbf{x}) + f^*(\mathbf{y}) \leq \mathbf{y}^T\mathbf{x}, \qquad \forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^n. \tag{3.46}$$

The *bi-conjugate* function f^{**} is defined by

$$f^{**}(\mathbf{z}) = \sup_{\mathbf{y}\in \operatorname{dom} f^{**}} \left(\mathbf{z}^T\mathbf{y} - f^*(\mathbf{y})\right), \tag{3.47}$$

where f^* is the convex conjugate of f. The name 'bi-conjugate' may suggests that the twice conjugated function is again the original function. This proposition holds if f is convex and closed, then $f^{**} = f$ and $\mathbf{x} = \mathbf{z}$. Generally for any real function $f(\mathbf{x})$ it holds that

$$f(\mathbf{x}) \geq f^{**}(\mathbf{z}), \tag{3.48}$$

which shows that f^{**} is the largest convex function below f and thus is called the *convex envelope*, as shown in Figure 3.11b.

3.3.4 Subdifferential

In all gradient based optimizations the function f has to be differentiable at any point. Let us recall the convexity definition of Eqn. (3.41) which states that a *global underestimator* of a convex and differentiable function $f\colon \mathbb{R}^n \mapsto \mathbb{R}$ is given by the first order approximation. But in many (non-smooth) convex optimizations the function f is convex but not differentiable, e.g. the absolute value function or L1-norm. Hence, we introduce the subgradient to generalize the derivative for *non-smooth* functions. A vector g is called the subgradient of f at a point $\mathbf{x}_0$ for any $\mathbf{x} \in \operatorname{dom} f$ if

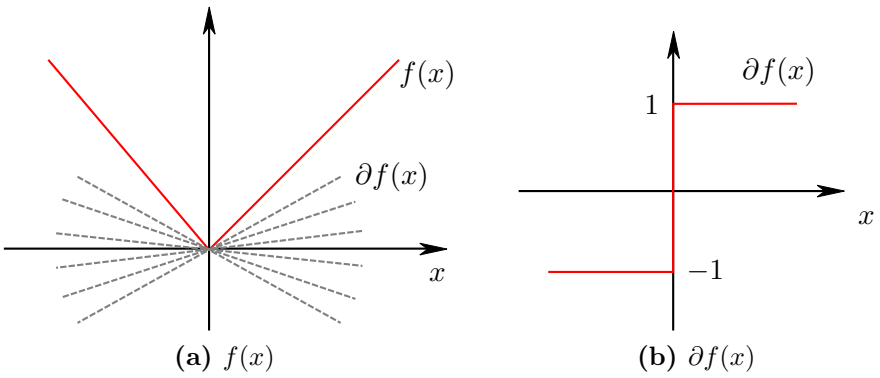
$$f(\mathbf{x}) \geq f(x_0) + \langle \mathbf{x} - \mathbf{x}_0, g \rangle. \tag{3.49}$$

The set of all subgradients of f at $\mathbf{x}_0$ is called the *subdifferential* $\partial f(\mathbf{x}_0)$. Note that if the function f is differentiable at $\mathbf{x}$ the subgradient only contains the gradient: $\partial f(\mathbf{x}) = \{\nabla f(\mathbf{x})\}$.

For example the subdifferential of the absolute function $f(x) = |x|$ results in

$$\partial f(x) = \begin{cases} +1 & x > 0 \\ -1 & x < 0 \\ [-1, 1] & else, \end{cases} \tag{3.50}$$

as illustrated in Figure 3.12.



(a) $f(x)$ **(b)** $\partial f(x)$

Figure 3.12: Exemplar subdifferential of the absolute function $f(x) = |x|$.

3.3.5 Proximal Mapping

The proximal mapping, also known as the resolvent- or proximity-operator, is a method to restrict closed proper convex functions to a certain bound, as proposed in [96]. Let

$f\colon \mathbb{R}^n \mapsto \mathbb{R} \cup +\infty$ be a convex, proper lower-semicontinuous function, which means that

$$\text{epi}(f) = \{(\mathbf{x}, t) \in \mathbb{R}^n \times \mathbb{R}\colon f(\mathbf{x}) \leq t\} \tag{3.51}$$

is a nonempty closed convex set and

$$\text{dom}(f) = \{\mathbf{x} \in \mathbb{R}^n\colon f(\mathbf{x}) < +\infty\} \tag{3.52}$$

is the set of points for which f takes on finite values.

Hence, the proximal mapping is given by

$$\text{prox}_{\lambda f}(\mathbf{y}) = (I + \lambda \partial f)^{-1}(\mathbf{y}) \arg\min_x \left\{ f(\mathbf{x}) + \frac{1}{2\lambda}\|\mathbf{x} - \mathbf{y}\|_2^2 \right\}. \tag{3.53}$$

The proximal operator $\text{prox}_f\colon \mathbb{R}^n \mapsto \mathbb{R}^n$ can be interpreted as moving points inside the domain towards the minimum, and points outside the domain towards the boundary of the domain in direction of the minimum. The parameter λ controls the extent of the movement, as shown in [81]. This interpretation is depicted in Figure 3.13.

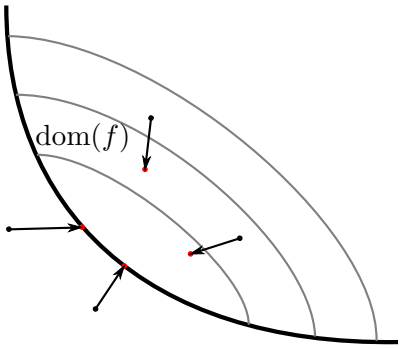


Figure 3.13: Proximal mapping. The proximal operator moves points outside of the domain on the boundary of the domain and points inside the domain towards the minimum (adopted from [81]).

3.4 Convex Optimization

The field of convex optimization is a subfield of mathematical optimization and studies the problem of finding a minimum in a convex function. We already briefly introduced in the form of *convex optimization problems* in Eqn. (3.33). The main advantage of convex problems over non-convex problems is that every local optimum must also be a global optimum. Hence, the global optimum can always be found, independent from the initialization. If we provide a reasonable amount of time to iterate, the accuracy of our model only depends on its design. Since variational models have been proven to be particularly useful to solve various imaging problems, we use them to solve the convex optimization

problems in our work. For a broader overview of we will refer to the works of Rockafellar [97, 98], Boyd [11], Chambolle and Pock [21], or Nesterov [74].

Let us consider the convex optimization problem of finding the global optimizer $\mathbf{x}^*$ such that

$$f(\mathbf{x}^*) = \min_{\mathbf{x}\in\mathbb{R}^n} f(\mathbf{x}), \qquad f\colon \mathbb{R}^n \mapsto \mathbb{R}, \tag{3.54}$$

where f is a real-valued convex and differentiable function. Where the gradient defined by $\nabla f(\mathbf{x})$ is Lipschitz continuous. For optimization the *optimality condition* $\nabla f(\mathbf{x}^*) = 0$ is a necessary and sufficient to define the solution. Although, in some cases the optimality can be directly calculated by setting the gradient to zero it is not possible for most large-scale imaging problems.

3.4.1 Gradient Descend

The basis of most optimization techniques and also the most simple optimization method is Gradient Descend (GD). It strictly minimizes the function according to the *optimality condition* $\nabla f(\mathbf{x}^*) = 0$ and optimizes along the gradient, with the restriction that f is convex and smooth (see Section 3.3.4). The simple *GD* update is given by

$$\mathbf{x}^{k+1} = \mathbf{x}^k - \tau \nabla f(\mathbf{x}^k), \tag{3.55}$$

for all iterations $k \geq 0$ and any $\mathbf{x}^0 \in \mathbb{R}^n$. The variable τ is the step size of the gradient update. It is chosen such that the function $f(\mathbf{x})$ is minimized until convergence. The convergence can be defined by e.g. a fixed number of iterations k or until f reaches a stationary point. In this context the measured speed (in number of iterations) in which the convergent sequence approaches its limits is called the *convergence rate*.

There are multiple optimization techniques that improve the convergence rate of *GD* by modifying the step size or adding terms to the gradient update. A famous extension of the basic *GD* is the *heavy-ball* method proposed by Polyak [86]. The basic update rule is given by

$$\mathbf{x}^{k+1} = \mathbf{x}^k - \tau \nabla f(\mathbf{x}^k) + \alpha \left(\mathbf{x}^k - \mathbf{x}^{k-1}\right), \qquad \text{for } \mathbf{x}^0 \in \mathbb{R}^n,\ \mathbf{x}^{-1} = \mathbf{x}^0, \tag{3.56}$$

for all $k \geq 0$, where the additional momentum term is inspired by the friction of a body motion in a potential field.

3.4.2 Proximal Gradient

In large scale optimization problems especially in computer vision the optimization has a more complex form such that the function may not be continuously differentiable in every

point. These unconstrained optimization problems often have the form

$$\min_{\mathbf{x}\in\mathbb{R}^n} f(\mathbf{x}) = g(\mathbf{x}) + h(\mathbf{x}), \tag{3.57}$$

where the (sub-)function $g(\mathbf{x})\colon \mathbb{R}^n \mapsto \mathbb{R}$ is convex and continuously differentiable and $h(\mathbf{x})\colon \mathbb{R}^n \mapsto \mathbb{R} \cup +\infty$ is convex but non-smooth. In this case the gradient update is calculated by the proximal mapping of $h(\mathbf{x})$, as shown in Section 3.3.5.

Starting from $\mathbf{x}^0 \in \mathbb{R}^n$ the iteration step of the proximal gradient method is then given by

$$\mathbf{x}^{k+1} = \text{prox}_{\lambda h}\left(\mathbf{x}^k - \tau\nabla g(\mathbf{x}^k)\right) \tag{3.58}$$

for $k \geq 0$ and $\tau > 0$. In the case where $h \equiv 0$ the algorithm results in standard *GD*. Similar to variants of *GD* there exist a lot of variants for accelerations or generalizations to the proximal gradient method, such as the FISTA (Fast Iterative Shrinkage Thresholding Algorithm) [6], the Douglas-Rachford Splitting or Alternating Direction Method of Multipliers (ADMM) algorithm [63].

3.4.3 Primal-Dual Algorithm

Variational energy minimization problems in computer vision often have non-differentiable parts and often the whole functional is non-differentiable. Variational problems often have the form of

$$\min_{\mathbf{x}\in\mathbb{R}^n} f(\mathbf{x}) = g(\mathbf{K}\mathbf{x}) + h(\mathbf{x}), \tag{3.59}$$

where g and h are convex but possibly non-smooth functions and $\mathbf{K}$ is a linear operator (i.e. $\mathbf{K} \in \mathbb{R}^{m\times n}$). A very prominent example of this problem is the ROF energy functional used for e.g. image denoising as proposed by Rudin et al. [99]. The energy functional has the form

$$\min_{\mathbf{x}\in\mathbb{R}} \|\nabla\mathbf{x}\|_1 + \frac{\lambda}{2}\|\mathbf{x} - \mathbf{d}\|_2^2, \tag{3.60}$$

where $\mathbf{d}$ is the noisy input and $\mathbf{x}$ is the optimizer. Chambolle [20] proposed an optimization algorithm to solve this functional transforming the non-smooth Total Variation (TV) into the dual formulation (see Section 3.3.3). Generalizations of the *primal-dual* formulation where proposed by Chambolle and Pock [21] and Esser et al. [26], where an optimal convergence rate in terms of Nesterov [74] has been shown.

Considering the class of non-smooth optimization problems

$$\min_{\mathbf{x}\in\mathbb{R}^n} g(\mathbf{K}\mathbf{x}) + h(\mathbf{x}), \tag{3.61}$$

where g and h are non-smooth functions but convex function. When substituting $\mathbf{Kx} = \mathbf{z}$ in Eqn. (3.61) our problem is defined as

$$\min_{\mathbf{x},\mathbf{z}} g(\mathbf{z}) + h(\mathbf{x}), \quad \text{s.t.} \quad \mathbf{Kx} - \mathbf{z} = 0. \tag{3.62}$$

Transferred into the Lagrange dual the above function is given by

$$\inf_{\mathbf{x},\mathbf{z}} g(\mathbf{z}) + h(\mathbf{x}) + \left\langle \mathbf{K}^{\mathrm{T}}\mathbf{y}, \mathbf{x} \right\rangle - \langle \mathbf{y}, \mathbf{z} \rangle \tag{3.63}$$

$$= - \sup_{\mathbf{z}} \left\{ \langle \mathbf{y}, \mathbf{z} \rangle - g(\mathbf{z}) \right\} - \sup_{\mathbf{x}} \left\{ \left\langle \mathbf{x}, -\mathbf{K}^{\mathrm{T}}\mathbf{y} \right\rangle - h(\mathbf{x}) \right\}. \tag{3.64}$$

Hence, the respective dual problem of Eqn. (3.61) is given by

$$\max_{\mathbf{y}\in\mathbb{R}^m} -g^*(\mathbf{y}) - h^*(-\mathbf{K}^T\mathbf{y}), \tag{3.65}$$

where g^* and $f*$ are the convex conjugate functions of g and f. This dual problem has the same optimal values as the primal problem for $\mathbf{x}$ and $\mathbf{y}$. This *strong duality* is given by holds if

$$\min_{\mathbf{x}\in\mathbb{R}^n} g(\mathbf{Kx}) + h(\mathbf{x}) = \max_{\mathbf{y}\in\mathbb{R}^m} -g^*(\mathbf{y}) - h^*(-\mathbf{K}^T\mathbf{y}). \tag{3.66}$$

An optimal value for $\mathbf{x}$ and $\mathbf{y}$ of this primal-dual formulation exists iff it is a solution of the transferred convex-concave saddle-point problem (primal-dual problem)

$$\min_{\mathbf{x}\in\mathbb{R}^n} \max_{\mathbf{y}\in\mathbb{R}^m} \langle \mathbf{Kx}, \mathbf{y} \rangle + h(\mathbf{x}) - g^*(\mathbf{y}) \tag{3.67}$$

exists.

The primal dual algorithm aims to iteratively maximize the primal-dual problem given in Eqn. (3.67) w.r.t the dual variable $\mathbf{y}$ (gradient ascend) and the minimize the problem w.r.t the primal variable $\mathbf{x}$ (gradient descend). Using the proximal mapping (Section 3.3.5) on g^* and h as used in previous algorithms the update scheme of the primal-dual algorithm is given by

$$\begin{aligned} \mathbf{y}^{k+1} &= \mathrm{prox}_{\sigma g^*}\left(\mathbf{y}^k + \sigma \mathbf{K}\bar{\mathbf{x}}^k\right) \\ \mathbf{x}^{k+1} &= \mathrm{prox}_{\tau h}\left(\mathbf{x}^k - \tau \mathbf{K}^T\mathbf{y}^{k+1}\right) \\ \bar{\mathbf{x}}^{k+1} &= \mathbf{x}^{k+1} + \Theta\left(\mathbf{x}^{k+1} - \mathbf{x}^k\right) \end{aligned} \tag{3.68}$$

The last part in the update scheme is an overrelaxation step added by Chambolle [20] proves that a guaranteed convergence exists for $\Theta = 1$. The step sizes $\tau > 0$ and $\sigma > 0$ are chosen such that the Lipschitz-continuity $\tau\sigma L^2 \leq 1$, with $L = \|\mathbf{K}\|$ is satisfied. As can be seen from the algorithm, the linear operator $\mathbf{K}$ must have a simple structure such

that L can be computed in reasonable time and the proximal mapping on g^* and h has a closed form representation such that a unique minimizer can be computed.

An improvement in terms of convergence speed of the primal-dual algorithm was proposed by Pock and Chambolle [84] where the step sizes τ and σ are defined through diagonal preconditioners. This diagonal preconditioners used as point-wise step sizes are defined as two symmetric and the positive definite matrices $\mathbf{T}$ and $\mathbf{\Sigma}$, such that $\|\mathbf{\Sigma}^{1/2}\mathbf{K}T^{1/2}\| \leq 1$ holds. In [84] it is shown that that the diagonal preconditioners $\mathbf{T} = \text{diag}(\tau_1, \ldots, \tau_n)$ and $\mathbf{\Sigma} = \text{diag}(\sigma_1, \ldots, \sigma_m)$ can be calculated as

$$\tau_j = \frac{1}{\sum_{i=1}^{m} |A_{i,j}|^{2-\alpha}} \quad \text{and} \quad \sigma_i = \frac{1}{\sum_{j=1}^{n} |A_{i,j}|^{\alpha}}, \quad \forall \alpha \in [0, 1]. \tag{3.69}$$

Because of its general applicability the primal-dual optimization scheme is used in many imaging problems, e.g. denoising, superresolution, stereo reconstruction, optical flow estimation or segmentation. The paper by Chambolle and Pock [21] gives a good overview of the different models and applications in computer vision.

4 Depth Superresolution

Contents

Modern Time of Flight (ToF) sensors often suffer from a low lateral resolution and high acquisition noise. In this chapter we show different optimization techniques to increase the resolution and quality of depth measurements by combining them with other sensor modalities and data. Hence, we propose different approaches for depth image superresolution for single *ToF* depth acquisitions, combined with an intensity sensors or by including a learned model into the optimization.

4.1 Introduction

In recent years, once prohibitively expensive range sensors reached their way to the mass market with the introduction of Microsoft Kinect, ASUS Xtion Pro or the Creative Senz3D camera. These cameras can now capture scene depth in real time and enable a variety of different applications in computer vision including 3D reconstruction, pose estimation or driver assistance. As described above this Time of Flight (ToF) sensors are measuring the depth actively through the runtime of light. The depth measurement is therefore independent from scene texture and largely independent from environmental lighting conditions. It delivers a dense depth map even at very close ranges [56, 103]. No additional calculations are necessary, which results in depth measurements at high frame rates. Its small packet size and low energy consumption makes these camera generally applicable even in mobile devices. But this compact design does not come without any quality restrictions and acquisitions remain afflicted by less than ideal attributes. Most of these sensors reached a natural upper limit on the spatial resolution of the sensor and also the precision of each depth sample decreases with the energy consumption. Henceforth, the two main disadvantages of the *ToF* measurement principle are the low lateral resolution and the high acquisition noise. This noise is composed of systematic parts, non-systematic parts and gross outliers. The main systematic errors are caused by different object reflectance. A lower object reflectance results in a depth offset. The non-systematic errors occur due to measurement inaccuracies depending on the signal-to-noise ratio of the reflected light. These inaccuracies result in random noise with zero mean. Outliers occur when the region that is acquired by one pixel contains large depth discontinuities, e.g. foreground and background. This error source is commonly known as the mixed pixel problem. An analysis of all these errors and their compensation can be found in [34, 38, 62, 68].

In this chapter the goal is to develop different approaches to increase and lateral resolution of *ToF* sensors while reducing noise and outliers. Hence our work combines increase spatial resolution (denoising and outlier removal) and the superresolution in depth image space (increase the lateral resolution). In our work we use a combination of different modalities or a learned model to achieve this goal. The different combinations are formulated as convex optimization problem, where the optimizer in our model is the resulting high resolution depth image. In particular we propose different sensor fusion models which are generally applicable to all of the following superresolution frameworks. Since the superresolution in our work is calculated by a fusion with other modalities the frameworks can my roughly divided by the used modalities. First, we will show the general variational superresolution (SR) framework. In this section we introduce the *SR* problem and show how we formulate this problem as a convex energy minimization using higher order regularization. Second, we show how the optimization of a superresolved depth image can be calculated by the low resolution (LR) *ToF* depth aided by an additional intensity image with higher resolution, as presented in Section 4.4. Third, we propose a single depth image superresolution where we combine the *ToF* image with a learned model of

low and high resolution dictionaries in Section 4.5. In the last section in this chapter we will show an extensive qualitative and quantitative evaluation of all the above approaches in Section 4.6.

4.2 Related Work

The field of image *SR* is a widely researched area in computer vision. The general problem of image *SR* is to find a high resolution image $\mathbf{I}_H$ out of a low resolution input image l. Hence, the image formation process is defined as

$$\mathbf{I}_L = f(\mathbf{I}_H) + \mathbf{v}, \tag{4.1}$$

where $\mathbf{v}$ defines the noise inherit during the acquisition process. The function $f(.)$ defines the images formation from the low resolution space into high resolution space. This includes the downsampling, the blur operator and the mapping.

The research on depth image *SR* mostly originates from intensity image *SR*. For depth image *SR* the main difference is the importance of denoising in the upsampling process. While the noise level of intensity images is low the amount of noise and outliers for depth sensing is much higher as explained above. The main focus of most approaches is to increase a depth measurement resolution and accuracy aided by other information sources. This additional information comes from multiple acquisitions, additional sensors or trained models. Hence, the main principle can be divided into: (1) the low level depth error characterization and calibration of the *ToF* sensor (2) the temporal and spacial fusion of multiple acquisitions (3) depth *SR* aided by high resolution intensity images and (4) the large field of *SR* of single depth images, namely single image superresolution (SISR). In this section we propose a general overview of these principles and show the work related to each field.

Sensor Calibration

A common low-level approach is to exactly investigate and calibrate each error source. The basic calibration of depth measurements aims to find an offset model for the individual sensors and error sources. In the image formation process shown in Eqn. (4.1) the calibration approaches aim to improve the accuracy of input image $\mathbf{I}_L$ and model the acquisition noise $\mathbf{v}$.

For the Kinect stereo sensor this is shown in [18, 45, 108], where the goal is to find the best possible model or parameters to calculate a real depth out of the disparity values from the sensor. For *ToF* measurements the depth calibration faces different problems. Due to the various error sources of the *ToF* measurement principle a constant offset calibration is not sufficient. In literature most methods calibrate the depth error by fitting non-linear functions or defining look-up tables that relate the measured depth at each pixel to the

corrected depth value.

Kahlmann et al. [53] proposed to acquire look-up tables for the distance related error at different integration times. Based this work Lindner and Kolb [61] calibrated the distance related error of a *ToF* camera with B-Splines. In [62] they extended their work by a intensity related component based on the surface reflectivity. Fuchs and Hirzinger [35] modeled the distance related error by a third order polynomial function which is directly calibrated with the camera intrinsics.

Belhedi et al. [7] propose a non-parametric method to compensate for the depth errors. In contrast to previously presented methods no underlying model is assumed, but the measurement volume in front of the camera is discretized and a depth offset is estimated for each voxel in the calibration step. Since not all voxels can be hit during calibration, a regularization term fills in the missing values. The authors only utilize the depth as input feature, in contrast our method also makes use of the intensity image and does not require a volumetric representation and thus no discretization of the world.

It is shown in all these works that the *ToF* offset is can not be directly separated a distinctive source. Hence the distance, intensity and integration related errors are correlated. Reynolds et al. [95] proposed a method to quantify the confidence of a *ToF* measurement. Instead of modeling each error as a separate mapping they propose to train non-linear mapping from depth and intensity features to a measurement confidence. Hence, a Random Forest (RF) [14] is trained using ground truth depth together with acquired depth and intensity features. As output it delivers a per-pixel-confidence according to each measurement. In our work [30] we go one step further and directly model the depth error in a Alternating Regression Forest (ARF) [104] setting. Instead of the pixel wise input features the approach uses patchwise information around each center pixel including the first and second order derivatives and the position in the image. An analysis of all measurement related errors of *ToF* acquisitions is shown in [38, 62, 91].

Multiple Depth Image Superresolution

A common way to improve the resolution and quality of depth information is to fuse multiple depth measurements from different viewpoints into one depth map of higher resolution and quality. In input measurements of a scene are acquired either while the depth camera is moved or by multiple 3D sensors. Relating to the image formation from Eqn. (4.1) these methods aim to estimate one super-resolved image $\mathbf{I}_H$ from multiple low resolution measurements $\mathbf{I}_L^k$. Furthermore, the mapping in the formation function $f(.)$ and the noise $\mathbf{v}$ is estimated.

Schuon et al. [105] proposed a method to fuse *ToF* acquisitions of slightly moved viewpoints, where the movement between the images is known. In this work a Markov Random Field (MRF) optimization framework is used with a bilateral regularization term. The method further incorporates the *ToF* sensor characteristics to increase the input quality. Based on this work, Schuon et al. [106] used a set of fused depth maps with larger

displacements, where additionally to the *SR* result the camera movement of the individual scans is estimated. In Cui et al. [24] this method was extended by scanning complete 3D shapes.

Newcombe et al. [76] proposed a method to create a dense surface model of a scene by integrating depth measurements over time from multiple viewpoints. In this approach the first depth scan is used to initialize a 3D volume. For each consecutive measurement the camera pose is tracked as the sensor is moved. According to this pose the new scan is integrated into the 3D volume. In [75] they extended this method for dynamic scenes. In this approach the scene geometry is reconstructed while simultaneously a dense volumetric 6D motion field is estimated. This motion field is used to warp the estimated geometry into the actual acquisition to increase the quality of the dense reconstruction.

Recent works addressed the fusion of different depth sensing techniques to increase resolution and quality. Gudmundsson et al. [37] presented a method for stereo and *ToF* depth map fusion in a dynamic programming approach. Similar work has been proposed by Zhu et al. [140] using an accurate depth calibration and fusing the measurements in a *MRF* framework. Additionally to this spatial fusion also a temporal fusion was performed by measuring the frame-to-frame displacement acquired with high speed intensity cameras. In the work of Ferstl et al. [27] a framework is presented to fuse the acquisitions of different depth senors from different viewing directions in a convex energy minimization framework.

Image Guided Depth Superresolution

With the new generation of range sensors that combine the capture of low resolution depth images with the acquisition of high resolution intensity images this class of approaches uses this additional intensity information as depth cue for image upsampling. In the estimation of a super-resolved output the goal is to use intensity information to infer information for the denoising of the solution without losing fine details. The intuition behind this approaches is that depth discontinuities in a scene often co-occur with color or brightness changes. Hence, in the upsampling process this textural information is used to guide the denoising process, where textural edges are lesser denoised than homogeneously textured regions.

Among the first works which exploited this possibility was presented by Diebel and Thrun [25]. In their work they have show an upsampling using a *MRF* formulation. In their model the unary term measures the quadratic distance between the high resolution estimate and the low resolution input where depth data exists. The binary term is formulated as the quadratic distance between neighboring pixels in the solution, weighted according to texture derivatives. In the work of Kopf et al. [55] a depth image interpolation technique was proposed leveraging a joint bilateral filter [115]. Similar Yang et al. [133] proposed a *SR* approach as post-processing for stereo reconstructions. They use a joint bilateral filtering of a depth cost volume and a RGB image in an iterative refinement process. Chan et al. [22] used a noise aware joint bilateral filter to increase the resolution

and to reduce depth map errors at multiple frames per second. In their method a term is introduced to blend between the results of standard upsampling and joint bilateral filtering depending on the regional statistics of the depth map.

Based on the seminal work of Diebel and Thrun [25] a successful extension was proposed by Park et al. [82]. The weighted in the *MRF* binary term is formulated as a combination of low level segmentation, image gradients, edge saliency and non-local means for depth upsampling. Similar, this combination of intensity and depth data for non-local regularization was proposed by Li et al. [60] for optimization of Bayesian Framework.

Single Image Superresolution

Typically, *SR* approaches based on dictionary learning build upon sparse coding [79]. Yang et al. [132] used the background of sparse coding to reconstruct high resolution test patches as sparse linear combination of atoms from a learned dictionary of paired high and low resolution training patches. Zeyde et al. [137] build upon this framework and improve the quality by adding several modifications. For training they use a combination of *K-SVD* [2] and Orthogonal Matching Pursuit (OMP) [116] for the low resolution dictionary and a direct regression of the high resolution dictionary using the pseudo-inverse. In the sparse coding approach of Mandal et al. [67] they additional penalized the input and output gradient in each low resolution patch during sparse optimization. Very recently, Timofte et al. [114] accelerated the inference of sparse coding by relaxing the ℓ_0 regularization with ℓ_2 regularization and replacing the single dictionary with many smaller sub-dictionaries which are pre-calculated. Hence, finding the sparse representation becomes a quadratic problem for each sub-dictionary which can be solved in closed form.

Other works use a dictionary of sample patches in a multi-class labeling problem in a *MRF*. In the work of Freeman and Liu [32] the goal is to minimize the difference of the set of high resolution dictionary atoms to the low resolution input, where the label being optimized represents the high resolution patch. Additionally, the overlap between neighboring patches is penalized in a binary term. Similar, Aodha et al. [66] proposed a *MRF* framework especially focused on depth image *SR* with higher noise. In their work an additional depth normalization is proposed to penalize the patch overlap. In a post-processing step they use a novel noise-removal algorithm to increase the quality. Instead of using a dictionary from an external database, Hornáček et al. [50] proposed a similar method where the low and high resolution patch-pairs of arbitrary size are searched in the image itself.

Most methods where the low resolution patches are reconstructed by a combination of dictionary entries highly suffer from input noise as reported in previous works. But there is also a great number of *SR* approaches that rely on a more general prior, as shown in [73]. Most related to our approach is the variational *SR* which is based on a known Point Spread Function (PSF) or blur-kernel. Mitzel et al. [70] used this model together with a Total Variation (TV) regularization and optical flow estimation for the image *SR*

of multiple image. This work was extended by Unger et al. [117] proposing a more robust model using the Huber-Norm. In [134] the *TV* regularization is weighted with an adaptive spatial algorithm based the scene curvature.

Discussion

Our work on depth image *SR* can be mostly related to the field of single image superresolution and image guided depth superresolution. In our work we define the basic superresolution problem in a variational framework where the blur kernel (*PSF*) is approximately known. Building on this formulation we show different methods to increase the quality by an intensity image guidance or by combining it with a learning approach based on sparse coding. Compared to previous works, this gives us the possibility to successfully superresolve depth images with higher amount of noise, where only an approximate blur-kernel is set. Since most man-made environments can be well represented with planar surfaces, we use a Total Generalized Variation (TGV) for regularization which aids the optimization to reconstruct piecewise planar surfaces. This helps to improve on both approaches based on learned dictionaries and on variational *SR* methods using a known blur-kernel.

4.3 Superresolution Problem

In this section we show the general *SR* problem which is used throughout all our estimations. The standard *SR* problem [69, 117] is to recover a high resolution depth map $\mathbf{I}_H^D \in \mathbb{R}^{\Omega_H} \mapsto \mathbb{R}$ out of a low resolution and noisy depth map $\mathbf{I}_L^D \in \mathbb{R}^{\Omega_H} \mapsto \mathbb{R}$, where Ω_H and Ω_L denote the high and low resolution image space. Note that if we only want to increase the spacial resolution (denoising) the high and the low resolution image spaces are equal.

In our optimization we will rely on the traditional *SR reconstruction constraint* [134] for intensity images: An observed low resolution image $\mathbf{I}_L^D$ is a blurred and down-sampled version of the noisy high resolution image $\mathbf{I}_H^D$:

$$\mathbf{I}_L^D = \mathbf{DBI}_H^D + \mathbf{v}, \tag{4.2}$$

where $\mathbf{D}$ represents the downsampling operator and $\mathbf{B}$ the blur filter. It is assumed that $\mathbf{D}$ performs a decimation by a fixed factor and $\mathbf{B}$, representing the blur-kernel, applies a low-pass filter to the image based on the downsampling operator. The additional variable $\mathbf{v}$ denotes an unknown amount of noise on the low resolution image. This reconstruction constraint holds for both intensity and depth images. The *SR* remains extremely ill-posed since for a given low resolution input $\mathbf{I}_L^D$, infinitely many high resolution images $\mathbf{I}_H^D$ satisfy the *reconstruction constraint* even if the blur kernel is exactly known.

To solve the ill-posed *reconstruction constraint* from Eqn. (4.2) we formulate it as a convex minimization problem. The fundamental estimation of the *SR* problem as shown

in [69, 117] is formulated as a variational problem defined by

$$\mathbf{u} = \arg\min_{\mathbf{u}} \left\{ \mathcal{D}(\mathbf{u}, \mathbf{I}_L^D) + \lambda \mathcal{R}(\mathbf{u}) \right\}, \tag{4.3}$$

based on the general variational model as introduced in Section 3.4. This model basically consists of a data term $\mathcal{D}(.)$ which measures the fidelity of the optimizer $\mathbf{u}$ from the low resolution measurements $\mathbf{I}_L^D$ and the regularization term $\mathcal{R}(.)$ that reflects prior knowledge of the smoothness of our solution. In the convex model both $\mathcal{D}(.)$ and $\mathcal{R}(.)$ are convex lower semi-continuous functions. The scalar value $\lambda \in \mathbb{R}$ is used to balance the relative weight between both terms.

The data term in our energy model is designed to ensure the data consistency to the base depth measurements $\mathbf{I}_L^D$. Origin from the reconstruction constraint defined in Eqn. (4.2) the data term in our optimization problem is formulated as

$$\mathcal{D}(\mathbf{u}, \mathbf{I}_L^D) = \int_{\Omega_H} |\mathbf{DBu} - \mathbf{I}_L^D|_Y \mathrm{d}\mathbf{x}, \tag{4.4}$$

where Y defines the normed vector space. In literature the space is defined either as the ℓ_1- or the ℓ_2-norm depending on the modality of the input data. While the ℓ_2 norm is optimal for Gaussian noise in the input fails at gross outliers. Contrary the ℓ_1-norm is better suited at outliers but has a worse performance in the presence of Gaussian noise. An common alternative is the Huber-norm which defines a combination by penalizing with ℓ_2 for lower values and ℓ_1 for higher values, as defined in Section 3.1. The Huber parameter $\varepsilon \in \mathbb{R}$ denotes the tradeoff between the ℓ_1 and the ℓ_2 norm in the penalization. Hence, the data term gets more robust against Gaussian noise as well as gross outliers in the input depth. Note that if $\varepsilon = 0$ this equals to the ℓ_1-norm and if $\varepsilon = \infty$ is the ℓ_2-norm. A definition of these normed vector spaces can be found in Section 3.1.

Given a fixed scalar upsampling factor $s \in \mathbb{R}$ the linear downsampling operator $\mathbf{D}\colon \mathbb{R}^{\Omega_H} \to \mathbb{R}^{\Omega_L}$ is defined by calculating the mean of a pixel region $s \times s$. The formation of one pixel $\mathbf{I}_L^D$ at position $\mathbf{x}$ is calculated as

$$\mathbf{I}_L^D(\mathbf{x}) = \frac{1}{s^2} \int_{\mathbf{y} \in \mathcal{N}(\mathbf{x})} \mathbf{I}_H^D(\mathbf{y}) \mathrm{d}\mathbf{y}. \tag{4.5}$$

For a regular grid mapping between low and high resolution image space the pixel neighborhood $\mathcal{N}$ is defined by

$$\mathcal{N}(\mathbf{x}) = (xs, ys) + \left[-\frac{s}{2}, \frac{s}{2} \right]^2, \quad \text{for} \quad \mathbf{x} = (x, y)^{\mathrm{T}}. \tag{4.6}$$

The quality of traditional *SR* methods rely on the quality of the blur-kernel, as shown in [69]. In our work we aim to present a more general algorithm where the blur-kernel can be freely defined according to the input modality. A general definition of the blurring

operator $\mathbf{B}\colon \mathbb{R}^{\Omega_H} \rightarrow \mathbb{R}^{\Omega_H}$ is modeled by a Gaussian kernel with a standard deviation $\sigma = \frac{1}{4}\sqrt{s^2 - 1}$ and 3σ for the kernel size. Both linear operators are fixed and can be set in a pre-processing step.

In natural environments depth images have less fine-grained texture components compared to intensity images. Hence, the regularization term $R(.)$ has to meet the challenges of producing a high resolution depth map that smooths small gradients caused by kernel inaccuracies while preserving strong edges and planar surfaces. Most current regularization terms are based on the *TV*-norm [77] defined by

$$\mathcal{R}(\mathbf{u}) = \int_{\Omega_H} \|\nabla \mathbf{u}\|_1. \tag{4.7}$$

Due to the ℓ_1-norm on the depth image gradient the *TV* favors constant values in the solution. This property is nice for the denoising of intensity images but is problematic at depth image *SR* because it causes staircase artifacts in the solution. A easy and very common way to circumvent this problem is to use the Huber-norm for the gradient penalization defined by

$$\mathcal{R}(\mathbf{u}) = \int_{\Omega_H} \|\nabla \mathbf{u}\|_\varepsilon. \tag{4.8}$$

The Huber-norm of the gradient prevents staircasing and only slightly smooths sharp edges in the solution.

In our model we use a more general regularization namely the *TGV* [13] of second order. While the *TV* penalizes the first derivative of the optimizer, the *TGV* penalizes the derivative of arbitrary order. E.g. the *TGV* of second order penalizes the second derivative and hence, favors piecewise affine solutions instead of piecewise constant solution as in th *TV* case.

Generally, a model that is regularized by a *TGV* term of order k favors a solution that is piecewise composed of polynomials of order $k-1$. This general formulation of the *TGV* regularization term of order $k \geq 1$ is given by the dual definition

$$\mathrm{TGV}^k_\lambda = \sup \left\{ u \operatorname{div}^k v \mathrm{d}\mathbf{x} \mid v \in \mathcal{C}^k_c(\Omega, \mathrm{Sym}^k(\mathbb{R}^d)), \|\operatorname{div}^l v\|_\infty \leq \lambda_l, l = 0, \ldots, k-1 \right\}. \tag{4.9}$$

The variables $\lambda_0, \ldots, \lambda_{k-1} > 0$ are individual scalar weights for each order k. $\mathcal{C}^k_c(\Omega, \mathrm{Sym}^k(\mathbb{R}^d))$ denotes the space of tensors of order k.

The definition of the *TGV* at order $k = 1$ is given by

$$\mathrm{TGV}^1_\lambda = \lambda \sup \left\{ u \operatorname{div} v \mathrm{d}\mathbf{x} \mid v \in \mathcal{C}^1_c(\Omega, \mathrm{Sym}^1(\mathbb{R}^d)), \|v\|_\infty \leq 1 \right\}, \tag{4.10}$$

$$= \lambda \, \mathrm{TV}(u). \tag{4.11}$$

which directly defines the dual definition of the *TV*. For our *SR* estimation it turns out

that the second order *TGV* is an acceptable tradeoff between accuracy and computational complexity. It not only includes the first derivative but also the second order derivatives to approximate the superresolved depth surface by piecewise affine parts. The primal definition of the second order *TGV* is formulated as

$$\mathrm{TGV}_{\lambda}^{2} = \min_{\mathbf{u},\mathbf{v}} \left\{ \lambda_1 \int_{\Omega} \|\nabla \mathbf{u} - \mathbf{v}\|_1 \, \mathrm{d}x + \lambda_0 \int_{\Omega} \|\nabla \mathbf{v}\|_1 \, \mathrm{d}x \right\}, \tag{4.12}$$

where additional to the first order smoothness of the depth image $\mathbf{u}$, the auxiliary variable $\mathbf{v}$ is introduced to enforce second order smoothness. The scalars λ_0, $\lambda_1 \in \mathbb{R}_{\geq 0}$ are used to weight each order. Because the *TGV* regularizer is convex in $\mathbf{u}$ and $\mathbf{v}$ it allows to compute a globally optimal solution.

Compared to the standard *TV* the *TGV* of second order avoids the piecewise constancy in the solution and in contrast to the Huber-norm it directly optimizes for piecewise affine high resolution surface instead of an approximation. Hence, we directly can optimize for planar surfaces while compared to the Huber-norm we can still preserve sharp object boundaries.

The complete optimization model from Eqn. (4.3) with Huber penalization in the data term and *TGV* regularization which is defined as

$$\min_{\mathbf{u},\mathbf{v}} \left\{ \int_{\Omega_L} |\mathbf{D}\mathbf{B}\mathbf{u} - \mathbf{I}_L^D|_{\varepsilon} + \lambda_1 \int_{\Omega_H} \|\nabla \mathbf{u} - \mathbf{v}\|_1 \, \mathrm{d}x + \lambda_0 \int_{\Omega_H} \|\nabla \mathbf{v}\|_1 \, \mathrm{d}x \right\}, \tag{4.13}$$

where $\varepsilon \in \mathbb{R}_{\geq 0}$ defines the Huber parameter. In order to optimize this model we discretize the continuous image space into a regular Cartesian grid. The low and high resolution space is discretized by $\Omega_L \mapsto \mathbb{R}^{M_l \times N_l}$ and $\Omega_H \mapsto \mathbb{R}^{M \times N}$ respectively. Hence, the discrete optimization problem from Eqn. (4.13) is given by

$$\min_{\mathbf{u} \in \mathbb{R}^{MN}, \mathbf{v} \in \mathbb{R}^{2MN}} \left\{ \|\mathbf{D}\mathbf{B}\mathbf{u} - \mathbf{I}_L^D\|_{\varepsilon} + \lambda_1 \|\nabla \mathbf{u} - \mathbf{v}\|_1 + \lambda_0 \|\nabla \mathbf{v}\|_1 \right\}. \tag{4.14}$$

The proposed optimization problem (4.14) is convex but non smooth due to ℓ_1 terms in the *TGV* regularization and the Huber-norm in the data term (if $\varepsilon = 0$). Therefore the optimization of such problems is not a trivial task. To find a fast, global optimal solution for our problem we use the primal-dual energy minimization scheme, as shown in Section 3.4.3. After introducing Lagrange multipliers for the constraints and biconjugation using the Legendre-Fenchel transform (LF) we are able to reformulate the non-smooth problem in a convex-concave saddle-point problem. The optimization problem can be efficiently minimized through gradient ascend of the dual energy and gradient descend of the primal energy. The transformed saddle-point problem of our energy functional is given by

$$\min_{\mathbf{u},\mathbf{v}} \max_{\mathbf{q},\mathbf{p}_\mathbf{u},\mathbf{p}_\mathbf{v}} \left\{ \langle \mathbf{q}, \mathbf{D}\mathbf{B}\mathbf{u} - \mathbf{I}_L^D \rangle_Q - \frac{\varepsilon}{2} \|\mathbf{q}\|_2^2 + \lambda_1 \langle \mathbf{p}_\mathbf{u}, \nabla \mathbf{u} - \mathbf{v} \rangle_{P_\mathbf{u}} + \lambda_0 \langle \mathbf{p}_\mathbf{v}, \nabla \mathbf{v} \rangle_{P_\mathbf{v}} \right\}, \tag{4.15}$$

introducing the dual variables $\mathbf{q}$, $\mathbf{p_u}$ and $\mathbf{p_v}$. The feasible sets of these variables are defined by

$$P_{\mathbf{u}} = \left\{\mathbf{p_u}\colon \Omega_H \to \mathbb{R}^{2MN} |\ \|\mathbf{p_u}(\mathbf{x})\|_2 \leq 1\right\}, \tag{4.16}$$
$$P_{\mathbf{v}} = \left\{\mathbf{p_v}\colon \Omega_H \to \mathbb{R}^{4MN} |\ \|\mathbf{p_v}(\mathbf{x})\|_2 \leq 1\right\}, \tag{4.17}$$
$$Q = \left\{\mathbf{q}\colon \Omega_L \to \mathbb{R}^{M_L N_L} |\ -1 \leq \mathbf{q}(\mathbf{x}) \leq 1\right\}, \tag{4.18}$$
$$\forall \mathbf{x} \in \mathbb{R}^{M\times N}. \tag{4.19}$$

This formulation is used in the primal-dual algorithm, where the primal and dual variables are iteratively optimized for the individual pixels in three steps. First, the dual variables $\mathbf{q}$, $\mathbf{p_u}$ and $\mathbf{p_v}$ are updated using gradient ascend. Second, the primal variables are updated using gradient-descent. Third, the primal variables are refined in an overrelaxation step. The primal and dual variables are initialized with $\mathbf{u}^0, \mathbf{v}^0, \mathbf{q}, \mathbf{p_u}, \mathbf{p_v} = \mathbf{0}$. For any iteration $n \geq 0$ the primal-dual steps are calculated according to

$$\begin{cases} \mathbf{p_u}^{n+1} &= \mathcal{P}_{P_\mathbf{u}}\left\{\mathbf{p_u}^n + \sigma_\mathbf{u}\lambda_1\left(\nabla\bar{\mathbf{u}}^n - \bar{\mathbf{v}}^n\right)\right\}, \\ \mathbf{p_v}^{n+1} &= \mathcal{P}_{P_\mathbf{v}}\left\{\mathbf{p_v}^n + \sigma_\mathbf{t}\lambda_0\nabla\bar{\mathbf{v}}^n\right\}, \\ \mathbf{q}^{n+1} &= \mathcal{P}_Q\left\{\mathbf{q}^n + \sigma_\mathbf{q}\dfrac{\mathbf{DB}\bar{\mathbf{u}}^n - \mathbf{I}_L}{1+\sigma_\mathbf{q}\varepsilon}\right\}, \\ \mathbf{u}^{n+1} &= \mathbf{u}^n + \tau_\mathbf{u}\left(\lambda_1\nabla^\mathrm{T}\mathbf{p_u}^{n+1} - \mathbf{B}^\mathrm{T}\mathbf{D}^\mathrm{T}\mathbf{q}\right), \\ \mathbf{v}^{n+1} &= \mathbf{v}^n + \tau_\mathbf{v}\left(\lambda_0\nabla^\mathrm{T}\mathbf{p_v}^{n+1} + \lambda_1{\mathbf{p_u}}^{n+1}\right), \\ \bar{\mathbf{u}}^{n+1} &= \mathbf{u}^{n+1} + \theta(\mathbf{u}^{n+1} - \bar{\mathbf{u}}^n), \\ \bar{\mathbf{v}}^{n+1} &= \mathbf{v}^{n+1} + \theta(\mathbf{v}^{n+1} - \bar{\mathbf{v}}^n), \end{cases} \tag{4.20}$$

until a stopping criterion is reached. The operators $\mathcal{P}_Q$, $\mathcal{P}_{P_\mathbf{u}}$ and $\mathcal{P}_{P_\mathbf{v}}$ are point-wise Euclidean projection functions onto the previously defined feasible sets Q, $P_\mathbf{u}$ and $P_\mathbf{v}$ defined by

$$(\mathcal{P}_Q(\tilde{\mathbf{q}}))(\mathbf{x}) = \min\left\{1, \max\{-1, \tilde{\mathbf{q}}(\mathbf{x})\}\right\}, \tag{4.21}$$
$$(\mathcal{P}_{P_\mathbf{u}}(\tilde{\mathbf{p_u}}))(\mathbf{x}) = \left\{\frac{\tilde{\mathbf{p_u}}(\mathbf{x})}{\max\{1, \|\tilde{\mathbf{p_u}}(\mathbf{x})\|_2\}}\right\}, \tag{4.22}$$
$$(\mathcal{P}_{P_\mathbf{v}}(\tilde{\mathbf{p_v}}))(\mathbf{x}) = \left\{\frac{\tilde{\mathbf{p_v}}(\mathbf{x})}{\max\{1, \|\tilde{\mathbf{p_v}}(\mathbf{x})\|_2\}}\right\}, \tag{4.23}$$
$$\forall \mathbf{x} \in \mathbb{R}^{M\times N}. \tag{4.24}$$

The overrelaxation stepsize is set to $\theta = 1$ and the timesteps of the optimization are set by the pixel-wise Lipschitz boundaries determined by diagonal preconditioning as shown

in Section 3.4.3. Hence, the timesteps are given by

$$\sigma_{\mathbf{q}}(i) = \frac{\sigma}{\sum_j |(\mathbf{DB})(i,j)|}, \tag{4.25}$$

$$\sigma_{\mathbf{p_u}} = \frac{\sigma}{3\lambda_1} \tag{4.26}$$

$$\sigma_{\mathbf{p_v}} = \frac{\sigma}{2\lambda_0}, \tag{4.27}$$

$$\tau_{\mathbf{u}}(j) = \frac{\tau}{4\lambda_1 + \sum_i |(\mathbf{DB})(i,j)|}, \tag{4.28}$$

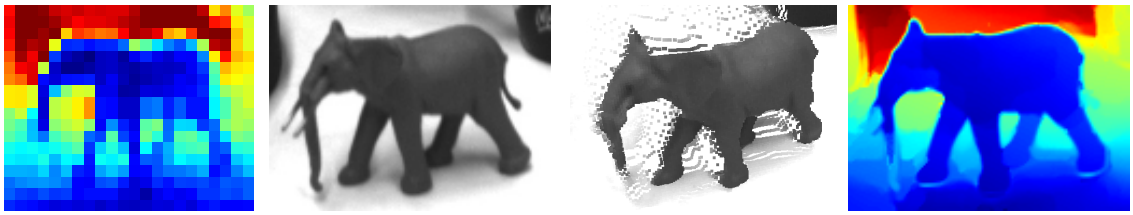
$$\tau_{\mathbf{v}} = \frac{\tau}{\lambda_1 + 4\lambda_0}, \tag{4.29}$$

$$\forall i,j \colon i = 1 \dots M_L N_L,\ j = 1 \dots MN, \tag{4.30}$$

where the initial timesteps are set such that $\sigma\tau \leq 1$. This scheme achieves a fast and guaranteed convergence to the global optimal solution for different linear downsampling and blurring operators. The gradient and divergence operators are approximated using forward/backward differences with Neumann and Dirichlet boundary conditions, respectively.

4.4 Image Guided Depth Superresolution

In this section we present a novel method for the *SR* of depth images in a *RGB-D* system. The goal of this method is to super-resolve a noisy and low resolution depth image $\mathbf{I}_L^D$ by using a high resolution intensity image $\mathbf{I}_H^I$ which guides the upsampling estimation, as shown in Figure 4.1. We use the optimization scheme proposed in the previous section which is iteratively optimized based on the primal-dual formulation.



(a) LR depth $\mathbf{I}_L^D$ **(b)** HR intensity $\mathbf{I}_H^I$ **(c)** HR depth upsampling result $\mathbf{I}_H^D$

Figure 4.1: Upsampling of a low resolution depth image (a) using an additional high resolution intensity image (b) through image guided anisotropic Total Generalized Variation (c). Depth maps are color coded for better visualization.

In a real world *RGB-D* system such as the Microsoft Kinect V1 and V2 as well as the Intel Senz3D the depth image $\mathbf{I}_L^D$ and the intensity image $\mathbf{I}_H^I$ are acquired by different sensors. Although the acquisition is time synchronized, the camera center and viewing

direction differ. A direct image to image fusion system is not possible and hence, a mapping of the images into one common coordinate system is required.

4.4.1 Image Mapping

In our sensor fusion system the coordinate system of $\mathbf{I}_H^I$ does not coincide with the coordinate system of $\mathbf{I}_L^D$. Hence, we propose to define a target coordinate system and map the data from the other systems into the target system in a preprocessing step. This mapping is calculated by the projection and back-projection of depth measurements, as defined in Section 3.2. The intrinsic and extrinsic calibration is calculated with the automatic camera calibration framework of Ferstl et al. [30]. Each depth measurement $\mathbf{I}_L^D(\mathbf{x})$ at the homogeneous image coordinate $\mathbf{x}^h = (x, y, 1)^{\mathrm{T}} \in \Omega_L$ is projected into the high resolution image space Ω_H. This projection is calculated as

$$\begin{aligned} \mathbf{X} &= \frac{(\mathbf{R}_L\mathbf{K}_L)^{-1}\mathbf{x}^h}{\|(\mathbf{R}_L\mathbf{K}_L)^{-1}\mathbf{x}^h\|_2}\mathbf{I}_L^D(\mathbf{x}) + \mathbf{C}_L \\ \tilde{\mathbf{x}}^h &= \mathbf{P}_H\mathbf{X}^h \quad \forall \mathbf{x} \in \Omega_L, \end{aligned} \tag{4.31}$$

where $\mathbf{R}_L$, $\mathbf{C}_L$ and $\mathbf{K}_L$ are the calibrated camera parameters in the low resolution space used to get a 3D point $\mathbf{X}$ the world coordinate system. The camera center and $\mathbf{X}$ the 3D point. Each 3D point is back projected by multiplication with the projection matrix of the intensity camera $\mathbf{P}_H$. Hence, we get a projected depth image $\mathbf{I}_S^D \in \mathbb{R}^{\Omega_H}$ consisting of a sparse set of base depth points at position $\tilde{\mathbf{x}}$ in the intensity image space Ω_H where the depth value is given by the distance to the 3D point $\mathbf{X}$ (see Figure 4.2).

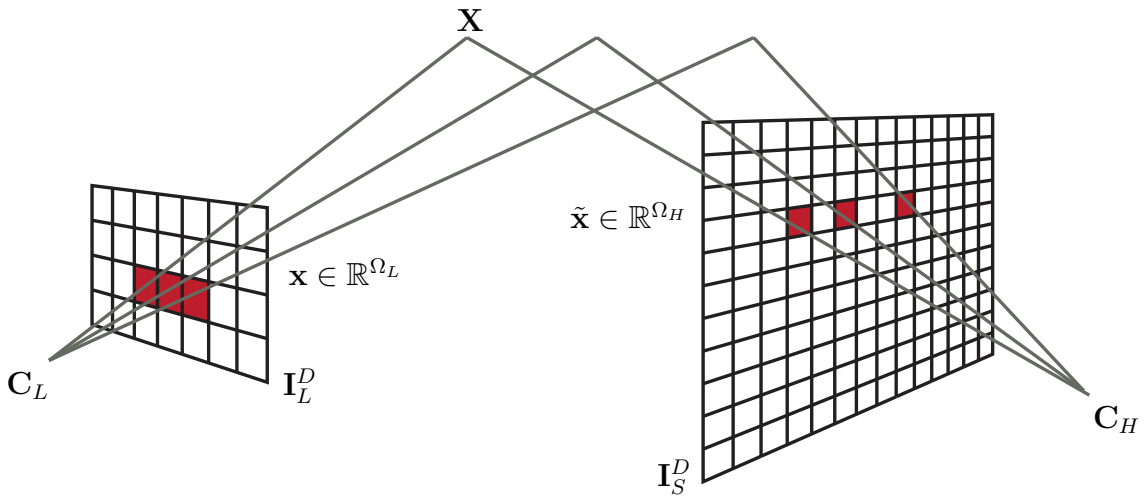


Figure 4.2: Projection from a low resolution depth map $\mathbf{I}_L^D$ to a high resolution sparse depth map $\mathbf{I}_S^D$ in the high resolution camera coordinate system.

Although, one low resolution sensor pixel $\mathbf{I}_L^D(\mathbf{x})$ measures the average depth of multiple pixels in the high resolution space we only project it to its central pixel $\tilde{\mathbf{x}}$. Therewith, we minimize the error which can occur due to this averaging in the high resolution space. In the superresolution reconstruction constraint defined in Eqn. (4.2) the high resolution (HR) solution is defined as a downsampled and blurred version of the *LR* input depth. In our *RGB-D* fusion model, the downsampling operator $\mathbf{D}$ is defined by the pixelwise backward projections from the *HR* image pixels $\tilde{\mathbf{x}}$ to the *LR* pixel positions $\mathbf{x}$. Hence, the downsampling operator $\mathbf{D} \in \mathbb{R}^{\Omega_L \times \Omega_H}$ is given by an irregular grid according to the projected points $\tilde{\mathbf{x}} \mapsto \mathbf{x}$, compared to the standard reconstruct constraint where the downsampling is defined on a regular grid. Hence, the sparse operator $\mathbf{D}$ is 1 at all pixel correspondences $(\mathbf{x}, \tilde{\mathbf{x}})$ and 0 everywhere else. In our optimization framework, the area in between the *LR* depth measurements is interpolated by the regularization force.

4.4.2 Intensity Image Guided Regularization

In our optimization model the high resolution intensity image is used to aid the optimization. Our model builds on the assumption that high depth continuities most likely correspond to texture edges in the intensity image. Hence, we calculate a tensor based on the intensity image gradients to produce a more accurate *SR* result. The tensor is used as a weighting term in the regularization term. There are different possibilities to formulate the weighting in the regularization term. Previous works use a pixel wise scalar weighting calculated from the gradient magnitude of the intensity image, as shown in [25] or a non-local operator, as shown in [82].

In our *SR* model we propose to use the anisotropic diffusion tensor $\mathbf{T}^{\frac{1}{2}}$ based on the Nagel-Enkerlmann operator [72] since it delivers the overall best results compared to other methods. This symmetric 2×2 tensor is defined by

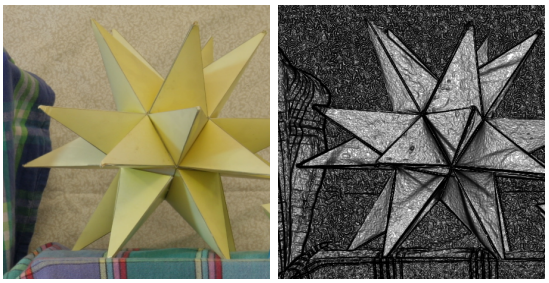
$$\mathbf{T}^{\frac{1}{2}} = \exp\left(-\beta \left|\nabla \mathbf{I}_H^I\right|^{\gamma}\right) nn^{\mathrm{T}} + n^{\perp} n^{\perp^{\mathrm{T}}}, \tag{4.32}$$

where n is the normalized direction of the image gradient $n = \frac{\nabla \mathbf{I}_H^I}{\|\nabla \mathbf{I}_H^I\|_2}$, $n^{\perp}$ is the normal vector to the gradient and the scalars β, γ adjust the magnitude and the sharpness of the tensor. The image gradients are calculated by the Sobel operator to decrease the influence of acquisition noise in the input intensity images. An exemplar tensor calculation is shown in Figure 4.3a.

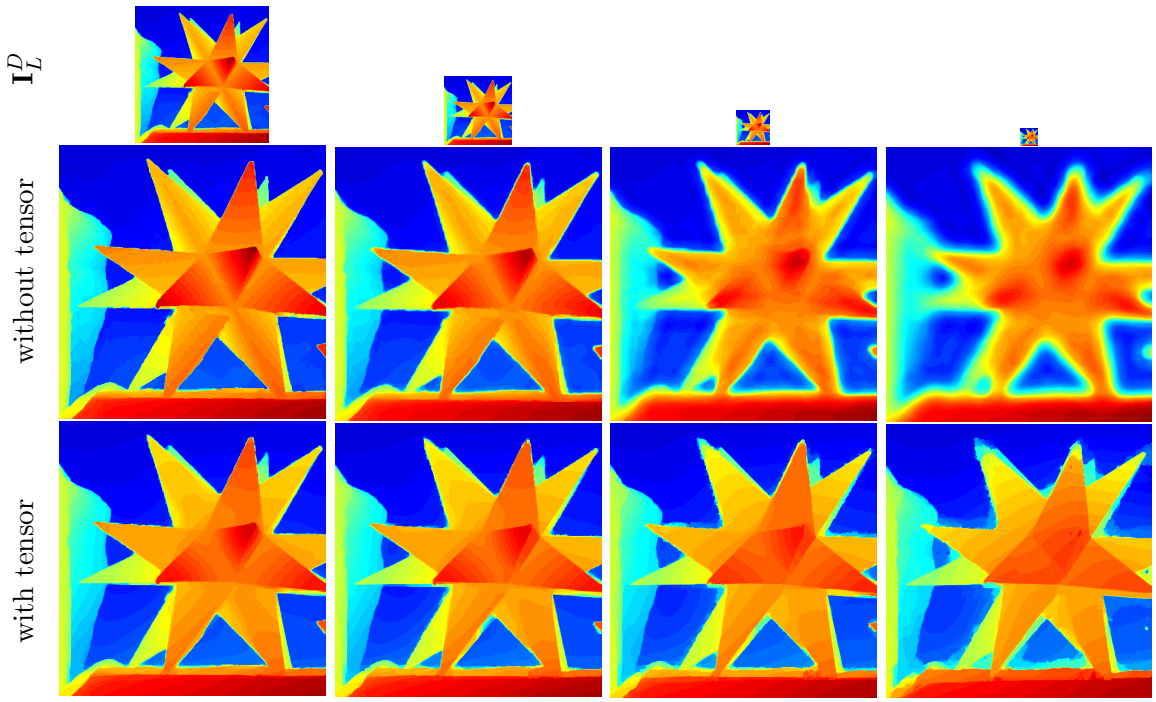
This anisotropic tensor is directly included as a weighting term in the convex regularization force. In our model we include the anisotropic tensor into *TGV* model. Included in the regularization term, the anisotropic diffusion tensor not only weights the first order depth gradient but also orients the gradient direction during the optimization process. Hence, the model is able to penalize high depth discontinuities at homogeneous regions and allow sharp depth edges at corresponding texture differences. With the additional anisotropic tensor information the optimization result leads to sharper and more defined

edges in our solution. Further, the regions where the depth data is interpolated are filled out more reasonably. In the complete regularization term, the anisotropic tensor is multiplied by the first order *TGV* term results in the anisotropic *TGV* regularization ($aTGV$) given by

$$\mathcal{R}_{\mathrm{aTGV}}(\mathbf{u},\mathbf{v}) = \lambda_1 \|\mathbf{T}^{\frac{1}{2}}(\nabla\mathbf{u} - \mathbf{v})\|_1 + \lambda_0 \|\nabla\mathbf{v}\|_1. \tag{4.33}$$



(a) Input intensity $\mathbf{I}_H^I$ and tensor $\mathbf{T}^{\frac{1}{2}}$



(b) ×2 **(c)** ×4 **(d)** ×8 **(e)** ×16

Figure 4.3: Superresolution Quality for different magnification factors with and without an anisotropic tensor $\mathbf{T}^{\frac{1}{2}}$ at the given input depth $\mathbf{I}_L^D$.

4.4.3 Numerical Optimization

The image guided anisotropic *TGV* is included into our superresolution problem shown in Eqn. (4.15). Hence, the complete convex-concave saddle-point problem is changed to

$$\min_{\mathbf{u},\mathbf{v}} \max_{\mathbf{q},\mathbf{p_u},\mathbf{p_v}} \left\{ \langle \mathbf{q}, \mathbf{DBu} - \mathbf{I}_L^D \rangle_Q - \frac{\varepsilon}{2}\|\mathbf{q}\|_2^2 + \lambda_1 \langle \mathbf{p_u}, \mathbf{T}^{\frac{1}{2}}(\nabla \mathbf{u} - \mathbf{v}) \rangle_{P_\mathbf{u}} + \lambda_0 \langle \mathbf{p_v}, \nabla \mathbf{v} \rangle_{P_\mathbf{v}} \right\}, \tag{4.34}$$

and, since $\mathbf{T}^{\frac{1}{2}} = \mathbf{T}^{\frac{1}{2}\mathrm{T}}$, the primal-dual optimization steps are calculated by

$$\begin{cases} \mathbf{p}_\mathbf{u}^{n+1} & = \mathcal{P}_{P_\mathbf{u}} \left\{ \mathbf{p}_\mathbf{u}^n + \sigma_\mathbf{u} \lambda_1 \left(\mathbf{T}^{\frac{1}{2}} (\nabla \bar{\mathbf{u}}^n - \bar{\mathbf{v}}^n) \right) \right\}, \\ \mathbf{p}_\mathbf{v}^{n+1} & = \mathcal{P}_{P_\mathbf{v}} \left\{ \mathbf{p}_\mathbf{v}^n + \sigma_\mathbf{t} \lambda_0 \nabla \bar{\mathbf{v}}^n \right\}, \\ \mathbf{q}^{n+1} & = \mathcal{P}_Q \left\{ \mathbf{q}^n + \sigma_\mathbf{q} \dfrac{\mathbf{DB}\bar{\mathbf{u}}^n - \mathbf{I}_L}{1 + \sigma_\mathbf{q}\varepsilon} \right\}, \\ \mathbf{u}^{n+1} & = \mathbf{u}^n + \tau_\mathbf{u} \left(\lambda_1 \nabla^\mathrm{T} \mathbf{T}^{\frac{1}{2}} \mathbf{p}_\mathbf{u}^{n+1} - \mathbf{B}^\mathrm{T} \mathbf{D}^\mathrm{T} \mathbf{q} \right), \\ \mathbf{v}^{n+1} & = \mathbf{v}^n + \tau_\mathbf{v} \left(\lambda_0 \nabla^\mathrm{T} \mathbf{p}_\mathbf{v}^{n+1} + \lambda_1 \mathbf{T}^{\frac{1}{2}} \mathbf{p}_\mathbf{u}^{n+1} \right), \\ \bar{\mathbf{u}}^{n+1} & = \mathbf{u}^{n+1} + \theta(\mathbf{u}^{n+1} - \bar{\mathbf{u}}^n), \\ \bar{\mathbf{v}}^{n+1} & = \mathbf{v}^{n+1} + \theta(\mathbf{v}^{n+1} - \bar{\mathbf{v}}^n), \end{cases} \tag{4.35}$$

until a stopping criterion is reached. Following the diagonal preconditioning from Section 3.4.3 the timesteps from Eqn. (4.40) are updated to

$$\sigma_{\mathbf{p_u}}(j) = \frac{\sigma}{\lambda_1 \sum_j (|(\mathbf{T}^{\frac{1}{2}} \nabla)(i,j)|)}, \quad \sigma_{\mathbf{p_v}} = \frac{\sigma}{2\lambda_0}, \tag{4.36}$$

$$\sigma_\mathbf{q}(i) = \frac{\sigma}{\sum_j |(\mathbf{DB})(i,j)|}, \tag{4.37}$$

$$\tau_\mathbf{u}(j) = \frac{\tau}{\lambda_1 \sum_i |(\mathbf{T}^{\frac{1}{2}} \nabla)(i,j)| + \sum_i |(\mathbf{DB})(i,j)|}, \tag{4.38}$$

$$\tau_\mathbf{v}(j) = \frac{\tau}{\lambda_1 \sum_i |\mathbf{T}^{\frac{1}{2}}(i,j)| + 4\lambda_0} \tag{4.39}$$

$$\forall i,j \colon i = 1 \dots M_L N_L, \quad j = 1 \dots MN, \tag{4.40}$$

due to the introduction of the anisotropic tensor into our model.

Compared to the standard model without the usage of an high resolution intensity image the accuracy of the *SR* result is drastically increased. This tensor guidance is especially useful at object edges where the simple model blurs the result. An exemplar comparison can be seen in Figure 4.3 where the *SR* result with and without anisotropic tensor is shown. It can be seen that the quality improvement of our image guided model increases with increasing magnification factors.

This fusion of *LR* depth with a *HR* intensity image has superior performance compared the estimation without the guidance. Since most depth cameras come equipped with an

additional intensity camera this method is very suitable for many applications. However, the exact calibration of the camera system is a very crucial part of this method. The quality drastically decreases with calibration inaccuracies and with increasing baseline due to possible occlusions.

4.5 Single Depth Image Superresolution

In this section we show a novel method for image guided depth superresolution without the need of an additional camera. Hence, we propose a method which combines both learning based single image *SR* with our classical *SR* problem based on a known blur kernel. This combination is used in a variational *SR* together with anisotropic higher order regularization, as shown in the previous section. The whole workflow of our model is depicted in Figure 4.4. Similar to depth *SR* approaches that use a high resolution intensity image for guidance we use a sparse coding approach to pre-calculate edge priors out of the low resolution example. The sparse code is reconstructed in a variational energy minimization using a learned dictionary from an external database of low and high resolution examples. In addition to the traditional sparsity constraint we minimize the overlap of neighboring patches in our optimization formulating a *variational sparse coding* approach. The spacial coherence in image space leads to more accurate edges than traditional averaging across the overlap. The edge priors are used in the image guided regularization of our image guided *SR* model from Section 4.4. Hence, our method has the advantage that we do not need an additional intensity camera. The high resolution guidance is reconstructed via the edge priors. Furthermore, since we use a variational energy model with *TGV* as regularization we are able to handle depth inputs with higher amounts of noise.

In order to estimate the *HR* guidance image we adapt ideas which were recently proposed in intensity image superresolution. Zeyde et al. [137] proposed a method for single intensity image *SR* using the relationship of low and high resolution images through a *sparsity constraint*:

For a given low resolution patch the goal is to find the best entry in a dictionary of sample patches collected from an external database of low and high resolution image pairs. Sparse coding approaches aim to overcome this search by using an overcomplete dictionary based on sparse signal representations. Given a learned low resolution dictionary $\mathbf{A}_l$ the goal is to find the sparse representation α such that the patch is optimally reconstructed by the dictionary entries:

$$\mathbf{p}_l = \mathbf{D}_L \alpha, \tag{4.41}$$

where $\mathbf{p}_l \in \mathbb{R}^n$ is the low resolution input patch of size $\sqrt{n} \times \sqrt{n}$. The resulting high resolution patch is found through $\mathbf{p}_h = \mathbf{D}_H \alpha$ using the corresponding high resolution dictionary $\mathbf{D}_H$.

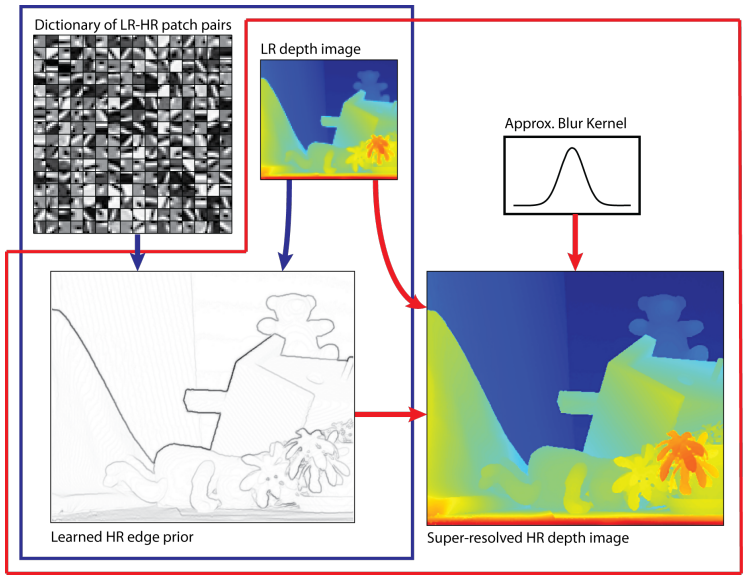


Figure 4.4: Single Depth Image Superresolution. Our method estimates strong edge priors from a given *LR* depth image and a learned dictionary using a novel sparse coding approach (blue part). The learned *HR* edge prior is used as anisotropic guidance in a novel variational *SR* using higher order regularization (red part).

4.5.1 Edge Prior Estimation

The goal of the this estimation is to find high resolution edge priors to guide the regularization in a variational superresolution. The estimation of the optimal patch priors for the depth regularization in our model is formulated as finding the best entry in a learned dictionary of sampled patches from low and high resolution image pairs using sparse coding. Similar to most State-of-the-Art (SoA) approaches we start from the *K-SVD* dictionary learning of Aharon et al. [2]. Because depth images contain a high variety of discontinuities caused by different scales and sensor modalities we use image features from normalized image patches as low resolution input. Similar to Zeyde et al. [137] we apply *PCA* dimensionality reduction projecting the features onto an even lower dimensional subspace. Further, we use Orthogonal Matching Pursuit (*OMP*) [116] to find the sparse code while training. The *OMP* is an algorithm to quickly solve the sparse coding problem.

In the training phase we start with a set of low and high resolution image pairs. From these training images we create a set of local patch pairs $\mathbf{Y} = \{\mathbf{Y}_l, \mathbf{Y}_h\} = \left\{F(\mathbf{p}_l^i), T(\mathbf{p}_h^{si})\right\}_i$ extracted at sub-sampled image locations $i = \{1 \dots p\}$ form $\mathbf{I}_L$ and si from $\mathbf{I}_H$, where s is the upsampling factor. The operator $F(.)\colon \mathbb{R}^n \rightarrow \mathbb{R}^f$ denotes the feature extraction and dimensionality reduction of the patch $\mathbf{p}_l$, where f is the feature length. $T(.)\colon \mathbb{R}^n \rightarrow \mathbb{R}^n$ denotes the calculation of the edge prior out of the high resolution image patch $\mathbf{p}_h$. In principle, different kinds of edge priors can be learned in our framework from different kinds of features. In our *SR* approach we use first and second order gradients as features to learn an anisotropic diffusion edge tensor as described later.

After determining the sampled patch pairs, the low resolution dictionary $\mathbf{D}_l \in \mathbb{R}^{f \times d}$ and the corresponding sparse code $\mathbf{\Lambda} \in \mathbb{R}^{d \times p} = \{\alpha_i\}$ is found by minimizing

$$\min_{\mathbf{\Lambda}, \mathbf{D}_l} \|\mathbf{Y}_l - \mathbf{D}_l \mathbf{\Lambda}\|_2^2, \qquad \text{s.t.} \quad \|\mathbf{\Lambda}\|_0 \leq L, \tag{4.42}$$

using the *K-SVD* algorithm, where the size of the dictionary d is fixed. L denotes the number of non-zero entries in the sparse code map $\mathbf{\Lambda}$. Given $\mathbf{\Lambda}$ the corresponding high resolution dictionary is calculated by the pseudo-inverse expression $\mathbf{D}_h \in \mathbb{R}^{n \times d} = \mathbf{Y}_h \mathbf{\Lambda}^T \left(\mathbf{\Lambda}\mathbf{\Lambda}^T\right)^{-1}$. This is given by the closed form solution of eqn. (4.42) for the dictionary in high resolution space, as shown in [137].

In the reconstruction phase traditional approaches solve eqn. (4.42) through *OMP* fixing the trained dictionary $\mathbf{D}_l$. The sparse code is estimated for each dictionary atom separately. After reconstruction, the code $\mathbf{\Lambda}$ is multiplied with the high resolution dictionary $\mathbf{D}_h$ to get the high resolution patches $\mathbf{Y}_h$. These patches are merged and averaged across the image space Ω_H to get the resulting image. The downside of this traditional approach of independent calculation and averaging without a neighboring coherence is that the result gets blurry in the overlapping region. This harms the *SR* quality which is based on the sharpness in the solution, as shown in [66].

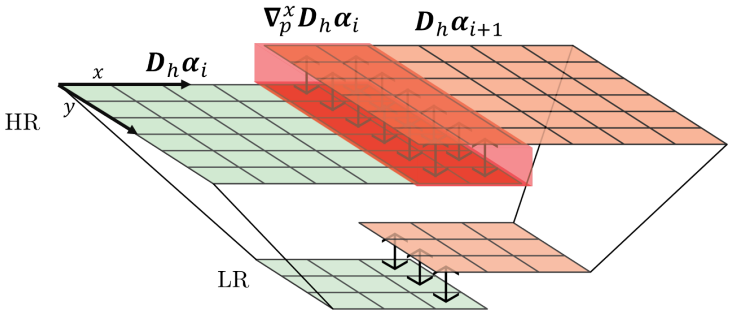


Figure 4.5: *Patch-Gradient.* The *patch-gradient* is formulated as the height difference between one patch $\mathbf{D}_h \alpha_i$ to its direct neighboring patch $\mathbf{D}_h \alpha_{i+1}$ in the image domain. It calculates the pixelwise difference in the overlapping region between two neighboring patches (red area).

In our work we introduce a binary term in the sparse optimization model to introduce spatial coherence of the patches. This enables to reconstruct the sparse code not only with respect to the input patch but also to the difference between neighboring patches. The low resolution patch-features are sparsely reconstructed using the following formulation:

$$\min_{\mathbf{\Lambda}} \|\mathbf{D}_l \mathbf{\Lambda} - \mathbf{Y}_l\|_2^2 + \lambda \|\mathbf{\Lambda}\|_1 + \gamma \|\nabla_p V(\mathbf{D}_h \mathbf{\Lambda})\|_1, \tag{4.43}$$

where the first term minimizes the distance of the low resolution dictionary atoms to the input and the second term minimizes the quantity of atoms used for reconstruction. The scalars $\lambda, \gamma \in \mathbb{R}$ weight the individual terms. The ℓ_0 constraint of the sparsity constraint is relaxed to a ℓ_1 norm constraint, as used in other methods [132]. The

additional third term reflects a regularization between overlapping regions of patches. The operator $V(.)\colon \mathbb{R}^{n\times p} \to \mathbb{R}^{np}$ denotes a vectorization of the matrix $\mathbf{D}_h\mathbf{\Lambda}$. The term $\nabla_p = \left[\nabla_p^x, \nabla_p^y\right]^T : \mathbb{R}^{np} \to \mathbb{R}^{2rp}$ denotes the novel *patch-gradient* operator, where r denotes the size of the overlapping region. Similar to the traditional *TV* regularization, the *patch-gradient* performs absolute forward differences between neighboring patches in x and y direction. For one high resolution patch it is defined as the sum of pixelwise differences in the overlapping region to its direct neighbor patch in image space. A visualization of this gradient is shown in Figure 4.5. The *patch-gradient* penalizer is applied by a simple matrix multiplication of the linear gradient operator ∇_p with the concatenated patch vector $V(\mathbf{D}_h\mathbf{\Lambda})$.

After optimization we get the concatenated high-resolution patches $\mathbf{D}_h\mathbf{\Lambda}$. Since our models finds dictionary entries where the neighbors are better aligned, the resulting image contains sharper edges after merging all the patches back together.

4.5.2 Anisotropic Edge Prior

In principle, different kinds of the edge priors can be learned in our framework (e.g. scalar weights, image gradients, guided image filters or shock filters). As shown in the last section we use an anisotropic diffusion tensor based on the Nagel-Enkelmann operator [72], since it worked best for all experiments. As already shown in Eqn. (4.32) for intensity images, given a high resolution depth patch $\mathbf{p}_h$ the anisotropic edge patch is calculated by

$$T(\mathbf{p}_h) = \exp\left(-\beta\left|\nabla\mathbf{p}_h\right|^\gamma\right)\mathbf{n}\mathbf{n}^{\mathrm{T}} + \mathbf{n}^{\perp}\mathbf{n}^{\perp\mathrm{T}}, \tag{4.44}$$

where $\mathbf{n}$ is the normalized direction of the image gradient $\mathbf{n} = \frac{\nabla\mathbf{p}_h}{|\nabla\mathbf{p}_h|}$, $\mathbf{n}^{\perp}$ is the normal vector to the gradient and the scalars $\beta, \gamma \in \mathbb{R}$ adjust the magnitude and the sharpness of the tensor. The gradients are calculated using the Sobel operator to reduce the influence of noise in the training data. The advantage of an anisotropic diffusion tensor is that it not only weights the regularization but also orients the gradient direction during the optimization process.

In our model the high resolution dictionary is composed of (ideally) incoherent edge tensor entries. After the sparse reconstruction the concatenated tensor entries $\mathbf{D}_h\mathbf{\Lambda}$ are merged to the image space resulting in the weighting tensor $\mathbf{T}_{\mathbf{\Lambda}} \in \mathbb{R}^{4\times\Omega_H}$. This tensor is used in to guide the regularization term in the image guided *SR* from Section 4.4.

4.5.3 Numerical Optimization

In this section we explain the details of the numerical optimization to solve the sparse reconstruction problem. Both proposed problems are convex but non-smooth due to the ℓ_1 and Huber norms in the different terms. Therefore, the optimization of such problems is not a trivial task. Since eqn. (4.43) is convex in $\mathbf{\Lambda}$ we make use of the dual principle. After introducing Lagrange multipliers for the constraints and biconjugation using the

LF we are able to reformulate the problems as convex-concave saddle point problems, as shown in Section 3.4.3, given by

$$\min_{\mathbf{\Lambda}} \max_{\mathbf{p},\mathbf{q}} \|\mathbf{D}_l\mathbf{\Lambda} - \mathbf{Y}_l\|_2^2 + \lambda \langle \mathbf{p}, \mathbf{\Lambda} \rangle_{Q_\mathbf{p}} + \gamma \langle \mathbf{q}, \nabla_p V(\mathbf{D}_h\mathbf{\Lambda}) \rangle_{Q_\mathbf{q}}, \tag{4.45}$$

where $\mathbf{p}$ and $\mathbf{q}$ denote the dual variables and $\mathbf{Y}_l$ the concatenated set of input features. The operators in the subscript denote a projection of the dual variable on the convex sets

$$Q_\mathbf{p} = \left\{ \mathbf{p} \in \mathbb{R}^{d \times p} \mid -1 \leq \mathbf{p}(i,j) \leq 1 \right\}, \qquad \forall i = 1 \dots d, j = 1 \dots p, \tag{4.46}$$

$$Q_\mathbf{q} = \left\{ \mathbf{q} \in \mathbb{R}^{2 \times rp} \mid \|\mathbf{p}_\mathbf{v}(k)\| \leq 1 \right\}, \qquad \forall k = 1 \dots rp. \tag{4.47}$$

With the formulation of the discrete saddle-point problem (4.45) we can apply the primal-dual algorithm [21]. The algorithm iteratively optimizes the primal and the dual variables for the individual pixels in three steps. First, the dual variables $\mathbf{p}$, $\mathbf{q}$ are updated using gradient ascend. Second, the primal variable $\mathbf{\Lambda}$ is updated using gradient descend. Third, the primal variable is refined in an over-relaxation step. The optimizer variables are initialized with $\mathbf{\Lambda} = \mathbf{0}$, $\mathbf{p} = \mathbf{0}$ and $\mathbf{q} = \mathbf{0}$. For any iteration $n \geq 0$ and every level the primal dual steps are calculated according to

$$\begin{cases} \mathbf{p}^{n+1} & = \mathcal{P}_{Q_\mathbf{p}} \left\{ \mathbf{p}^n + \sigma_\mathbf{p} \lambda \left(\bar{\mathbf{\Lambda}}^n \right) \right\}, \\ \mathbf{q}^{n+1} & = \mathcal{P}_{Q_\mathbf{q}} \left\{ \mathbf{q}^n + \sigma_\mathbf{q} \gamma \left(\nabla_p V(\mathbf{D}_h \bar{\mathbf{\Lambda}}^n) \right) \right\}, \\ \mathbf{\Lambda}^{n+1} & = \left(\mathbf{I} + \tau_\mathbf{\Lambda} \mathbf{D}_l^T \mathbf{D}_l \right)^{-1} \left(\mathbf{D}_l^T \mathbf{Y}_l - \lambda \mathbf{p} - \gamma \mathbf{D}_h^T V_I(\nabla_p^T \mathbf{p}) \right), \\ \bar{\mathbf{\Lambda}}^{n+1} & = 2\mathbf{\Lambda}^{n+1} - \bar{\mathbf{\Lambda}}^n, \end{cases} \tag{4.48}$$

Where $\mathbf{I}$ denotes the identity matrix and $V_I(.) \colon \mathbb{R}^{np} \to \mathbb{R}^{n \times p}$ denotes the inverse of the operator $V(.)$. The variables $\sigma_\mathbf{p}$, $\sigma_\mathbf{q}$ and $\tau_\mathbf{\Lambda}$ are the step sizes of the gradient ascend and gradient descend steps. The exact setting of the step sizes is explained in Section *Step Sizes*. The operators $\mathcal{P}_{Q_\mathbf{p}}$ and $\mathcal{P}_{Q_\mathbf{q}}$ are defined as the point-wise Euclidean projections onto the convex sets $Q_\mathbf{p}$ and $Q_\mathbf{q}$:

$$(\mathcal{P}_{Q_\mathbf{p}}(\tilde{\mathbf{p}})))(i,j) = \min\left\{1, \max\{-1, \tilde{\mathbf{p}}(i,j)\}\right\}, \tag{4.49}$$

$$(\mathcal{P}_{Q_\mathbf{q}}(\tilde{\mathbf{q}})))(k) = \left\{ \frac{\tilde{\mathbf{q}}}{\max\{1, \|\tilde{\mathbf{q}}(k)\|\}} \right\}, \tag{4.50}$$

The preconditioned step sizes for solving the sparse reconstruction Eqn. (4.48) are given by

$$\begin{aligned} \sigma_{\mathbf{p}} &= \frac{\sigma}{\lambda}, \quad \sigma_{\mathbf{q}}(i) = \frac{\sigma}{\sum_j |(\nabla_p \tilde{\mathbf{D}}_h)(i,j)|}, \\ \tau_{\mathbf{u}}(j) &= \frac{tau}{\sum_i |(\nabla_p \tilde{\mathbf{D}}_h)(i,j)|} \\ \forall i,j \colon i &= 1 \dots rp, \quad j = 1 \dots np. \end{aligned} \tag{4.51}$$

The matrix $\tilde{\mathbf{D}}_h$ denotes a concatenated diagonal matrix of the high resolution dictionary where $\mathbf{D}_h$ is repeated p times such that

$$\tilde{\mathbf{D}}_h = \begin{bmatrix} \mathbf{D}_h & & & \mathbf{0} \\ & \mathbf{D}_h & & \\ & & \ddots & \\ \mathbf{0} & & & \mathbf{D}_h \end{bmatrix} \in \mathbb{R}^{2rp \times dp}. \tag{4.52}$$

The result of this optimization gives us the reconstructed sparse code $\mathbf{\Lambda}$. The reconstructed high resolution patches $\mathbf{D}_h\mathbf{\Lambda}$ are merged into the image space to result in the final edge tensor $\mathbf{T}_{\mathbf{\Lambda}} \in \mathbb{R}^{\Omega_h} \mapsto \mathbb{R}^2$. This reconstructed edge tensor is directly used in the image guided depth *SR* from Eqn. (4.34) which is iteratively solved using the primal-dual scheme from Eqn. (4.35). The final optimization result gives the single depth *SR* result shown in Section 4.4.

4.6 Evaluation

In this section we show a quantitative and qualitative evaluation of the proposed *SR* methods. The evaluation in this section is divided into different part to evaluate all aspects and properties of the different variations. We compare our methods to multiple *SoA* methods from depth and intensity image superresolution. For an extensive analysis we investigate the performance compared on a variety of different synthetic and real datasets. For the synthetic evaluation we use the well known Middlebury datasets from [47, 100–102]. To follow *SoA* work we additionally evaluate our work on the Middlebury variations from Park et al. [82] (added depth dependent noise) and Aodha et al. [66] (filled out occlusions). For the real-world evaluations we use the Laser Scan Dataset of Aodha et al. [66] and our *ToF* camera dataset (*ToFMark*) proposed in [28].

We will first discuss some of the algorithm details for the single image superresolution from Section 4.5 such as used features and dictionary sizes. Further, we show the performance under increasing (acquisition) noise, where different levels of Gaussian noise are applied on the input data compared to other *SISR* approaches. After that we perform an extensive analysis of the proposed image guided *SR* and single image *SR* compared to

multiple *SoA* approaches in this fields. We evaluate all methods on the clean and noisy Middlebury dataset to show the performance of the different methods under each condition. This evaluation is shown in Section 4.6.3. Finally in Section 4.6.4 we show the *SR* results for real-world datasets where the *SR* performance for real *ToF* acquisitions is shown. We will evaluate all methods based on the Mean Absolute Error (MAE) and the Root Mean Squared Error (RMSE) since this widely used metrics give the best indication on the performance of noise reduction and edge sharpness.

Before we start our exhaustive evaluation we will shortly introduce a naming convention of the different approaches in Section 4.6.1. This naming convention is used throughout the whole section.

To allow for a fair comparison all weighting parameters in our model are set once and are kept constant over all experiments.

4.6.1 Method Namings

The namings in our evaluation are stated as follows. For the proposed *SR* methods the name is composed of the different parts that are used in the energy functional. The first part of the method name is the optimization model followed by the source of the image guidance. Hence, the complete method name is defined as

Model	Abbreviation
No guidance + *TGV* (Section 4.3)	*TGV-SR*
$\mathbf{I}_H^I$ guided + anisotropic *TGV* (Section 4.4)	*aTGV-IG*
Variational Sparse Coding $\mathbf{T}_{\mathbf{\Lambda}}$ guided + anisotr. *TGV* (Section 4.5)	*aTGV-VSC*

For evaluation of the *SoA* approaches we use the publicly available framework of Timofte et al. [114]. In the following we compare our method with the standard interpolation methods Nearest Neighbor, Bilinear and Bicubic upsampling as well as the sparse coding approach of Yang et al. [132] and Zeyde et al. [137]. We further show the results of both methods reported in [114], namely Global Regression (GR) and Anchored Neighborhood Regression (ANR), and the neighborhood embedding [8, 23] approaches (*NE+LS*, *NE+NNLS*, *NE+LLE*). Additionally, we compare to the *MRF* based methods of Aodha et al. [66] and Hornáček et al. [50].

The approaches for image guided depth *SR* which are compared in this evaluation are the *MRF* based methods of Diebel and Thrun [25] and Park et al. [82]. We further compare to the bilateral filter approaches of Yang et al. [133], Chan et al. [22] and Kopf et al. [55], and to the guided image filtering approach of He et al. [43].

In Table 4.1 we define the namings of the *SoA* methods which are used throughout the evaluation.

Papar	Principle	Abbreviation
-	nearest neighbor interpolation	*Nearest*
-	bilinear interpolation	*Bilinear*
-	bicubic interpolation	*Bicubic*
He et al. [43]	Guided Image Filtering	*GIF*
Yang et al. [132]	Sparse Coding	*Yang-SC*
Zeyde et al. [137]	Sparse Coding	*Zhang-SC*
Timofte et al. [114]	Sparse Coding/Neighbor Embedding	*GR*
Timofte et al. [114]	Sparse Coding/Neighbor Embedding	*ANR*
Bevilacqua/Chang et al. [8, 23]	Neighbor Embedding	*NE+LS*
Bevilacqua/Chang et al. [8, 23]	Neighbor Embedding	*NE+NNLS*
Bevilacqua/Chang et al. [8, 23]	Neighbor Embedding	*NE+LLE*
Aodha et al. [66]	Patch-Dictionary + *MRF*	*Aodha*
Hornáček et al. [50]	Patch-Search + *MRF*	*Hornáček*
Kopf et al. [55]	Joint Bilateral Upsampling	*JBU*
Chan et al. [22]	Joint Bilateral Filtering	*Chan-IG*
Yang et al. [133]	Joint Bilateral Filtering	*Yang-IG*
Diebel et al. [25]	Image guided *MRF*	*Diebel-IG*
Park et al. [82]	Image guided *MRF*	*Park-IG*

Table 4.1: Naming Overview. Overview of the *SoA* methods used in this evaluation separated by their Principle. Intensity image guided methods are indicated by *IG* and methods based on sparse coding with *SC*.

4.6.2 Discussion of Sparse Coding Details

In this section we evaluate the properties of our variational sparse coding (*VSC*) model compared to traditional sparse coding methods. As defined in Section 4.5 the *SR* models based on sparse coding use the feature of a *LR* input and estimate its *HR* version based on a learned dictionary. In this estimation the choice of the patch feature is a crucial part. For *SR* based on sparse representations one of the most basic features is to use the sample patches itself. Other methods such as [114, 132, 137] use the first and second order derivative for intensity image features. However, for depth images this is not directly applicable, because the range from minimum to maximum value greatly varies between different scenes. To tackle this problem Aodha et al. [66] proposed a patch normalization which accounts for the ranges of both high and low resolution patch. In our work we first normalize each patch to $[0, 1]$ and then use the first and second order gradients as patch-features. An additional PCA dimensionality reduction is applied to project the feature vector onto a lower dimensional subspace while preserving 99.9% of the average energy. E.g. an upscaling factor of 3 this reduces each feature of length 144 to a size of about 36. Throughout all experiments we use a patch-size of 3×3 in the low resolution image space,

which delivers the best results for all sparse coding approaches.

For all dictionary based methods we use the same synthetic range image data of [66] for training, which contains 30 scenes of size 800×800 in the high resolution space. The reported error is described as the *RMSE* to a known groundtruth.

The choice and the size of the dictionary are very critical parts in any sparse coding approach. Usually, the more incoherent atoms a dictionary contains the better the performance, however, this comes with a higher computational cost. In Figure 4.6 we show the influence of the dictionary size on the performance. As expected, the performance increases with the size of the dictionary. But while the performance of other sparse coding methods drastically increases with the size our method already starts at a much lower *RMSE* and is less influenced by the choice of the dictionary size.

For most depth *SR* approaches the correct noise handling plays a major role. Therefore, we test the accuracy under different levels of noise on the Middlebury dataset. We chose a depth dependent Gaussian noise with zero mean, as reported in [82]. The standard deviation of the noise ranges from $0 - 50\%$ of the depth range (minimum to maximum) in the input images for an upsampling factor of $\times 3$. In Figure 4.7a and Figure 4.7b the error results are shown for the different methods. In Figure 4.8 we show visual *SR* results of different methods for a standard deviation of 2%. Obviously, the error increases with the input noise for all methods. But, while the error drastically increases for methods which solely depend on the sparse reconstruction, the variational method [117] produces a higher error at lower noise factors and performs comparably better with increasing noise. This is caused by the regularization of the depth during optimization which reduces the noise but smooths the edges. Since we use a regularization which is only guided by a sparse edge reconstruction we get more accurate results over the whole noise range.

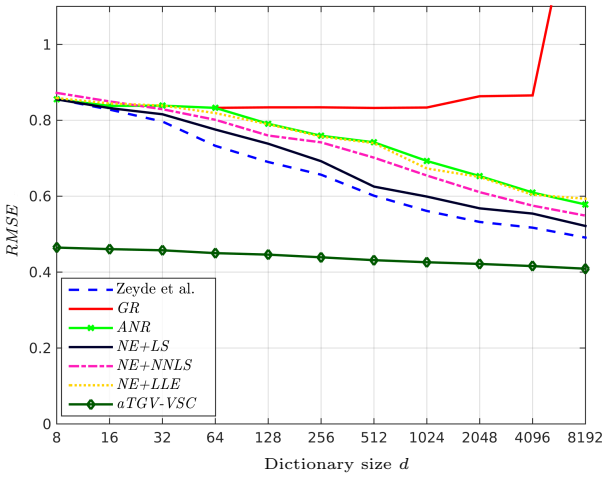
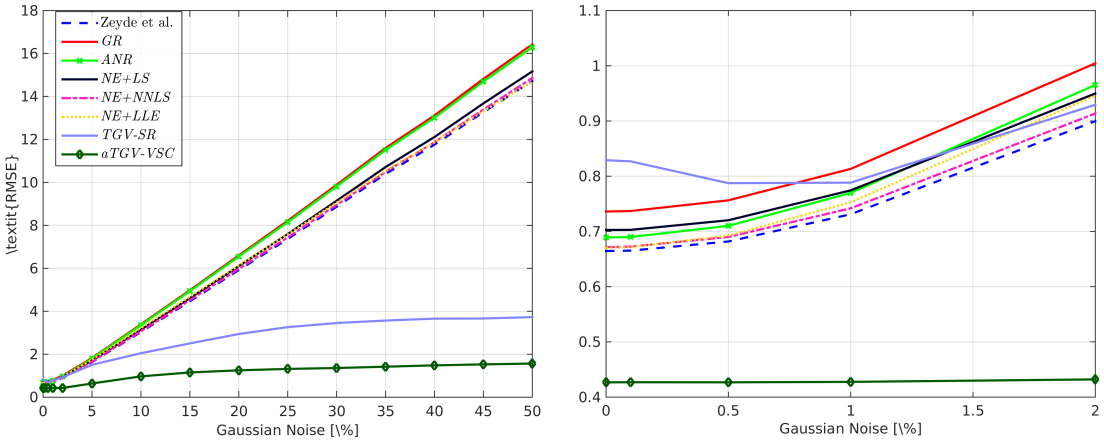


Figure 4.6: Influence of dictionary size on the average accuracy (*RMSE* in pixel disparity) for the Middlebury images *Teddy*, *Cones*, *Venus* and *Tsukuba* with a magnification of $\times 3$. All neighborhood embedding approaches where used with their best neighborhood size (as reported in [114]). In the results are shown where each sparse coding method uses the same dictionary.



(a) (b)

Figure 4.7: Influence of input noise on the average *RMSE* (in pixel disparity) for the Middlebury images *Teddy*, *Cones*, *Venus* and *Tsukuba* with a magnification of $\times 3$. All neighborhood embedding approaches where used with their best neighborhood size (as reported in [114]). In (a) the results for increasing Gaussian noise are shown, where every sparse coding method shares the same dictionary with 1024 entires. In (b) a magnified sector of the noise evaluation in (b) is shown, where the noise level ranges from 0-2%. Figure best viewed magnified in the electronic version.

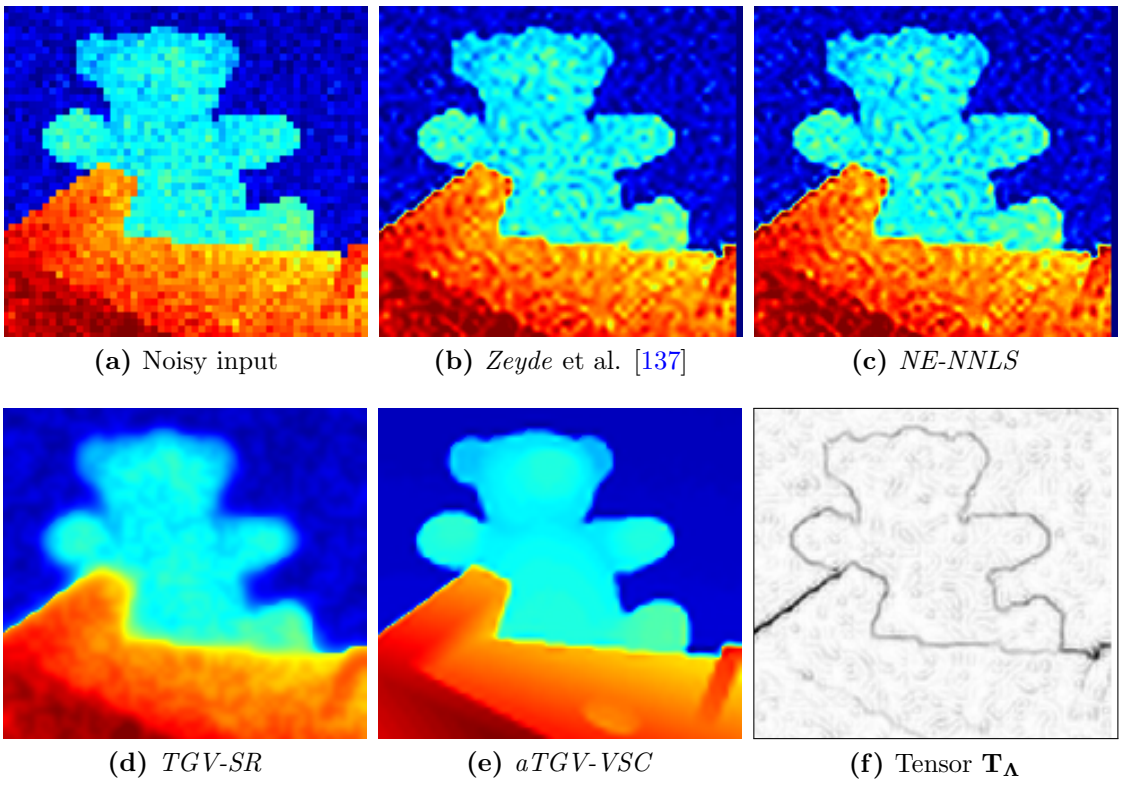
4.6.3 Middlebury Benchmark Evaluation

In this section we evaluate the performance of the different *SoA* methods on the publicly available Middlebury benchmark. Following most image guided *SR* methods we will first show the *MAE* results for the noisy and noise-free Middlebury datasets *Art*, *Books* and *Moebius* treating the disparity values as depth. In this evaluation the low resolution input depth is given by bilcubic downsampling with downsampling factors of $\times 2$, $\times 4$, $\times 8$ and $\times 16$. For the noisy example we use the noisy Middlebury dataset provided by Park et al. [82]. To simulate the noise characteristics of most depth cameras depends where the noise increases with the distance between camera and scene depth dependent Gaussian noise is added to the low resolution input depth images. According [82] this Gaussian noise is defined by

$$\mathbf{v}(k, \sigma_d) = k \exp\left(-\frac{1}{2(1+\sigma_d)^2}\right), \tag{4.53}$$

where σ_d is a value proportional to depth and k is the magnitude of the Gaussian noise.

Additionally, following [50, 66] we show the *MAE* for single images *SR* results on the Middlebury datasets *Teddy*, *Cones*, *Venus* and *Tsukuba* for upsampling factors of $\times 2$ and $\times 4$ where the groundtruth is downsampled by nearest neighbor interpolation. Further, we show the results for the real-world laserscan dataset (*Scan21*, *Scan30*, *Scan42*) proposed by [66] for an upsampling factor of $\times 4$.



(a) Noisy input **(b)** *Zeyde* et al. [137] **(c)** *NE-NNLS*

(d) *TGV-SR* **(e)** *aTGV-VSC* **(f)** Tensor $\mathbf{T}_{\mathbf{\Lambda}}$

Figure 4.8: Color-coded visual *SR* results for noisy input data. The figure shows a zoomed region of interest from the *Teddy* dataset for an upsampling factor of $\times 3$. On the low resolution input we applied Gaussian noise with zero mean and a standard deviation of 2% of the input disparity range.

The quantitative results for the noisy Middlebury datatset is shown in Table 4.2 and the results for the clean Middlebury dataset is shown in Table 4.3. Exemplar visual results are shown in Figure 4.9, 4.10, 4.11 and 4.12.

The comparison to [66] and [50] for single image superresolution is shown in Table 4.4 for nearest neighbor downsampling of the input image. To evaluate the influence of our sparse coding scheme to the overall solution we compare to a combination of the sparse coding method [137] for edge prior estimation in our variational *SR* ([137] + *aTGV-SR*).

4.6.3.1 Discussion

In the results it can be clearly seen that there is a big difference between between the clean and noisy data in the input depth maps. While approaches based on sparse coding are very well suited for clean data, image guided optimization approaches deliver much better results for noisy data, as shown in Table 4.3 and Table 4.2.

For the clean data it can be clearly seen that the methods based on a sparse representations (Yang-SC, Zeyde-SC, NE+NLL, ...) have slightly better performance than methods

based on a *MRF* (Diebel, Park, Yang) for lower upsampling factors. Further is can be seen that our methods are overall superior compared to most other methods. This quality improvement originates from a *TGV* regularization with anisotropic tensor together with the basic *SR* problem. The higher order regularization better captures the surface of real world scenes, while the anisotropic tensor delivers a defined guidance of the optimization. It can be seen that the guidance from a the variational sparse coding in our optimization ($aTGV\text{-}VSC$) is on par with the image guided approach ($aTGV\text{-}IG$). This is because the variational sparse coding does only rely on the depth input. Hence, if the intensity texture does not coincide with depth steps this will not lead to errors in the optimization.

When dealing with noise in the input acquisition one can see the clear advantage of the intensity image guided models. While the models based on sparse coding have problems in the presents of noise, models with a regularization term have a big advantage since they implicitly denoise the *SR* result. Furthermore, the image guided methods have the big advantage that they can rely on an intensity image for guidance, which has a comparably low amount of noise. Similar to the sparse coding methods the *SR* quality decreases with increasing levels of noise in the guidance image. The proposed *SR* method guided by the variational sparse coding delivers superior results, in contrast to all other sparse coding methods. For smaller magnification factor it delivers even better than most image guided methods. Overall, our optimization model with higher order regularization and anisotropic guidance from a noise-free intensity image delivers the best results for every magnification factor and dataset.

	×2			×4			×8			×16		
	Art	Books	Moebius	Art	Books	Moebius	Art	Books	Moebius	Art	Books	Moebius
Nearest	4.6529	4.6011	5.0752	5.0147	4.6848	5.1979	5.7090	4.8479	5.3051	7.0978	5.2274	5.6492
Bilinear	3.0846	2.9172	3.2084	3.5861	3.1195	3.4471	4.3931	3.3427	3.6238	5.9087	3.7118	3.9991
Bicubic	3.8124	3.7367	4.1121	4.1489	3.8127	4.2168	4.7827	3.9843	4.3069	6.1180	4.2867	4.6066
GIF	1.9190	1.6025	1.7722	2.3962	1.8164	2.0271	3.3209	2.3136	2.5960	5.0751	3.0592	3.3394
Yang-SC	6.8161	6.8761	7.6471	3.7816	3.4060	3.8134	-	-	-	-	-	-
Zeyde-SC	5.0151	5.1366	5.6733	4.7738	4.4034	4.8736	5.5444	5.1934	5.6873	5.9181	4.8617	5.3616
GR	5.8324	5.8823	6.4912	3.7864	3.0847	3.4006	6.4245	5.9010	6.4256	7.2734	5.9083	6.3829
ANR	5.5240	5.6523	6.2415	5.4898	5.0616	5.5975	6.1235	5.7548	6.2937	6.7339	5.6257	6.1772
NE+LS	5.1830	5.2901	5.8398	4.5811	4.1666	4.6158	5.7287	5.3374	5.8197	6.5745	5.3556	5.8672
NE+NNLS	5.1179	5.2162	5.7510	4.8598	4.3479	4.8478	5.6531	5.2209	5.6895	6.6501	5.3076	5.7679
NE+LLE	5.5112	5.6392	6.2251	5.4364	5.0805	5.5974	5.9192	5.5665	6.0901	6.4253	5.3023	5.7845
Chan-IG	2.0229	1.0366	1.1674	2.6310	1.3550	1.5482	3.8127	1.9398	2.2848	5.3957	3.0694	3.5528
Yang-IG	1.3592	1.1191	1.2506	1.9315	1.4721	1.6334	2.4535	1.8055	2.0559	4.5192	2.9235	3.2054
Diebel-IG	1.6198	1.3399	1.4653	2.5428	2.0840	2.2855	3.8514	2.8537	3.0946	5.6960	3.5412	3.8133
Park-IG	1.2401	0.9861	1.0343	1.8159	1.4301	1.4894	2.7047	1.9771	2.1289	4.3940	3.0358	3.0910
TGV-SR	1.4228	1.3103	1.4787	1.7824	1.3390	1.5007	3.0938	2.0296	2.2620	5.3096	3.1303	3.4328
aTGV-IG	**0.8365**	**0.5054**	**0.5695**	**1.2908**	**0.7526**	**0.8983**	**2.0562**	**1.1556**	**1.3786**	**3.5634**	**1.8872**	**2.1503**
aTGV-VSC	1.0729	0.9422	1.1477	2.6443	1.6603	1.9656	5.2530	2.6544	2.9765	8.3711	4.0739	4.3922

Table 4.2: Quantitative comparison on the Middlebury 2007 datasets with added noise. The error is measured as *RMSE* of the pixel disparity for four different magnification factors (×2, ×4, ×8, ×16).First, we show the results for standard interpolation methods. Second, the results for *SoA* image guided *SR* methods is shown. Third, we show the results for *SoA* sparse coding *SR*. Fourth the results of the proposed methods is shown. The best result for each dataset and upscaling factor is highlighted and the second best is underlined.

	×2			×4			×8			×16		
	Art	Books	Moebius	Art	Books	Moebius	Art	Books	Moebius	Art	Books	Moebius
Nearest	0.5155	0.1737	0.1824	1.0730	0.3513	0.3810	2.0722	0.6718	0.7517	4.0151	1.3545	1.4149
Bilinear	0.5247	0.1497	0.1676	1.0533	0.3108	0.3436	2.0684	0.6109	0.6758	3.9961	1.2042	1.2873
Bicubic	0.4797	0.1342	0.1501	0.9857	0.2878	0.3178	1.8842	0.5550	0.6185	3.6531	1.0817	1.1298
GIF	0.6598	0.2191	0.2392	1.0564	0.3557	0.3776	1.7726	0.5951	0.6099	3.6251	1.1593	1.1998
Yang-SC	0.9200	0.4473	0.3994	1.2170	0.4610	0.4788	-	-	-	-	-	-
Zeyde-SC	0.2834	0.0932	0.1146	0.7954	0.2611	0.2811	1.7458	0.5727	0.5945	3.9014	1.1131	1.1139
GR	0.4899	0.1383	0.1494	1.6043	0.4832	0.5137	1.9293	0.5825	0.6311	3.6108	1.1142	1.1278
ANR	0.3338	0.0979	0.1075	1.3494	0.3930	0.4482	1.5737	0.4613	0.5126	3.0099	0.8542	0.9213
NE+LS	0.2665	0.0798	0.0932	1.1480	0.3595	0.3873	1.4293	0.3948	0.4662	2.8984	0.7912	0.9228
NE+NNLS	0.3009	0.0919	0.1022	1.3000	0.3884	0.4354	1.5583	0.4358	0.4899	3.1328	0.8314	0.9302
NE+LLE	0.4801	0.3641	0.2916	2.3556	1.5656	1.6675	1.6269	0.5861	0.6644	3.0540	0.8475	0.9025
Yang-IG	0.5708	0.3013	0.3868	0.7002	0.4514	0.4760	1.5046	0.6373	0.6893	3.6903	1.4532	1.1366
Diebel-IG	0.6249	0.2166	0.2502	1.0052	0.3331	0.3679	1.9741	0.6162	0.6731	3.9370	1.2107	1.2884
Park-IG	0.4306	0.1954	0.1795	0.6745	0.3228	0.2965	1.0734	0.5542	0.5218	2.2117	1.0525	0.8965
TGV-SR	0.2683	0.0882	0.0894	0.7699	0.2897	0.3059	2.1093	0.8744	0.9574	3.9624	1.3666	1.4840
aTGV-IG	0.2590	0.0759	0.0841	**0.5085**	**0.1702**	**0.1880**	**1.0094**	0.3460	**0.3933**	**2.1483**	0.8209	0.9093
aTGV-VSC	**0.2494**	**0.0705**	**0.0838**	1.0865	0.3556	0.3739	1.3247	**0.3423**	0.4249	2.5730	**0.6696**	**0.7925**

Table 4.3: Quantitative comparison on the noise-free Middlebury 2007 datasets. The error is measured as *MAE* of the pixel disparity for four different magnification factors (×2, ×4, ×8, ×16). First, we show the results for standard interpolation methods. Second, the results for *SoA* image guided *SR* methods is shown. Third, we show the results for *SoA* sparse coding *SR*. Fourth the results of the proposed methods is shown. The best result for each dataset and upscaling factor is highlighted and the second best is underlined.

	×2				×4				×4		
	Cones	*Teddy*	*Tsukuba*	*Venus*	*Cones*	*Teddy*	*Tsukuba*	*Venus*	*Scan21*	*Scan30*	*Scan42*
Nearest	1.0943	0.8149	0.6123	0.2676	1.5309	1.1292	0.8328	0.3679	0.0177	0.0163	0.0396
Bicubic	0.9598	0.6917	0.5228	0.2274	1.2386	0.8936	0.6685	0.2938	0.0132	0.0125	0.0326
Diebel	0.7397	0.5265	0.4013	0.1703	1.1406	0.8010	0.5490	0.2426	-	-	-
Yang-SC	1.4794	1.0909	0.8583	0.3666	1.3239	0.9401	0.6849	0.3010	0.0138	0.0130	0.0337
Zeyde-SC	0.6920	0.4904	0.3871	0.1650	0.9617	0.6953	0.5477	**0.2199**	0.0100	0.0093	0.0246
GR	0.7780	0.5521	0.4289	0.1896	1.0790	0.8193	0.6480	0.2776	0.0117	0.0114	0.0271
ANR	0.6968	0.4954	0.3830	0.1666	1.0050	0.7564	0.6019	0.2452	0.0106	0.0101	0.0264
NE+LS	0.7066	0.4957	0.3939	0.1712	8.6221	10.3913	0.5641	14.7920	0.0818	0.1090	0.0725
NE+NNLS	0.6886	0.6073	0.3939	0.1646	0.9906	0.7346	0.5704	0.2431	0.0106	0.0101	0.0238
NE+LLE	0.6942	0.4995	0.3813	0.1654	0.9766	0.7396	0.5706	0.2406	0.0102	0.0097	0.0262
Aodha	1.1269	0.8247	0.6012	0.2761	1.5042	1.0259	0.8333	0.3365	0.0175	0.0170	0.0452
Hornáček	0.9936	0.7910	0.5802	0.2574	1.3986	1.1957	0.7272	0.4501	0.0205	0.0179	0.0299
TGV-SR	1.1342	0.8446	0.6445	0.2789	1.5797	1.1131	0.8438	0.3660	0.0170	0.0157	0.0415
aTGV-VSC	**0.6247**	**0.4397**	**0.3504**	**0.1433**	**0.9334**	**0.6670**	**0.4901**	0.2262	**0.0085**	**0.0083**	**0.0190**
Zeyde-SC + Our SR	0.6450	0.4543	0.3700	0.1573	0.9430	0.6769	0.4983	0.2363	0.0205	0.0179	0.0299

Table 4.4: Quantitative evaluation on the Aodha dataset. The *RMSE* is calculated for different *SoA* methods for the Middlebury and the Laserscan dataset for factors of ×2 and ×4. The first four rows show the comparison against two standard interpolation techniques and two depth *SR* which use an *HR* intensity image for guidance. The best result of all single image methods for each dataset and upscaling factor is highlighted and the second best is underlined. Additionally we show the sparse coding method [137] used for the edge prior estimation in our *SR* optimization. The error numbers are given in pixel disparity for the Middlebury and in [mm] for the Laserscan dataset.

4.6.4 Benchmarking based on Real Sensor Data

For the evaluation on real sensor acquisitions we use our publicly available *ToFMark* dataset from [28]. It is made using different scenes acquired with a *ToF* and an intensity camera simultaneously. For depth measurements we use a PMD Nano *ToF* camera delivering a 120×160 dense depth and IR amplitude image [83]. The intensity image is acquired by a CMOS camera with a sensor size of 810×610 pixel. In this experiment we could not evaluate methods based on sparse coding since the intensity image and the *LR* depth image are acquired from different camera poses.

4.6.4.1 ToFMark Evaluation

The groundtruth measurements are generated using a structured light scanner which consists of two 2048×2048 pixel high-speed intensity cameras and one high-speed projector. The whole multi-camera model is calibrated with our camera calibration framework proposed in [30].

The depth uncertainty of the complete structured light system at the given baseline is 1.2mm. to generate one consistent depth map, multiple acquisitions with slightly displaced projection angles are fused together while the position of the stereo cameras is kept constant. The acquired scenes are chosen to incorporate structures with high texture variations (see *Books* scene) as well as thin wiry elements (*Shark* and *Devil* scenes) to evaluate the upsampling accuracy. All scenes lie in the depth range of $0.8 - 1.2$m which reflects the operation distance of modern *ToF* cameras.

A quantitative accuracy evaluation of our *SR* for three real world datasets is shown in Table 4.5. The upsampling error is calculated by the *MAE* to the groundtruth depth map measured with the highly accurate structured light scanner. Additionally to the standard interpolation techniques and the methods based on sparse coding we compared our methods to two common image guided interpolation techniques, joint bilateral upsampling proposed by Kopf et al. [55] and the guided image filtering technique of He et al. [43]. The dictionary for the sparse coding techniques was trained for an upsampling factor of 6. As depth input to our methods we used the sparse *ToF* depth input $\mathbf{I}_S^D$ and the intensity image $\mathbf{I}_H^I$. The visual results for all three datasets are shown in Figure 4.13, 4.13 and 4.15.

4.6.4.2 Discussion

One issue that occurs in real world datasets is that wrong *ToF* measurements result in displaced surfaces in the upsampled result. Another problem arises due to the difference in the viewpoint of the observing cameras. Thus, the projected depth measurements near large depth steps can differ from correct depth values. Because the distance between the cameras is very small compared to the measured depth range, these wrong measurements have no large impact on the result and can be handled by the regularization in our variational model. The approximate upsampling factor of $\times 6.25$ for all datasets.

	Books	Devil	Shark
Nearest	18.7834	19.3200	20.9710
Bilinear	17.2005	17.4737	18.9881
Bicubic	17.5838	18.0166	19.4697
Yang-SC	16.5190	16.7171	18.3103
Zeyde	16.5377	16.9484	18.0739
GR	16.9745	17.1833	18.7332
ANR	17.1124	17.4507	18.8782
NE+LS	18.2209	18.3561	20.8852
NE+NNLS	17.9824	18.2196	20.4539
NE+LLE	17.1654	17.4983	18.8750
JBU	15.7371	16.3449	17.2538
GIF	15.2612	15.7724	16.8543
TGV-SR	16.7697	16.3153	21.5916
aTGV-IG	**12.2046**	**14.6083**	**15.0586**
aTGV-VSC	16.2903	16.3702	17.9358

Table 4.5: Quantitative evaluation on the real datasets *Books*, *Shark* and *Devil*. The error is calculated as *RMSE* to the measured groundtruth in mm. We compare standard interpolation methods as well as joint bilateral upsampling [55] and guided image filtering [43], and *SoA* sparse coding methods to our approach. The input density value shows the percentage of sparse depth values which are projected into the high resolution image space. This corresponds to an upsampling factor of approximately $\times 6.25$.

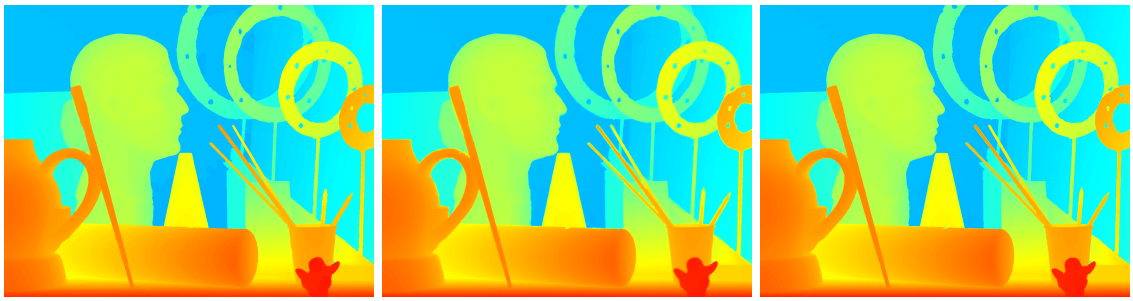
Our variational sparse coding approach (*aTGV-VSC*) delivers more accurate results than *SoA* methods based on sparse coding on all datasets. Similar to the Middlebury evaluation above this results from their inability to deal with input noise. Overall it can be seen that the methods without a guidance image deliver worse results compared to the guided methods (*JBU*, *GIF* and *aTGV-IG*). Further, it can be seen that our optimization model without guidance (*TGV-SR*) loses fine details and oversmooths the *SR* result compared to the model with guidance.

Through the additional incorporation of the anisotropic diffusion tensor and the higher order regularization into our optimization, the acquisition noise is drastically reduced, while sharp edges and smooth surfaces are preserved. Compared to other methods the image guided optimization *aTGV-IG* delivers superior results for all datasets.

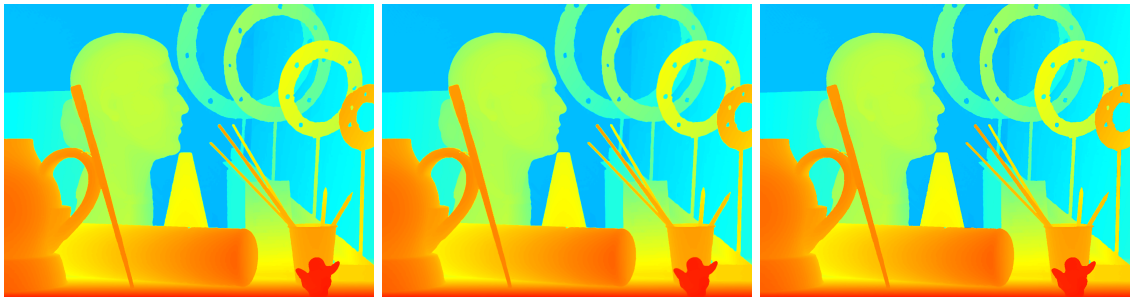
v

(a) *LR* input $\mathbf{I}_L^D$ **(b)** Intensity image $\mathbf{I}_H^I$ **(c)** Groundtruth $\mathbf{I}_H^D$

(d) *GIF* **(e)** *Zeyde-SC* **(f)** *NE+LS*

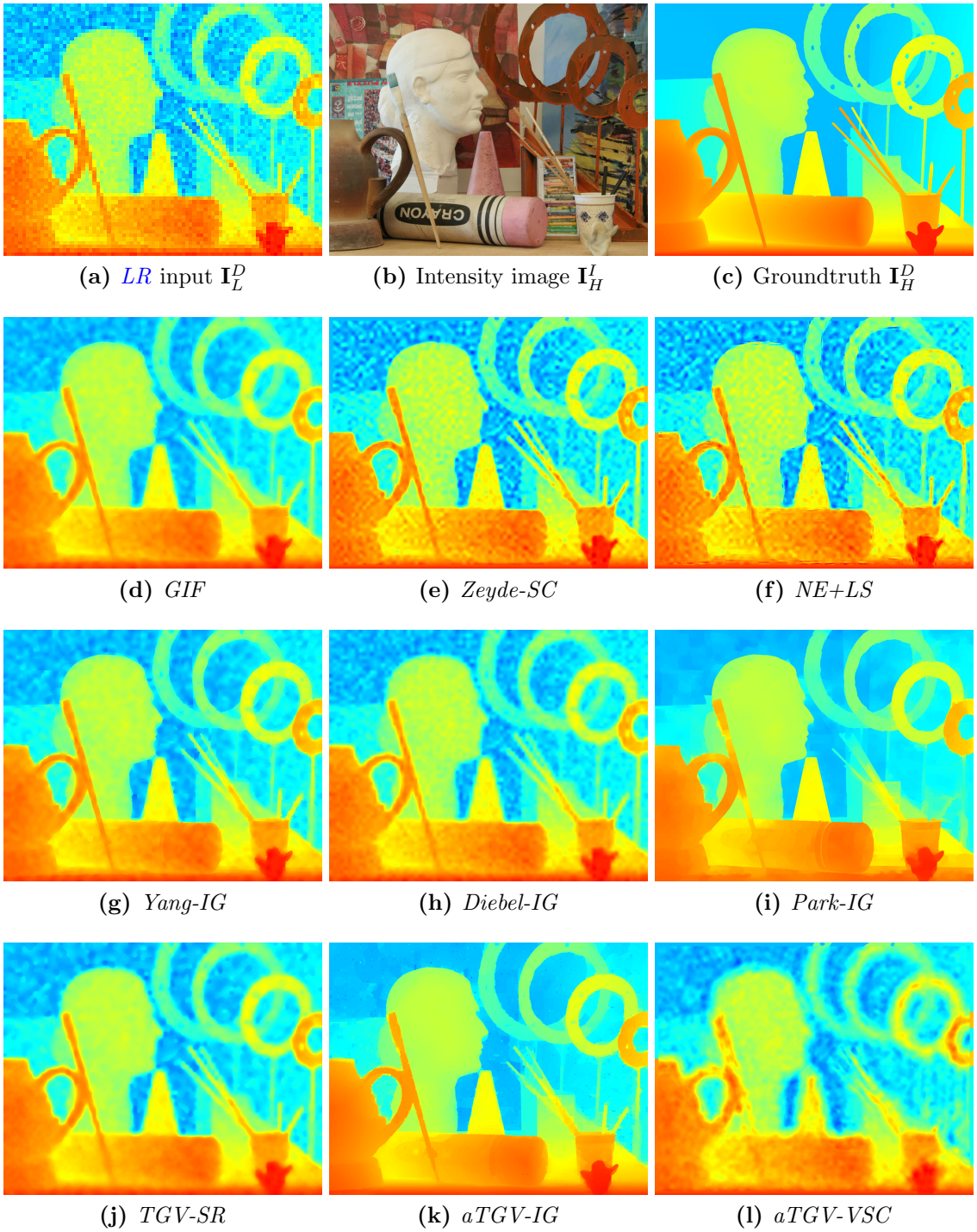


(g) *Yang-IG* **(h)** *Diebel-IG* **(i)** *Park-IG*



(j) *TGV-SR* **(k)** *aTGV-IG* **(l)** *aTGV-VSC*

Figure 4.9: Visual evaluation on the clean Middlebury *Art* dataset for a magnification factor of $\times 2$. In the first row the input and Groundtruth is shown. In the following rows a selection of the best methods for interpolation, sparse codding and optimization is shown, while in the last row the results of the proposed methods are shown.



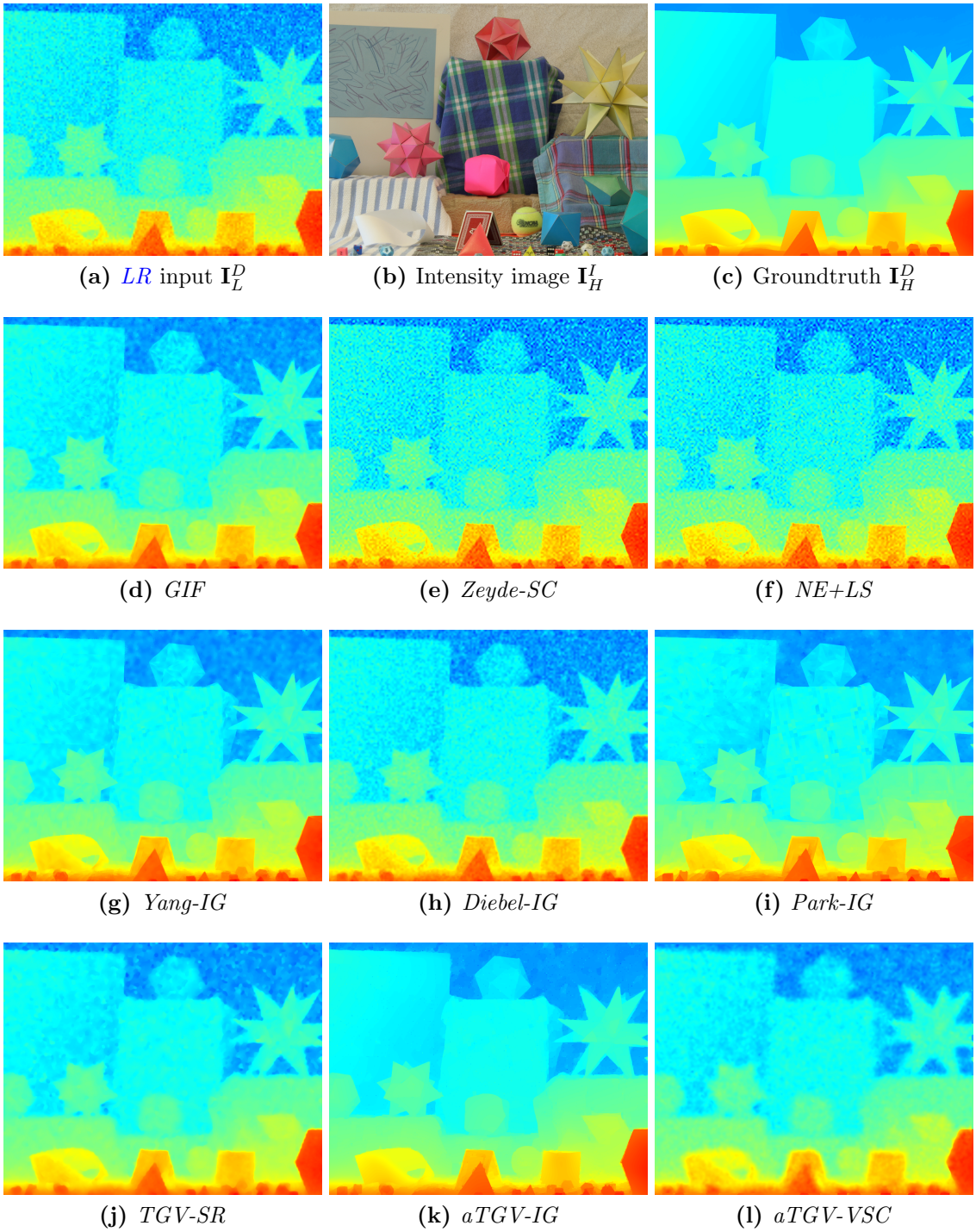
(a) *LR* input $\mathbf{I}_L^D$ **(b)** Intensity image $\mathbf{I}_H^I$ **(c)** Groundtruth $\mathbf{I}_H^D$

(d) *GIF* **(e)** *Zeyde-SC* **(f)** *NE+LS*

(g) *Yang-IG* **(h)** *Diebel-IG* **(i)** *Park-IG*

(j) *TGV-SR* **(k)** *aTGV-IG* **(l)** *aTGV-VSC*

Figure 4.10: Visual evaluation on the noisy Middlebury *Art* dataset for a magnification factor of $\times 16$. In the first row the input and Groundtruth is shown. In the following rows a selection of the best methods for interpolation, sparse codding and optimization is shown, while in the last row the results of the proposed methods are shown.



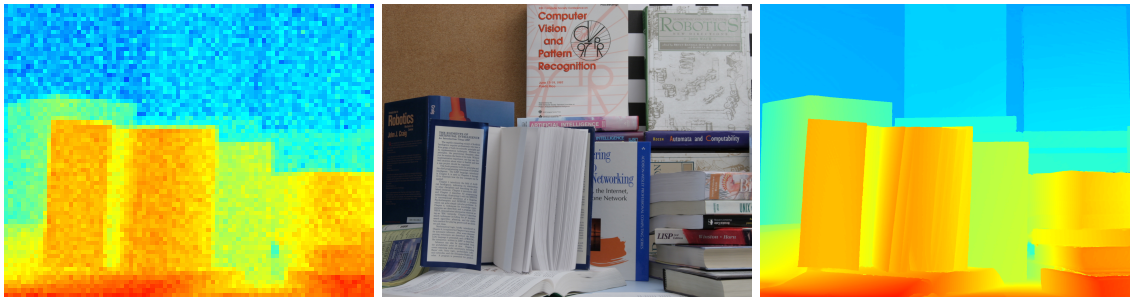
(a) *LR* input $\mathbf{I}_L^D$ **(b)** Intensity image $\mathbf{I}_H^I$ **(c)** Groundtruth $\mathbf{I}_H^D$

(d) *GIF* **(e)** *Zeyde-SC* **(f)** *NE+LS*

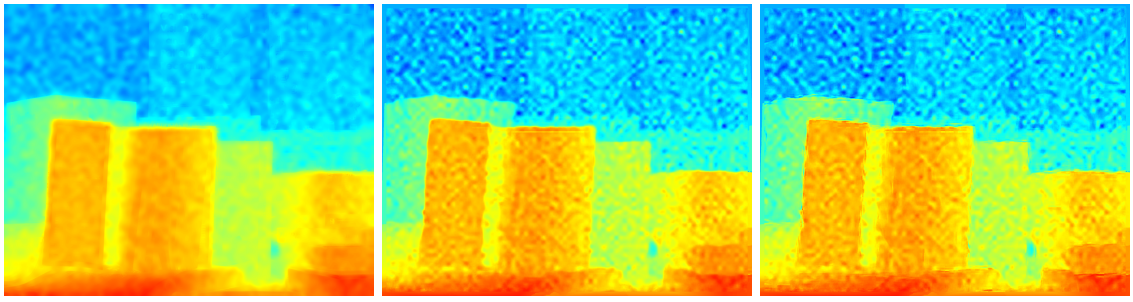
(g) *Yang-IG* **(h)** *Diebel-IG* **(i)** *Park-IG*

(j) *TGV-SR* **(k)** *aTGV-IG* **(l)** *aTGV-VSC*

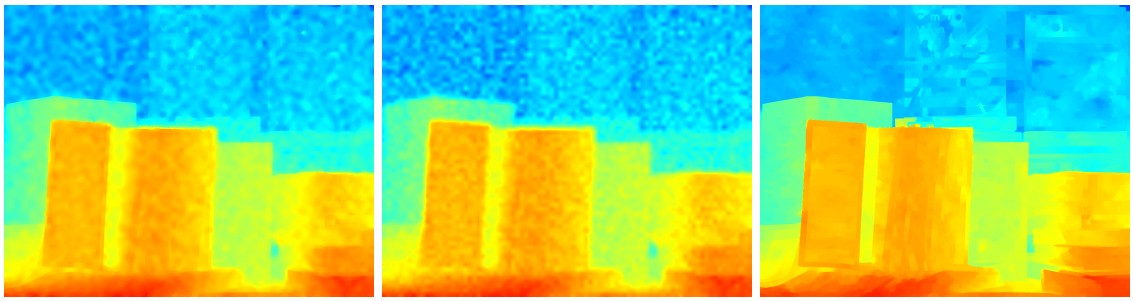
Figure 4.11: Visual evaluation on the noisy Middlebury *Moebius* dataset for a magnification factor of $\times 8$. In the first row the input and Groundtruth is shown. In the following rows a selection of the best methods for interpolation, sparse codding and optimization is shown, while in the last row the results of the proposed methods are shown.



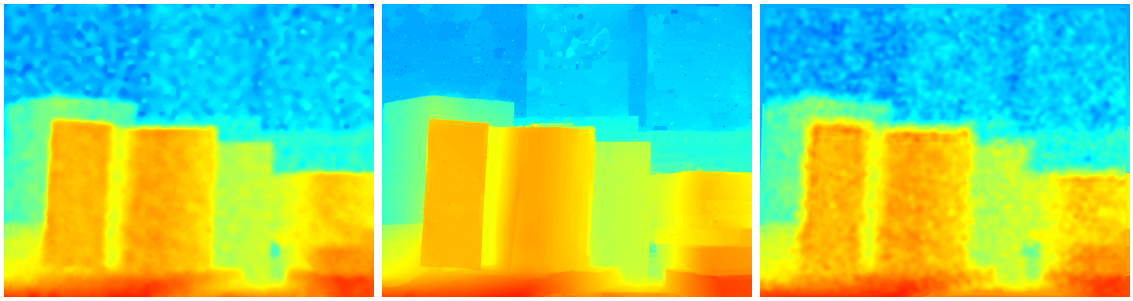
(a) *LR* input $\mathbf{I}_L^D$ **(b)** Intensity image $\mathbf{I}_H^I$ **(c)** Groundtruth $\mathbf{I}_H^D$



(d) *GIF* **(e)** *Zeyde-SC* **(f)** *NE+LS*

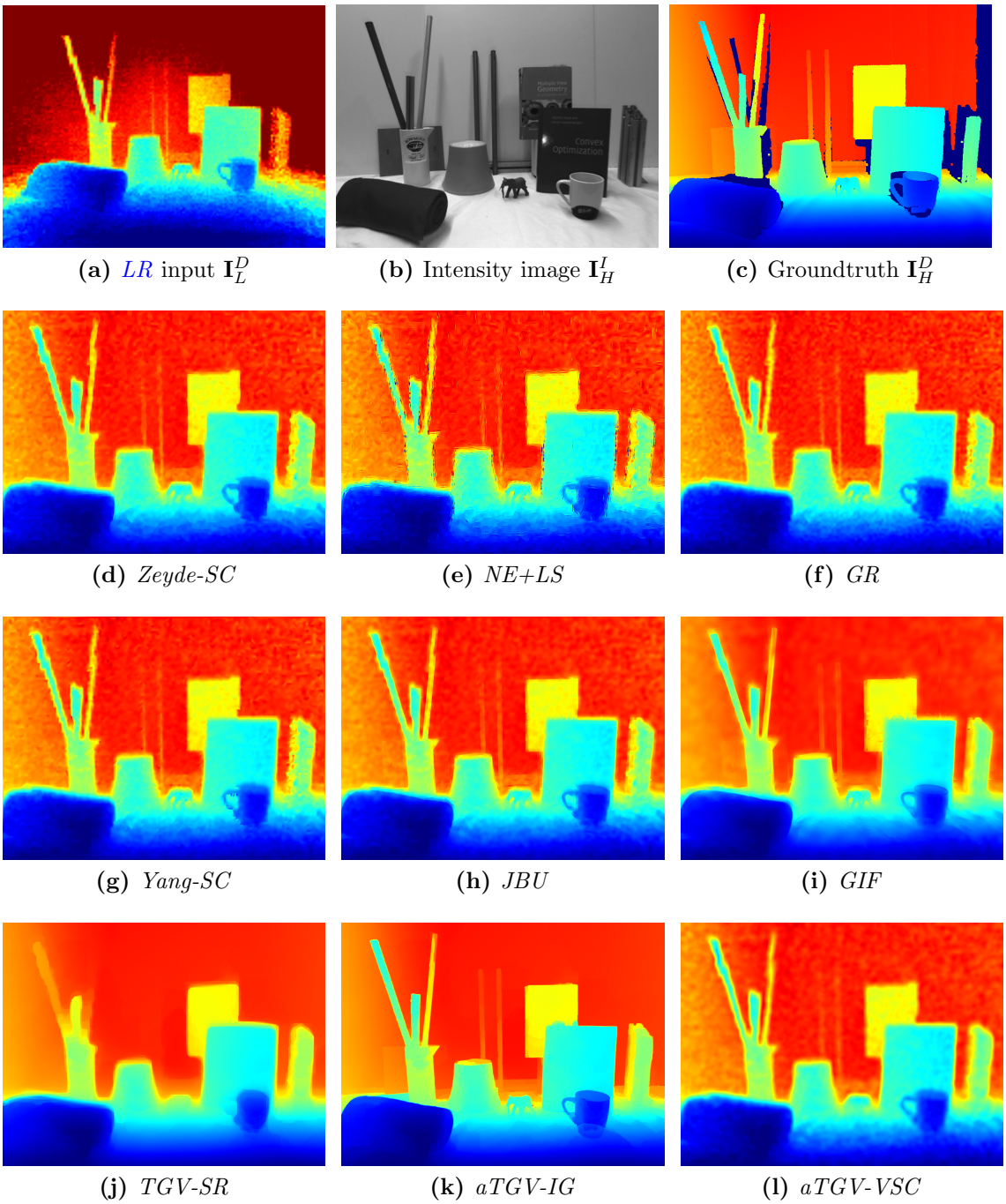


(g) *Yang-IG* **(h)** *Diebel-IG* **(i)** *Park-IG*



(j) *TGV-SR* **(k)** *aTGV-IG* **(l)** *aTGV-VSC*

Figure 4.12: Visual evaluation on the noisy Middlebury *Books* dataset for a magnification factor of $\times 16$. In the first row the input and Groundtruth is shown. In the following rows a selection of the best methods for interpolation, sparse codding and optimization is shown, while in the last row the results of the proposed methods are shown.



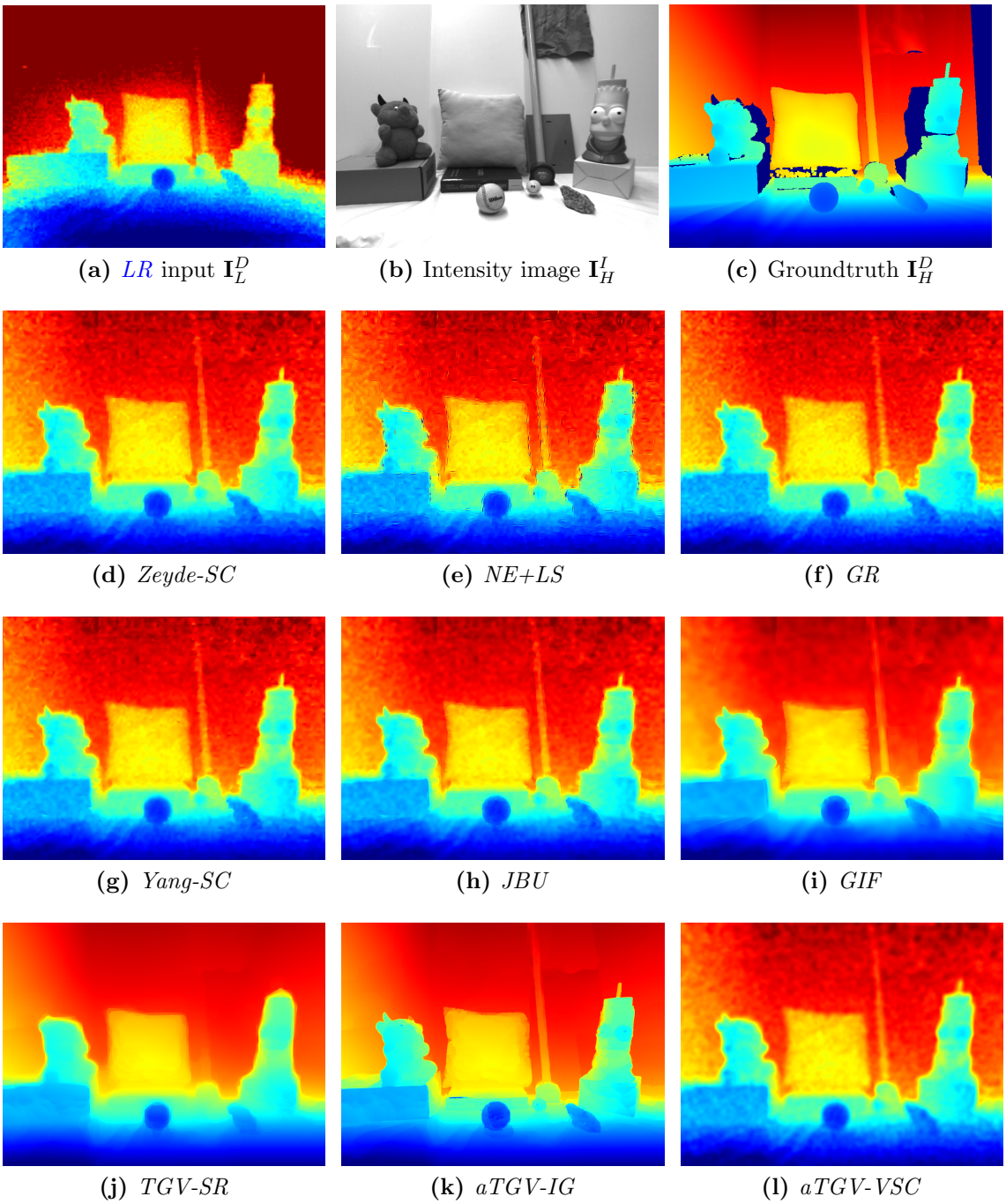
(a) *LR* input $\mathbf{I}_L^D$ **(b)** Intensity image $\mathbf{I}_H^I$ **(c)** Groundtruth $\mathbf{I}_H^D$

(d) *Zeyde-SC* **(e)** *NE+LS* **(f)** *GR*

(g) *Yang-SC* **(h)** *JBU* **(i)** *GIF*

(j) *TGV-SR* **(k)** *aTGV-IG* **(l)** *aTGV-VSC*

Figure 4.13: Visual evaluation of the *Books ToFMark* dataset. In the first row we show the *LR* input depth (enlarged), the input intensity image and the structured light groundtruth. In the middle rows *SoA SR* methods are shown. In the last row the results using our methods are depicted.



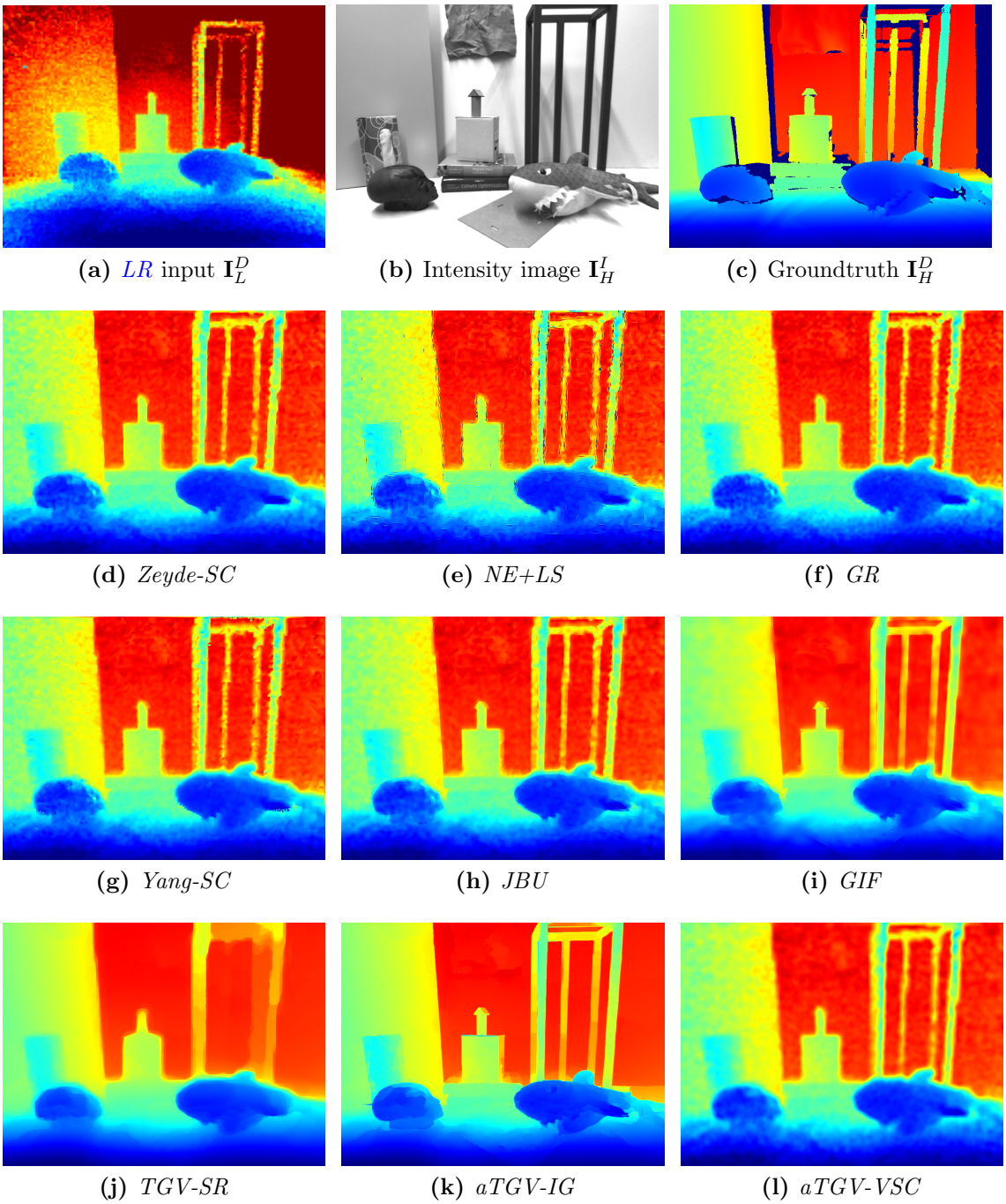
(a) *LR* input $\mathbf{I}_L^D$ **(b)** Intensity image $\mathbf{I}_H^I$ **(c)** Groundtruth $\mathbf{I}_H^D$

(d) *Zeyde-SC* **(e)** *NE+LS* **(f)** *GR*

(g) *Yang-SC* **(h)** *JBU* **(i)** *GIF*

(j) *TGV-SR* **(k)** *aTGV-IG* **(l)** *aTGV-VSC*

Figure 4.14: Visual evaluation of the *Devil ToFMark* dataset. In the first row we show the *LR* input depth (enlarged), the input intensity image and the structured light groundtruth. In the middle rows *SoA SR* methods are shown. In the last row the results using our methods are depicted.



(a) *LR* input $\mathbf{I}_L^D$ **(b)** Intensity image $\mathbf{I}_H^I$ **(c)** Groundtruth $\mathbf{I}_H^D$

(d) *Zeyde-SC* **(e)** *NE+LS* **(f)** *GR*

(g) *Yang-SC* **(h)** *JBU* **(i)** *GIF*

(j) *TGV-SR* **(k)** *aTGV-IG* **(l)** *aTGV-VSC*

Figure 4.15: Visual evaluation of the *Shark ToFMark* dataset. In the first row we show the *LR* input depth (enlarged), the input intensity image and the structured light groundtruth. In the middle rows *SoA SR* methods are shown. In the last row the results using our methods are depicted.

5 3D Scene Flow

Contents

In the last chapter we have shown how to improve Time of Flight (ToF) imaging by combining the depth output with other knowledge such as other sensors or learned dictionaries. But image enhancement is not limited to the images at a single time. Further, one can also improve the depth quality through the temporal domain by fusing acquisitions over time. Since most scenes are not static, the basis of a successful temporal fusion is the correct estimation of motion in time. In this chapter we show our approach to estimate motion using modern *ToF* cameras, for what this fusion can be used in general and especially how to fuse consecutive images to gain quality.

5.1 Introduction

The perception of motion describes the process of motion cognition in an observed area. It infers the speed and direction of elements in a scene based on visual inputs. It provides a rich source of information to understand the world and make decisions [125]. Although this process appears to be straightforward for a human it is very difficult to explain it in terms of neural processing and is proven to be a very hard problem from a computational perspective. It is studied in many disciplines such as neurology, neurophysiology, psychology, engineering and computer science. Although the field of research in this area seems endless, we are interested in the latter discipline in this thesis. For details on the biological process on motion perception we refer to [1] and [46].

In the last decades a lot of work has addressed pure two-dimensional flow, namely Optical Flow (OF). It describes the velocity field that represents the 3D motion of object points projected on a 2D image [54]. It describes the apparent motion of intensity patterns in an image by estimating a 2D movement vector in image space which links two points of two consecutive frames as shown by Verri and Poggio [121], and Horn and Schunk [48]. The optical flow field in an image is the velocity field that represents the 3D motion of object points across a 2D image [54]. Since the *OF* estimates only a 2D projection of 3D motion, it has problems to recover the true motion in space. Relatively simple 3D movements such as a translations towards the camera produce very complex 2D motions which are hard to estimate by current *OF* approaches.

In computer science we are mostly interested in predicting the motion in a sequence of acquisitions since the structure and 3D motion of objects are essential to characterize and understand a dynamic scene. While Structure from Motion (SfM) [85] on static scenes is well understood, non-rigid scenes still pose a challenging problem, commonly addressed as Scene Flow (SF). The applications for *SF* analysis range from driver assistance, surveillance, action recognition, tracking, segmentation, 3D reconstruction to camera pose estimation.

In contrast to measure the movement in the image space the estimation of 3D motion called *SF* has recently emerged [119]. The *SF* estimation aims to calculate the true motion of scene points in a metric space. A popular way to estimate *SF* through *OF* is to use a calibrated and synchronized multi-view setup to jointly solver for stereo and *OF*, as shown in [5, 52, 120, 124, 126].

With recent range sensor developments, direct depth measurements are a popular alternative to multi-view depth imaging. Such novel sensors e.g. Microsoft Kinect or Intel Gesture Camera already reached a sufficient level of accuracy and robustness to allow a wide usage in the mass market. With the help of these very affordable sensors it is no longer necessary to reconstruct the whole scene through a computationally expensive multi-view setup but directly access dense depth data from the sensor. This easily accessible depth information has brought major breakthroughs in many computer vision tasks such as reconstruction [76], human pose estimation [107] or object detection [109]. The idea of

using the depth cues for monocular *SF* estimation has also received increasing attention. Additionally using color information of the scene this problem is called *RGB-D SF*.

Although the depth cues from a single view can provide information about the scene motion, the acquisition setup poses new challenges for *SF* estimation. Additionally to the common *OF* problems such as the aperture problem and changing illumination, the noise, partially missing measurements and the limited resolution of the depth estimates cause new challenges.

In this thesis we propose a framework for dense dense *SF* estimation facing these challenges. We use a combination of depth and intensity data from the depth sensors. Our idea is to establish a *SF* framework, which is able to use a variety of complex data and regularization terms in a variational energy minimization approach.Originating from the classical *OF* warping on the pixel level, the motion in our *SF* framework is directly modeled in 3D. The warping in image space is calculated by the projection and back-projection of the 3D motion. This projective information is directly included in our optimization. In particular, we propose a method for the estimation of metric motion is space from a sequence of consecutive *RGB-D* acquisitions. The estimation method is formulated as a convex optimization problem which is solved by the primal-dual optimization scheme.

In this chapter we first give a broader overview of the approaches related to our work. From the general acquisition of the *RGB-D* images we derive the formulation of 3D motion in space. By projection of measured 2D points into 3D space we can can compare the intensity and depth input given the 3D motion. Under the constancy assumption in image space this comparison gives us an intensity and a depth data term, which builds the basis of our *SF* model, as shown in Section 5.3. Out of this basic terms we formulate convex optimization problem which shown in Section 5.4.

Although this basic model delivers sufficiently good results for *SF* estimation it still has problems when the illumination changes during the acquisition, or if low texture, low structure and noise appear in the acquisitions. To tackle these problems we show more robust variations of our data term in Section 5.5. In this section we first show some improvements on the pixel level like modeling the illumination changes of decompose only the textural part of the input. Second we propose how to include more robust data terms in our *SF* model which are calculating the image differences on a larger patch-level. In this context we show different variations of the patchwise constancy based on common matching terms. In Section 5.6 we show how the regularization term in the convex optimization can be improved in order to increase the quality of the *SF* result. Finally in Section 5.7 we show an extensive qualitative and quantitative evaluation of the proposed *SF* estimation in all its variants. It is further tested at different motions, on different levels of noise and to other recent State-of-the-Art (SoA) methods.

5.2 Related Work

Originating from the seminal works of Horn and Schunk [48] and Lucas and Kanade [64] a vast amount of work has been done on *OF* estimation. The main idea of *OF* estimation is to calculate the movements of pixes in a sequence of consecutive images. In the minimal case, the *OF* estimation calculates a two dimensional flow field $\mathbf{u}^{\mathrm{OF}}$ which measures the movement between two consecutive image acquisitions $\mathbf{I}_1$ and $\mathbf{I}_2$. This flow field is either estimated densely for every pixel in an image [48] or only at sparse image points as shown in [64]. Recently, Sun et al. [112] surveyed the different *OF* approaches and their principles.

In contrast to *OF* estimation, the calculation *SF* estimates the metric 3D motion in a space. The *SF* estimation for two consecutive acquisitions results in the three dimensional flow field $\mathbf{u}$. Hence, the goal of *SF* estimation is to captures the true metric motion in space instead of a 2D projection of 3D motion. The first definition of the terminology of *SF* was given by Vedula et al. [119, 120] to estimate the 3D motion from an image sequence in a calibrated multi-view setup. Following this approach, a lot of work has been done in the field of *SF* from multi-view intensity images, as shown in 5.2.1. With the recent availability of affordable depth sensors, methods for *SF* calculations from combined depth and intensity acquisitions (*RGB-D*) have emerged. In 5.2.2 we present an overview of the related work on *RGB-D* scene flow estimation.

5.2.1 Scene Flow from Multi-View

The goal in the field of *SF* from multi-view is to estimate the 3D motion field from a setup of two or more intensity cameras.

In the pioneering work of Vedula et al. [119] a fully calibrated multi-view setup is used to calculate local *SF* in a Lukas Kanade (LK) [64] framework. Based on the known *OF* the *SF* is derived knowing the exact surface model, knowing the pixel correspondences in the multi-view setup or by only knowing the pixel correspondences. The resulting *SF* is represented in a 3D array of voxels. In their work have shown that the *SF* estimation from multi-view is tightly connected to the 3D reconstruction. In the following works the 3D geometry is either calculated in a preprocessing step or together with the *SF* in a complete optimization. Huguet and Devernay [52] coupled dense stereo matching with *OF* estimation to recover dense *SF*. In their work they extend the dense *OF* estimation from Brox et al. [16] by adding constraints due to the epipolar geometry. Since the objective function from stereo and flow is very similar they propose to simultaneously reconstruct the disparity maps of two consecutive time-steps as well as the *OF* field for both cameras. In contrast to this work Wedel et al. [126, 128] proposed to partially decoupled the stereo depth and 2D motion estimation. This has the advantage that the method is independent from the stereo reconstruction method and runs in real-time. Valgaerts et al. [118] proposed an optimization framework that further estimates extrinsic parameters of the stereo setup in addition to depth and flow estimation. This allows for changes in

the relative pose of the stereo rig. To achive higher framerates Čech et al. [19] proposed a method to optimize depth and flow from sparse correspondence seeds and propagate them to their neighborhood. In all these studies the *SF* is formulated as 2D motion and depth. In contrast, Basha et al. [5] proposed a method to directly estimate the 3D *SF* and depth in a multi-view setup. In the optimization the smoothness is imposed directly on the 3D unknowns, while enforcing geometric consistency between the views. Instead of a regularization of the 3D motion Vogel et al. [123] proposed to penalize deviations from local rigidity of the motion to improve the robustness. Based on this work they proposed to model the scene as a collection of planar regions, each moving rigidly over time [124]. Based on an initial superpixel object segmentation the optimization jointly estimates 3D geometry, 3D motion vectors and updated superpixel boundaries. In [122] they extended this approach for multiple consecutive frames. By exploiting the consistency over time they show how both shape and 3D motion estimation can be improved.

5.2.2 Scene Flow from Depth and Intensity Data

With the recent developments of depth and intensity sensors packed in a *RGB-D* sensor, methods for *SF* calculations from *RGB-D* data have emerged. Similar to In contrast to *SF* from multi-view, these methods aim to directly estimate the 3D motion directly out of depth and intensity data.

The first work using *RGB-D* data was proposed by Spies et al. [110], where a global 2D *OF* estimation was enhanced by third channel for depth in a total least squares optimization. In there work a laser range finder is used for depth and the brightness of the reflected laser beam is used as intensity image. Lukins and Fisher [65] extended this approach by using to multiple color channels with an aligned depth depth image.

Hadfield and Bowden [39] proposed a local approach, where the *SF* calculation was modeled using a particle filter. To increase the density of the estimation scene particle estimation a ray resampling was proposed, where each particle is grouped into to the closest ray from the camera. An extension of this work was presented in [40] where the scene flow information is propagated through time to resolve ambiguities. This algorithm was applied to 3D hand tracking for sign language recognition. Due to the particle optimization possible oversmoothing of the flow field is avoided.

Similar, Quiroga et al. [88] proposed a method to directly calculate the *SF* in a *LK* framework. In their work the image flow is modeled as a function of 3D motion using the camera projection parameters. In [90] they embedded this model in a dense variational framework. The estimation of a dense flow field in a linear optimization scheme was proposed by Letouzey et al. [57]. In their approach a sparse set of correspondences is calculated through SIFT feature matching. This set is used as additional data term is the intensity constraint in the variational optimization to get a dense flow field. Gottfried et al. [36] proposed a method for depth camera calibration to estimate dense *OF* together with a depth flow estimation. Similar, Zhang et al. [138] combined a global energy

optimization and a bilateral filter to detect occlusions in a two-step framework. Using depth and intensity information Herbst et al. [44] showed how to generalize variational *OF* algorithms for *SF* calculation. They further show how *SF* aids object segmentation from motion. Similar to our work, Hornáček et al. [49] recently showed the advantages of estimating 3D motion directly through a patch matching in the point cloud. Unlike our method, they estimate this motion by a full rigid-body estimation for each patch using a *RGB-D* PatchMatch algorithm [10, 50]. This is especially useful for large motions. Following the stereo *SF* approach from Vogel et al. [124] Quiroga et al. [87] showed that a pure estimation of local and global rigid motions can be beneficial. In their work they model the *SF* through a field of locally rigid twists, which encourages piecewise smooth solutions of rigid body motions. An approach to jointly segment rigid objects and estimate their 3D motion was presented by Sun et al. [113]. From depth cues they create layer ordering information to handle occlusions and estimate a common 3D motion for each layer.

5.2.3 Discussion

Existing particle based approaches such as [39, 40] estimate *SF* on a sparse set of corresponding points. These approaches deliver only a dense flow field after interpolation. Other approaches such as [57] calculate local feature correspondences for depth images and a global flow estimation based on the intensity information separately. Hence, it will inevitably fail for wrong correspondence estimates. Our model builds on the success of global optimization methods as shown in [36, 44, 90, 138]. These methods estimate the flow through pixelwise brightness and depth constancy, where the motion is measured in pixels. In this thesis we present a *SF* estimation framework where the flow is directly modeled as metric motion in space defined by a projection and back-projection in the 3D space. We further show how the *SF* model can be extended above of the pixelwise constancy terms. We introduce a general framework for non-convex patchwise comparison terms such as the Ternary Census Transform (TCT) or the Normalized Cross Correlation (NCC). We further introduce a patchwise depth fidelity term which is directly calculated from the 3D point cloud by calculating the patchwise distance to the corresponding Closest Point (CP) estimates, similar to Iterative Closest Point (ICP). Introducing this new terms we show that the *SF* estimation gets more accurate in moderately structured or textured regions, more robust to acquisition noise and illumination changes. Recently methods using patchwise information such as [49] and [87] started using local patches in their data terms. Instead of the estimation of 3D flow vectors, these methods estimate the motion through the local translation and rotation in 3D. In [49] the fidelity of the local patches is calculated by a patch matching, which is useful for large motion, but is less capable of handling input noise or illumination changes.

For regularization most current methods use first order penalization with a squared ℓ_2 or approximations of the ℓ_1 norm like the Chabonnier norm. In contrast, we evalu-

ate higher order regularization terms with ℓ_1 penalizer to avoid oversmoothing and flow-flattening. Edge preserving properties and smooth transitions like rotations or non-rigid movements are still possible. Furthermore, we introduce different weighing terms for the regularization calculated from the input images. We point out benefits and drawbacks of all introduced terms in an extensive evaluation.

5.3 Scene Flow Model

The fundamental goal of dense 3D motion estimation is to calculate the metric motion of a scene. In our case we consider the flow as the movement of a 3D point $\mathbf{X}_t = (X, Y, Z)_t^{\mathrm{T}}$ in a scene over time t. Hence, the scene flow is defined as $\mathbf{u} = \frac{\partial \mathbf{X}}{\partial t} = (u_X, u_Y, u_Z)^{\mathrm{T}}$. When the scene is observed with a *RGB-D* camera over time a series of images $\mathbf{I}_t$ is acquired. This image series is divided in an intensity part $\mathbf{I}_t^I \in \mathbb{R}^\Omega \mapsto \mathbb{R}$ (assuming only gray scale) and a depth part $\mathbf{I}_t^D \in \mathbb{R}^\Omega \mapsto \mathbb{R}$, where Ω is the image space. The value in the depth image $\mathbf{I}_t^D(\mathbf{x}_t)$ at a given image pixel position $\mathbf{x}_t$ is given by the metric distance of the camera center to the observed scene point $\mathbf{X}_t$. Hence, the projection of a scene point $\mathbf{X}_t$ into the image space results in the image pixel position $\mathbf{x}_t = (x, y)_t^{\mathrm{T}}$. After a calibration of the camera intrinsics this projection is calculated by

$$\mathbf{x}_t^h = \mathbf{P}\mathbf{X}_t^h \tag{5.1}$$

as shown in Section 3.2. This movement in the 2D image space defined as $\mathbf{u}^{\mathrm{OF}} = \frac{\mathrm{d}\mathbf{x}}{\mathrm{d}t} = \left(u_x^{\mathrm{OF}}, u_y^{\mathrm{OF}}\right)^{\mathrm{T}}$ is well known as Optical Flow (*OF*). In Figure 5.1 the flow model is shown for two consecutive acquisitions $t = \{1, 2\}$.

As the image point $\mathbf{x}_t$ moves let us assume that its intensity (albedo, color) remains constant over time. This observation from the seminal works of Lukas and Kanade [64], and Horn and Schunk [48] builds the foundation of most visual *OF* and *SF* estimation methods, namely the *brightness constancy assumption*

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{I}_t^I(\mathbf{x}_t) = 0. \tag{5.2}$$

The basis for *OF* algorithms is the first order Taylor expansion of Eqn. (5.2) given by

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{I}_t^I(\mathbf{x}_t) = \frac{\partial \mathbf{I}_t^I(\mathbf{x}_t)}{\partial \mathbf{x}}\frac{\partial \mathbf{x}}{\partial t} + \frac{\partial \mathbf{I}_t^I(\mathbf{x}_t)}{\partial t} = 0, \tag{5.3}$$

where $\mathbf{u}^{\mathrm{OF}} = \frac{\mathrm{d}\mathbf{x}}{\mathrm{d}t}$ is the motion in the image space, $\nabla\mathbf{I}_t^I(\mathbf{x}_t) = \frac{\partial \mathbf{I}_t^I(\mathbf{x}_t)}{\partial \mathbf{x}}$ is the spacial image derivative and $\mathbf{I}_{\Delta t}^I(\mathbf{x}_t) = \frac{\partial \mathbf{I}_t^I(\mathbf{x}_t)}{\partial t}$ is the temporal image derivative. Evaluated at a initial *OF* field $\mathbf{u}_0^{\mathrm{OF}}$ the brightness constancy is expressed as classical intensity constraint [129]

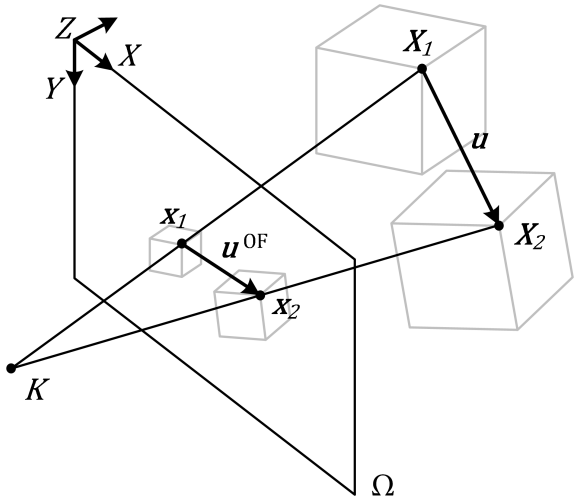


Figure 5.1: Geometric Flow Model. After a projection of the 3D scene points into the images the 3D flow $\mathbf{u}$ can be measured as 2D pixel movements $\mathbf{u}^{\mathrm{OF}}$.

given by

$$\rho_I(\mathbf{u}^{\mathrm{OF}}) = \mathbf{I}^I_{\Delta t} + \left(\nabla \mathbf{I}^I_t\right)^T \left(\mathbf{u}^{\mathrm{OF}} - \mathbf{u}^{\mathrm{OF}}_0\right), \tag{5.4}$$

using the first order Taylor expansion. Note that the first order Taylor expansion (linearization) is the convex underestimation of the function at point $\mathbf{u}$, as shown in Section 3.3.

This basic model with a variety of modifications has been applied to many *OF* estimation algorithms. Formulated as an energy minimization problem, the goal of *OF* estimation is to find a 2D flow field $\mathbf{u}^{\mathrm{OF}}$ which minimizes the above equation. The solution to this problem is ill-posed related to the *aperture problem*. It describes the ambiguity in the motion estimation when only a part of an object is visible (as seen through an aperture). The same problem arises when the object has a repetitive structure where the movement between frames can not be uniquely identified. The example shown in Figure 5.2 well describes the problem. Given two consecutive observations (a) and (b) it can not be exactly determined if the object is moving down, to the left or even in depth.

When dealing with *SF* the estimation has a very similar objective. The goal is to define a constancy term which is minimized for the 3D motion field $\mathbf{u}$, where we have the depth information as additional information. Equally to the projection from 3D to 2D space as shown in Eqn. (5.1) the scene points are given by the back-projection utilizing the acquired depth information. We define the camera center as the the world coordinate center. Hence, we do not have to rotate or translate the 3D points. As shown in Section 3.2, a 3D point $\mathbf{X}$ is calculated by multiplication of the normalized viewing ray at the image

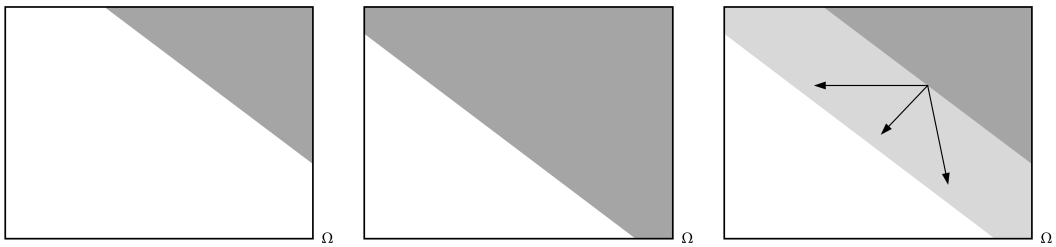


Figure 5.2: *Aperture Problem.* If only a part of an (untextured) object is visible the 2D motion direction is ambiguous.

point $\mathbf{x}$ with the scene distance. While the viewing ray is given by $K^{-1}x^h$ and the scene distance is given by the depth image pixel $\mathbf{I}^D(\mathbf{x})$ the projected 3D point is given by

$$\mathbf{X}_t = \frac{K^{-1}\mathbf{x}_t^h}{\|K^{-1}\mathbf{x}_t^h\|_2}\mathbf{I}_t^D(\mathbf{x}_t), \tag{5.5}$$

for each image pixel $\mathbf{x}$ and time t. Let us consider the flow between two consecutive *RGB-D* image acquisitions $t = \{1, 2\}$. Given projected scene points, the the connection in 3D space is given by

$$\begin{aligned} \mathbf{X}_2 &= \mathbf{X}_1 + \mathbf{u}, \\ \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}_2 &= \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}_1 + \begin{bmatrix} u_X \\ u_Y \\ u_Z \end{bmatrix}. \end{aligned} \tag{5.6}$$

Back-projected into the image space the motion is expressed as the warping function, as shown in Figure 5.3. Following the flow definition from Eqn. (5.6) the movement in image space is calculated as

$$\mathbf{x}_2^h = W(\mathbf{X}_1, \mathbf{u}) = \frac{K\left(\mathbf{X}_1 + \mathbf{u}\right)}{Z_1 + u_Z}. \tag{5.7}$$

Including the projection from Eqn. (5.5) the pixel to pixel warping from one frame to the next is given by

$$\mathbf{x}_2^h = W(\mathbf{x}_1, \mathbf{u}) = \frac{K\left(\frac{K^{-1}\mathbf{x}_1^h}{\|K^{-1}\mathbf{x}_1^h\|_2}\mathbf{I}_1^D(\mathbf{x}_1) + \mathbf{u}\right)}{\mathbf{I}_1^D(\mathbf{x}_1) + u_Z}. \tag{5.8}$$

This projection and back-projection depicts the geometric relationship between two consecutive images. With our definition of warping between images it is possible to use standard optical flow constraints and optimize for the 3D flow. As a result, the classical

intensity constraint from Eqn. (5.4) becomes

$$\rho_I(\mathbf{x}_1, \mathbf{u}) = \mathbf{I}^I_{\Delta t}(\mathbf{x}_1, \mathbf{u}_0) + \left(\nabla \mathbf{I}^I_2(W(\mathbf{x}_1, \mathbf{u}_0))\right)^T \frac{\partial W(\mathbf{x}_1, \mathbf{u}_0)}{\partial \mathbf{u}_0} (\mathbf{u} - \mathbf{u}_0)\,, \tag{5.9}$$

evaluated at the initial stationary flow $\mathbf{u}_0$, where $\frac{\partial W(\mathbf{x}_1, \mathbf{u}_0)}{\partial \mathbf{u}_0}$ is the Jacobian matrix of the warping function W. From now on we will write $W_0 = W(\mathbf{x}_1, \mathbf{u}_0)$ for brevity. The temporal derivative in the 3D *SF* case is defined as

$$\mathbf{I}^I_{\Delta t}(\mathbf{x}_1, \mathbf{u}_0) = \mathbf{I}^I_2(W_0) - \mathbf{I}^I_1(\mathbf{x}_1). \tag{5.10}$$

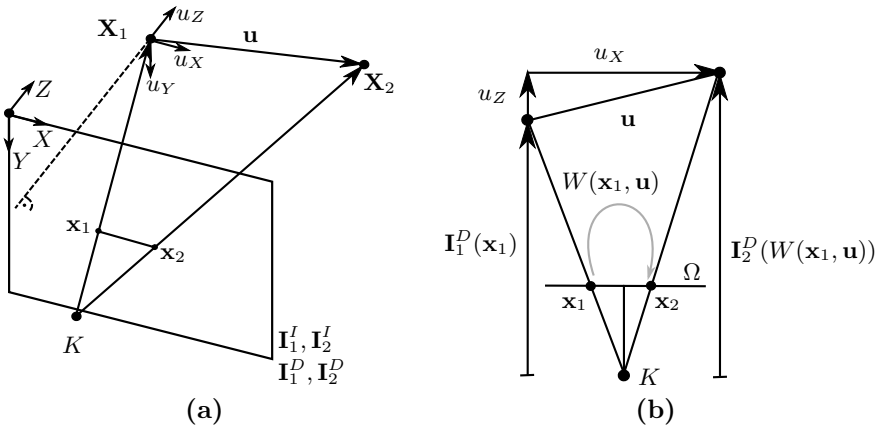


Figure 5.3: Flow Geometry. A scene point $\mathbf{X}_1$ acquired in the first frame moves to $\mathbf{X}_2$ in the second frame, as shown in (a). This 3D movement between two acquisitions is defined as flow $\mathbf{u}$. The projection in the image space from point $\mathbf{x}_1$ to $\mathbf{x}_2$ is defined as the warping $W(\mathbf{x}_1, \mathbf{u})$. A projection of (a) in Y direction is shown in (b).

Although the *SF* constraint in the intensity space can be optimized for the motion field the solution is very underdetermined in $\mathbf{u}$. Hence it is necessary to utilize the depth information acquired by the *RGB-D* camera as additional information. Similar to the intensity constraint, the temporal flow derivative in depth space in consecutive acquisitions is expressed as

$$\mathbf{I}^D_{\Delta t}(\mathbf{x}_1, \mathbf{u}_0) = \mathbf{I}^D_2(W_0) - \mathbf{I}^D_1(\mathbf{x}_1) - u_Z. \tag{5.11}$$

This connection is depicted in Figure 5.3b. The difference to the temporal intensity derivative is that for depth image space the motion in depth u_Z is included in the formulation. Following the temporal derivative, the basic depth constraint of our model is given by the

first order Taylor expansion of Eqn. (5.11):

$$\rho_D(\mathbf{x}_1, \mathbf{u}) = \mathbf{I}^D_{\Delta t}(\mathbf{x}_1, \mathbf{u}_0) + \left(\nabla \mathbf{I}^D_2(W_0)\right)^T \frac{\partial W_0}{\partial \mathbf{u}_0} (\mathbf{u} - \mathbf{u}_0) \,. \tag{5.12}$$

This function defines the energy functional given two consecutive depth images. Depth images usually contain less structural information (high frequency parts) compared to intensity images. In an energy optimization this makes it harder to get a unique solution. Hence it is difficult to estimate the *SF* from depth acquisitions alone.

Given the intensity constraint Eqn. (5.9) and the depth constraint Eqn. (5.12) we have established the basic *data fidelity* terms of our model. As aforementioned, the *SF* model given our two constraints is an inverse problem and hence is ill-posed in its solution. The *SF* problem from *RGB-D* data can be solved by different kinds of algorithms. For *OF* they can be mainly divided into local or global models.

Local models, as originated from the well-known *LK* model [64], estimate the image motion at sparse positions in the consecutive images (points of interest). The flow vectors are determined by comparing the intensity values of a small local neighborhood around the pixel within consecutive images. The final motion of each point of interest is estimated by solving an over-determined equation system by minimizing the squared error. Because the *LK* model is only able to estimate small displacements it is often used in a coarse-to-fine approach, where the flow is solved in an image pyramid of subsampled input images.

Although local models estimate the motion at high frame-rates and are widely used in literature they only deliver a very sparse flow field and totally miss in in homogeneous image regions. In contrast, global models estimate the motion for every pixel in an image, as shown in the seminal work of Horn and Schunk [48]. Additionally to the data constraints a *regularization term* is added to the optimization. This regularization term includes some prior information of smoothness and noise into the model. Hence, in a global model, the flow in homogeneous regions is interpolated from surrounding estimates while local models would end up with poorly conditioned estimates in these regions. The result of the global models is a dense flow field which gives a flow vector for every pixel in the image. In our work we used a variational formulation to estimate the dense flow field in a global model, which is described in the next section.

5.4 Variational Model

The fundamental goal of our estimation task is to estimate the metric 3D motion field of a scene given a sequence of consecutive *RGB-D* frames from a stationary camera. The *SF* estimation in our approach is formulated as a general variational problem, following [97]. The standard variational minimization is given by

$$\min_{\mathbf{u}} \mathcal{D}(\mathbf{u}) + \mathcal{R}(\mathbf{u}). \tag{5.13}$$

The term $\mathcal{D}$ is the *data term* which penalizes deviations from the defined cost functions given a flow field $\mathbf{u}$. Since the variational model is ill-posed we add constraints on noise and smoothness to our model. This is expressed as the *regularization term* $\mathcal{R}$ which penalizes deviations of neighboring estimates and hence is used for denoising and interpolation of the global flow field.

Both the regularization and the data function are penalized by a defined norm. In the simplest case the penalty norm $\psi(f)$ in $\mathcal{D}$ and $\mathcal{R}$ is the ℓ_2-norm. This quadratic penalization has shown some success in the early *OF* [48] or *SF* approaches [119] as it is convex and its derivative can be easily calculated. This makes the whole model suitable for all variations of gradient descend algorithms. On the downside it is not robust to outliers in the constancy terms and flow estimates. A popular choice to increase this robustness to outliers is to use the ℓ_1-norm given by $\psi(f) = |f|$ or the Charbonnier-norm $\psi(f) = \sqrt{f^2 + \varepsilon^2}$ which is a smooth approximation of ℓ_1 norm. Recently, some optimization techniques have proposed that use non-convex penalty functions such as the ℓ_p-norm for $0 < p < 1$. These non-convex terms in the optimization have the advantage that finer details in the solution can be better reconstructed. But, since these function is non-convex, the problem is much harder to optimize and can only be solved through alternating optimization approaches by lifting of the non-convex parts, as shown in [78, 93].

One big advantage of the variational model is that it is capable to include all kinds of different data and regularization terms into the model, as long as they are convex or can be linearized in a reasonable way. In the following we propose how the basic *SF* constraints are included in a variational model using the ℓ_1 norm as penalization and the well known Total Variation (TV) as regularization, forming the *TV*-ℓ_1 optimization for *SF* estimation. We further show several new *SF* models which are based on the intensity and depth constraints we have stated above. In Section 5.5 we propose various *data terms* for *SF* estimation. The goal is to find a data term which best models errors of the flow field given the input. Further, in Section 5.6 we show different techniques for regularization of the flow field to achieve the best possible results in terms of accuracy and robustness.Since our model sticks to the linearization of the flow field the iterative update are only valid in a small neighborhood of the stationary point $\mathbf{u}_0$ hence we follow traditional *OF* and perform a coarse-to-fine warping scheme of the optimization, which is shown in Section 5.4.2.

5.4.1 TV-ℓ_1 Scene Flow

The TV-ℓ_1 *SF* model the basis of our *SF* estimation. In this section we show how the model is defined in detail and how it is optimized using the primal-dual formulation. While the traditional *OF* model of Horn and Schunk [48] uses only a quadratic norm for regularization, the TV-ℓ_1 model is composed of a *TV* regularization together with a ℓ_1 norm in the data term. The *TV* is a very common regularization term in variational models for computer vision. It aims to minimize the first order gradient in the optimization domain. Hence, the energy term increases if neighboring pixels in the flow field differ from

each other. There are two possible definitions of the *TV* regularization term which are

$$\mathcal{R}(\mathbf{u}) = \int_\Omega \|\nabla \mathbf{u}(\mathbf{x})\|_1 \mathrm{d}x \quad \text{and} \quad \mathcal{R}(\mathbf{u}) = \sum_{i\in\{X,Y,Z\}} \int_\Omega |\nabla u_i(\mathbf{x})| \mathrm{d}\mathbf{x}. \tag{5.14}$$

While the first definition minimizes the flow gradient vector by the ℓ_1 norm, the second definition penalizes each gradient direction separately. Since it is likely that objects move in more than one coordinate direction the first model is better suited in the general case. But it has the disadvantage that it weights all axis equally which could lead to errors if the directions have different metric resolutions.

In our model we have chosen the first definition since motions in different coordinate directions are likely to co-appear. Hence, the flow in the different directions is connected and can be penalized together.

Our basic *SF* model is composed of the TV regularization together with the ℓ_1 norm of the *SF* constraints we have established in the previous section. The combined model is given by

$$\min_{\mathbf{u}} \int_\Omega \|\nabla \mathbf{u}(\mathbf{x})\|_1 \mathrm{d}\mathbf{x} + \lambda_D \int_\Omega |\rho_D(\mathbf{x},\mathbf{u})| \mathrm{d}\mathbf{x} + \lambda_I \int_\Omega |\rho_I(\mathbf{x},\mathbf{u})| \mathrm{d}\mathbf{x}\,, \quad \text{where} \tag{5.15}$$

$$\rho_D(\mathbf{x}_1,\mathbf{u}) = \mathbf{I}^D_{\Delta t}(\mathbf{x}_1,\mathbf{u}_0) + \left(\nabla \mathbf{I}^D_2(W(\mathbf{x}_1,\mathbf{u}_0))\right)^T \frac{\partial W(\mathbf{x}_1,\mathbf{u}_0)}{\partial \mathbf{u}_0}(\mathbf{u}-\mathbf{u}_0) \quad \text{and} \tag{5.16}$$

$$\rho_I(\mathbf{x}_1,\mathbf{u}) = \mathbf{I}^I_{\Delta t}(\mathbf{x}_1,\mathbf{u}_0) + \left(\nabla \mathbf{I}^I_2(W(\mathbf{x}_1,\mathbf{u}_0))\right)^T \frac{\partial W(\mathbf{x}_1,\mathbf{u}_0)}{\partial \mathbf{u}_0}(\mathbf{u}-\mathbf{u}_0)\,. \tag{5.17}$$

where optimizer $\mathbf{u} \in \mathbb{R}$ is the metric 3D *SF*, and the scalars $\lambda_D, \lambda_I \in \mathbb{R}$ are parameters to weight each data term. This model is robust to gross outliers in the intensity constraints due to the ℓ_1 regularization of the data terms, which may come from wrong pixel correspondences. Furthermore, it preserves sharp discontinuities due to the TV regularization. This model has been extensively used in other computer vision problems such as denoising or *OF* estimation, as shown in [21, 136].

In the following we explain how to optimize the above model using the primal-dual scheme introduced in Section 3.4.3. We discretize the continuous images space into a regular Cartesian grid of size $M \times N$, hence $\Omega \mapsto \mathbb{R}^{M\times N}$. The discrete minimization problem is given by

$$\min_{\mathbf{u}} \sum_{\mathbf{x}\in\mathbb{R}^{M\times N}} \|\nabla \mathbf{u}(\mathbf{x})\|_1 + \lambda_D \sum_{\mathbf{x}\in\mathbb{R}^{M\times N}} |\rho_D(\mathbf{x},\mathbf{u})| + \lambda_I \sum_{\mathbf{x}\in\mathbb{R}^{M\times N}} |\rho_I(\mathbf{x},\mathbf{u})|. \tag{5.18}$$

The first step is to transform the convex but non-smooth model into a convex-concave saddle-point problem, defined as

$$\min_{\mathbf{u}} \max_{\mathbf{p},\mathbf{q_D},\mathbf{q_I}} \langle \mathbf{p}, \nabla \mathbf{u}\rangle_{P_\mathbf{p}} + \lambda_I \langle \mathbf{q}_D, \rho_D(\mathbf{u})\rangle_{Q_D} + \lambda_I \langle \mathbf{q}_I, \rho_I(\mathbf{u})\rangle_{Q_I}, \tag{5.19}$$

where the dual variables $\mathbf{p}$, $\mathbf{q}_D$ and $\mathbf{q}_I$ are introduced. The convex sets for the dual variables result in

$$\begin{aligned} P_{\mathbf{p}} &= \left\{\mathbf{p}\colon \Omega \to \mathbb{R}^{6MN} |\ \|\mathbf{p}(i,j)\| \leq 1\right\}, \\ Q_D &= \left\{\mathbf{q}_D\colon \Omega \to \mathbb{R}^{MN} |\ -1 \leq q_D(i,j) \leq 1\right\}, \\ Q_I &= \left\{\mathbf{q}_I\colon \Omega \to \mathbb{R}^{MN} |\ -1 \leq q_I(i,j) \leq 1\right\}, \\ i &= 1,...,M, j = 1,...,N. \end{aligned} \tag{5.20}$$

This primal-dual problem is efficiently optimized by three consecutive steps in each iteration. First, the dual problem from Eqn. (5.19) is maximized by gradient ascend. Second, the primal problem is minimized by gradient ascend and third, the primal variable is refined in an extrapolation step (overrelaxation) which accelerates the convergence. These three steps together form the primal dual optimization scheme of the above problem and result in

$$\begin{cases} \mathbf{p}^{n+1} = \mathcal{P}_{P_{\mathbf{p}}}\left\{\mathbf{p}^n + \sigma_{\mathbf{p}} \nabla \bar{\mathbf{u}}^n\right\} \\ \mathbf{q}_D^{n+1} = \mathcal{Q}_{Q_D}\left\{\mathbf{q}_D^n + \sigma_D \lambda_D \left(I_{\Delta t}^D(\mathbf{x}_1, \mathbf{u}_0) + \nabla I_2^D(W_0) \frac{\partial W_0}{\partial \mathbf{u}_0}(\mathbf{u}^n - \mathbf{u}_0)\right)\right\} \\ \mathbf{q}_I^{n+1} = \mathcal{Q}_{Q_I}\left\{\mathbf{q}_I^n + \sigma_I \lambda_I \left(I_{\Delta t}^I(\mathbf{x}_1, \mathbf{u}_0) + \nabla I_2^I(W_0) \frac{\partial W_0}{\partial \mathbf{u}_0}(\mathbf{u}^n - \mathbf{u}_0)\right)\right\} \\ \mathbf{u}^{n+1} = \mathbf{u}^n - \tau_{\mathbf{u}}\left(-\nabla^{\mathrm{T}} {\mathbf{p}_{\mathbf{u}}}^{n+1} + \lambda_D \left(\nabla I_2^D(W_0) \frac{\partial W_0}{\partial \mathbf{u}_0} - \mathbf{t}\right) \mathbf{q}_D^{n+1} \right. \\ \qquad\qquad \left. + \lambda_I \nabla I_2^I(W_0) \frac{\partial W_0}{\partial \mathbf{u}_0} \mathbf{q}_I^{n+1}\right) \\ \bar{\mathbf{u}}^{n+1} = \mathbf{u}^{n+1} + \theta(\mathbf{u}^{n+1} - \bar{\mathbf{u}}^n) \end{cases}, \tag{5.21}$$

for every iteration $n > 0$. At $n = 0$ the primal and dual variables are initialized with zeros. The newly introduced vector $\mathbf{t} = [0, 0, 1]^T$ which is the result of the derivative $\frac{\partial u_Z}{\partial \mathbf{u}}$. The operators $\mathcal{P}_{P_{\mathbf{p}}}$, $\mathcal{Q}_{Q_I}$ and $\mathcal{Q}_{Q_D}$ are point-wise Euclidean projections onto the convex sets $P_{\mathbf{p}}$, Q_I and Q_D and result in

$$(\mathcal{P}_{P_{\mathbf{p}}}(\tilde{\mathbf{p}}))(i,j) = \left\{\frac{\tilde{\mathbf{p}}(i,j)}{\max\{1, \|\tilde{\mathbf{p}}(i,j)\|\}}\right\}, \tag{5.22}$$

$$(\mathcal{Q}_{Q_D}(\tilde{q}_D))(i,j) = \min\left\{1, \max\{-1, \tilde{q}_D(i,j)\}\right\}, \tag{5.23}$$

$$(\mathcal{Q}_{Q_I}(\tilde{q}_I))(i,j) = \min\left\{1, \max\{-1, \tilde{q}_I(i,j)\}\right\}, \tag{5.24}$$

$$\forall i = 1,...,M, j = 1,...,N. \tag{5.25}$$

The timesteps of the optimization are given by the Lipschitz boundaries, defined as

$$\sigma_{\mathbf{p}} = \frac{\sigma}{2}, \quad \sigma_I = \sigma_D = \sigma, \quad \tau_{\mathbf{u}} = \frac{\tau}{4 + \lambda_D |\nabla I_2^D(W_0) \frac{\partial W_0}{\partial \mathbf{u}_0} - \mathbf{b}| + \lambda_I |\nabla I_2^I(W_0) \frac{\partial W_0}{\partial \mathbf{u}_0}|}, \tag{5.26}$$

s.t. $\tau\sigma \leq 1$. The overrelaxation stepsize is set to $\Theta = 1$.

In order to optimize the flow constraints we have introduced a first order Tailor expansion on the simple intensity and depth constraints. Since this linearization is only

valid in a small region around $\mathbf{u}_0$ we restrict the flow update in each iteration such that the change from the stationary point $\mathbf{u}_0$ is bounded by $\mathbf{u}_{\max} \geq \|\mathbf{u} - \mathbf{u}_0\|_2$. Although this linearization makes it possible to optimize such a non-convex problem the downside of this linearization is that it only covers motions smaller than $\mathbf{u}_{\max}$. To overcome this problem two main techniques have become popular in literature, namely warping and the coarse-to-fine optimization which are explained in the next section.

5.4.2 Warping and Coarse-to-Fine Optimization

The constancy equation of optical and scene flow estimation is non-convex. In order to globally minimize for the constancy formulation, it is linearized at an initial operating point. In our scene flow formulation this results in the depth and intensity constraints. Through this approximation by the Taylor expansion the problem becomes convex and a global minimum of the function can be estimated. The main disadvantage of this linearization is that not the original problem but the approximated problem gets minimized. The more the solution differs from the operating point the more it also differs from the original problem. As a result the estimated flow in this way is only valid for small variations from the operating point. To estimate large displacements two main strategies have emerged, namely the *iterative warping* and the *coarse-to-fine optimization*. In *SoA* methods these two techniques are often combined. In our estimation procedure we followed this combination.

At the *iterative warping* scheme a warping is applied after a fixed number of iterations. In other words, after m iterations the operation point is updated by setting $\mathbf{u} = \mathbf{u}_0$. In our *SF* formulation this results in an update of the warped second *RGB-D* frame $(\mathbf{I}_2^D, \mathbf{I}_2^I)$ and consequently an update of the spatial and temporal derivatives $(\mathbf{I}_{\Delta t}, \nabla\mathbf{I}_2 \frac{\partial W(\mathbf{x}_1, \mathbf{u}_0)}{\partial \mathbf{u}_0})$ after each *warp*. The whole optimization routine is further embedded into a *coarse-to-fine* framework as shown in [9, 16, 136] for *OF* estimation. In this framework the optimization is solved in a pyramid of a defined number of levels L starting from the coarsest. The image pyramids are employed with a fixed downsampling factor of $\nu \in R|(0, 1)$ between each level. Starting at the coarsest level $l = L$ the flow is initialized with $\mathbf{u}^L = \mathbf{u}_0^L = \mathbf{0}$. After the optimization for a fixed number of iterations and warps in each level the result is propagated to the next finer level by bicubic interpolation, where it is used as initialization. Additionally the camera matrix is updated in each level according to the downsampling factor by $K_l = K \operatorname{diag}\left([\nu^l, \nu^l, 1]\right)$, $\forall l = \{0, \ldots, L\}$. Subsequently, the result of the finest level $l = 0$ is the final result of the *SF* estimation. The whole coarse-to-fine pyramid propagation is shown in Figure 5.4. In our experiments we have shown that a bilinear interpolation delivers the best results. The warping and coarse-to-fine propagation helps our optimization to estimate larger displacements and avoids to get stuck in local minima when the displacement of pixels is bigger than the discretization level.

In contrast to our approach other techniques have been proposed to estimate larger displacements and to refine the estimated solution. In [3] Amiaz et al. have shown that

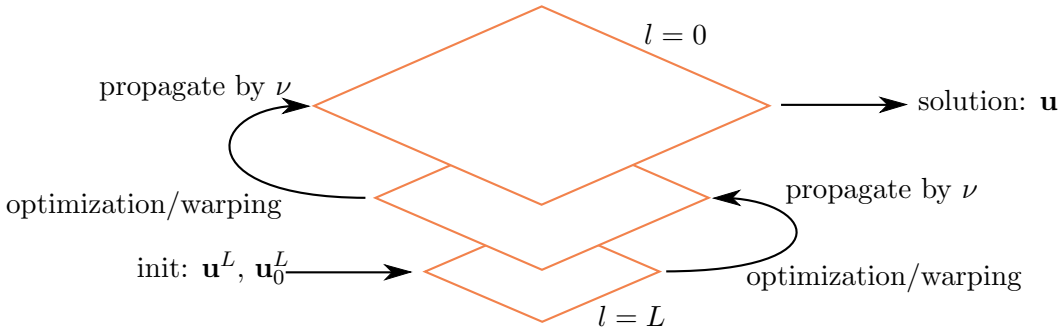


Figure 5.4: Coarse to fine optimization. Starting at the coarsest level $l = L$ the primal-dual optimization is solved for a fixed number of iteration and then propagated to the next level, where the output of level 0 is the solution.

it can be beneficial to propagate the pyramid to higher levels than the original image resolution in the sense of *course-to-overfine*. Contrary, Letouzey et al. [57] have proposed to first calculate a sparse set of flow vectors by SIFT feature matching in the intensity images before optimizing the flow constraints.

5.5 Data Terms Robust to Illumination Changes and Noise

In our variational optimization the data term $\mathcal{D}$ is the term which connects input data to the optimizer, and measures the, so-called, *data fidelity*. The data fidelity is penalized by a penalty function. In our case the input data is a set of two *RGB-D* acquisitions, the optimizer is the 3D flow field and the cost function connecting both is composed of the *SF* constraints. In the Section 5.3 the intensity and the depth image constraints are shown, which form the basis of the data term. In the optimal case the temporal and spatial derivative in our model are minimal and hence the intensity and depth constraints are sufficient for a successful *SF* estimation. This optimality is given when both the intensity and depth images are noise-free, without any illumination changes or reflections and with enough texture/structure to determine the brightness and depth constancy in every pixel in the image. In practice this is not the case. On the one side, the scene contains some reflective or partly transparent objects, the illumination changes between the acquisitions and shadows are present. On the other side, the acquired images contain acquisition noise and often also blur. In the following we will show different variants to improve the data term in terms of quality and robustness.

We separate the following methods into pixel-wise, where the constancy is measured on the pixel level, and patchwise, where the constancy between images is measured on a larger region around each point.

5.5.1 Pixel-Wise Constancy Assumption

In this section we introduce models for improved data terms that rely on the pixel-wise constancy assumption. Hence, we only compare the two *RGB-D* frames on the pixel level. In this section we focus on improving the pixelwise intensity constraint to get more robust against changes in the illumination, reflection effects and shadows. Compared to constraints based on a patch level these pixelwise models are much faster in terms of computational complexity but have still problems to deal with acquisition noise. In the following we will show different pixel-wise models and amendments to the traditional constraints to increase the quality and robustness of the *SF* estimation.

5.5.1.1 Modeling Illumination Effects

In the brightness constancy assumption which we have stated at the beginning of this chapter it is assumed that the illumination does not change between the acquisitions. Hence even the slightest change in the gray values will violate this assumption. Since the early days of motion estimation researchers have discovered that the simple pixel-wise brightness assumption does not hold over time, as shown in [71] or [121]. One approach is to model the changes in illumination in the flow constraints. In this context earlier works on *OF* estimation modeled the brightness changes as an additive or multiplicative compensation mask to the constraint. In our case we follow the work of Werlberger [129] for *OF* estimation to add an additional optimizer to the intensity constraint which models not only the amount of brightness that changes between the images but also upcoming shadows (lowering brightness) or reflections (increasing brightness). An example of this brightness compensation is shown in Figure 5.5.

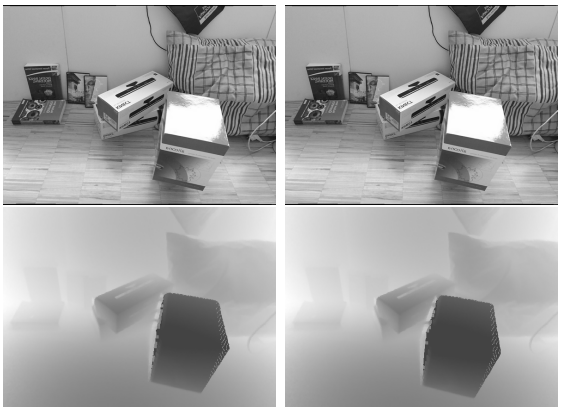
In this model the compensation term is defined as an additional factor directly inside the *SF* intensity constraint. The *brightness compensation factor* is modeled as a separate optimizer $\mathbf{b}\colon \mathbb{R}^{\Omega} \mapsto \mathbb{R}$ which is estimated together with the flow such that the brightness constancy including the compensation factor is given by

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{I}_t^I(\mathbf{x}_t) = -\gamma\mathbf{b} \tag{5.27}$$

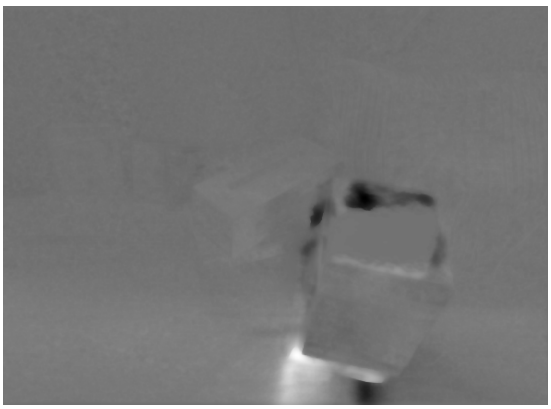
which states that the brightness constancy is a constant map, as shown in [129]. In our *SF* model, we include this compensation map into the intensity constraint which changes to

$$\rho_I(\mathbf{x}_1, \mathbf{u}, \mathbf{b}) = \mathbf{I}_{\Delta t}^I(\mathbf{x}_1, \mathbf{u}_0) + \left(\nabla \mathbf{I}_2^I(W(\mathbf{x}_1, \mathbf{u}_0))\right)^T \frac{\partial W(\mathbf{x}_1, \mathbf{u}_0)}{\partial \mathbf{u}_0}(\mathbf{u} - \mathbf{u}_0) + \gamma\mathbf{b}, \tag{5.28}$$

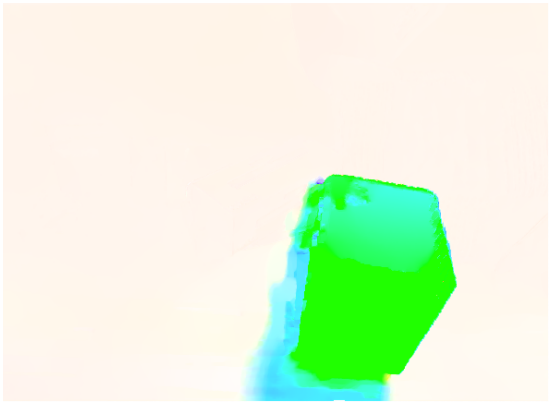
where the scalar $\gamma \geq 0 \in \mathbb{R}$ specifies the degree of compensation in the model. Larger values of γ allow larger brightness variations and vice versa. If γ is set to zero the compensation model is equal to the classical intensity constraint. Note that if the influence is set too high the quality can deteriorate since valuable information can be compensated as well.



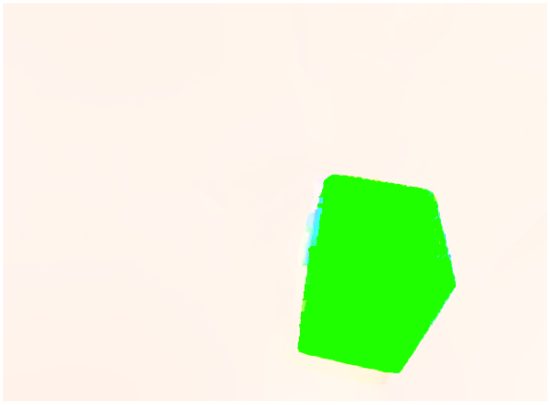
(a) Depth and Intensity input $\mathbf{I}_1$ / $\mathbf{I}_2$



(b) Compensation Model $\mathbf{b}$



(c) Flow without compensation



(d) Flow with compensation

Figure 5.5: Modeling Illumination Effects by a compensation model. The compensation model reduces the influence of brightness changes between the image acquisitions. Hence, changes in reflection, ambient light, shadows and occlusions are compensated in the *SF* estimation. The scene is acquired with a Microsoft K4Wv2 sensor.

To avoid that the compensation field $\mathbf{b}$ contains important texture information we enforce it to be smooth. Hence, we introduce a penalization on the gradient of $\mathbf{b}$ normed by the Huber-norm, as introduced in Section 3.1. This penalizer ensures a smooth compensation model where additional sharp boundaries can be modeled such that shadows or partial reflective parts can be modeled. Since the compensation model is only a linear addition of a convex function to the *SF* model it is still convex and can be optimized in the same way as the traditional model. The complete minimization problem from Section 5.4.1 is changed to

$$\min_{\mathbf{u}} \sum_{\mathbf{x} \in \mathbb{R}^{M \times N}} \left(\|\nabla \mathbf{u}(\mathbf{x})\|_1 + \lambda_D |\rho_D(\mathbf{x}, \mathbf{u})| + \lambda_I |\rho_I(\mathbf{x}_1, \mathbf{u}, \mathbf{b})| + \|\nabla \mathbf{b}(\mathbf{x})\|_\varepsilon \right) \tag{5.29}$$

and hence, the convex-concave saddle-point problem of the above model is given by

$$\min_{\mathbf{u},\mathbf{c}} \max_{\mathbf{p},\mathbf{q}_D,\mathbf{q}_I,\mathbf{r}} \langle \mathbf{p}, \nabla \mathbf{u} \rangle_{P_\mathbf{p}} + \lambda_I \langle \mathbf{q}_D, \rho_D(\mathbf{u}) \rangle_{Q_D} + \lambda_I \langle \mathbf{q}_I, \rho_I(\mathbf{u}, \mathbf{b}) \rangle_{Q_I} + \langle \mathbf{r}, \nabla \mathbf{b} \rangle_{P_\mathbf{r}} - \varepsilon \frac{\|\mathbf{r}\|_2^2}{2}, \tag{5.30}$$

where $\mathbf{r}$ is the primal variable for the compensation gradient and maps onto the convex set $P_\mathbf{r}$ given by

$$P_\mathbf{r} = \left\{ \mathbf{r} \colon \Omega \to \mathbb{R}^{2MN} \middle| \ \|\mathbf{r}(i,j)\| \leq 1 \right\}, \quad \forall i = 1, ..., M, j = 1, ..., N. \tag{5.31}$$

The whole model is optimized using the primal-dual framework as shown above and the resulting optimization steps are given by

$$\begin{cases} \mathbf{p}^{n+1} = \mathcal{P}_{P_\mathbf{p}} \left\{ \mathbf{p}^n + \sigma_\mathbf{p} \nabla \bar{\mathbf{u}}^n \right\} \\ \mathbf{r}^{n+1} = \mathcal{P}_{P_\mathbf{r}} \left\{ \frac{\mathbf{r}^n + \sigma_\mathbf{r} \nabla \bar{\mathbf{b}}^n}{1 + \sigma_\mathbf{r} \varepsilon} \right\} \\ \mathbf{q}_D^{n+1} = \mathcal{Q}_{Q_D} \left\{ \mathbf{q}_D^n + \sigma_D \lambda_D \left(I_{\Delta t}^D(\mathbf{x}_1, \mathbf{u}_0) + \nabla I_2^D(W_0) \frac{\partial W_0}{\partial \mathbf{u}_0} (\mathbf{u}^n - \mathbf{u}_0) \right) \right\} \\ \mathbf{q}_I^{n+1} = \mathcal{Q}_{Q_I} \left\{ \mathbf{q}_I^n + \sigma_I \lambda_I \left(I_{\Delta t}^I(\mathbf{x}_1, \mathbf{u}_0) + \nabla I_2^I(W_0) \frac{\partial W_0}{\partial \mathbf{u}_0} (\mathbf{u}^n - \mathbf{u}_0) \right) \right\} \\ \mathbf{u}^{n+1} = \mathbf{u}^n - \tau_\mathbf{u} \left(-\nabla^\mathrm{T} {\mathbf{p}_\mathbf{u}}^{n+1} + \lambda_D \left(\nabla I_2^D(W_0) \frac{\partial W_0}{\partial \mathbf{u}_0} - \mathbf{t} \right) \mathbf{q}_D^{n+1} \right. \\ \qquad\qquad \left. + \lambda_I \left(\nabla I_2^I(W_0) \frac{\partial W_0}{\partial \mathbf{u}_0} \mathbf{q}_I^{n+1} + \gamma \mathbf{b}^n \right) \right) \\ \mathbf{b}^{n+1} = \mathbf{b}^n + \tau_\mathbf{b} \lambda_I \gamma \nabla^T \mathbf{r}^{n+1} \\ \bar{\mathbf{u}}^{n+1} = \mathbf{u}^{n+1} + \theta(\mathbf{u}^{n+1} - \bar{\mathbf{u}}^n) \\ \bar{\mathbf{b}}^{n+1} = \mathbf{b}^{n+1} + \theta(\mathbf{b}^{n+1} - \bar{\mathbf{b}}^n) \end{cases} . \tag{5.32}$$

As in our original TV-ℓ_1 optimization the resolvent operator $\mathcal{P}_{P_\mathbf{r}}$ describes the Euclidean projection onto the unit circle. The newly introduced time-steps for the compensation and flow update $\sigma_\mathbf{r}$, $\tau_f l$ and $\tau_\mathbf{b}$ are again calculated with diagonal preconditioning as shown in Section 3.4.3. Hence the updated time-steps result in

$$\sigma_\mathbf{r} = \frac{\sigma}{2}, \quad \tau_\mathbf{b} = \frac{\tau}{4 + \lambda_I \gamma} \tag{5.33}$$

$$\tau_\mathbf{u} = \frac{\tau}{4 + \lambda_D |\nabla I_2^D(W_0) \frac{\partial W_0}{\partial \mathbf{u}_0} - \mathbf{b}| + \lambda_I |\nabla I_2^I(W_0) \frac{\partial W_0}{\partial \mathbf{u}_0} + \gamma|}. \tag{5.34}$$

The model is optimized to the global illumination factor in the *ToF* case or to the high reflective and shadowed parts of the intensity image.

5.5.1.2 Structure/Texture decomposition

Similar to a separate illumination compensation model which is estimated in the optimization, methods have been proposed where the illumination part is separated to each input image separately in a pre-processing step. Furthermore, these models do not only separate

the illumination part but also all the structural information of intensity images. This is based on the idea that only the textural parts are necessary to successfully estimate a flow model.

In general these decomposition models aim to separate the high-frequent part from the rest of the input image. Most of methods propose to use a higher order term of the input image as input in the constancy assumption. Common higher order terms are to use the gradient, Hessian, Laplacian of the intensity image or the norm of them. In the work of Papenberg et al. [80] an overview of the different variants is shown.

One of the most promising separation from *OF* is to decompose the input image into the structure part and the texture part, as presented in [127]. In this sense, the structure part represents the low frequency component of the image and the texture the high frequency component. The combination of both parts results in the original input image. For intensity images it can be said that most illumination changes are encoded in the structure part while the textural information necessary to estimate the motion are encoded in the texture part. An example is shown in Figure 5.6 for moving boxes in a reflective environment. Based on a successful decomposition it is therefore beneficial to only use the texture part for flow estimation.

Defining the composition problem, the original input image is defined as

$$\mathbf{I} = (1 - \xi)\mathbf{I}_{\text{Tex}} + \xi\mathbf{I}_{\text{Struct}}. \tag{5.35}$$

In this decomposition $\mathbf{I}^I_{\text{Tex}}$ is the texture part and $\mathbf{I}^I_{\text{Struct}}$ is the structure part. The scalar weighting term $\xi \in \mathbb{R} \mid (0, 1)$ defines the tradeoff between both parts.

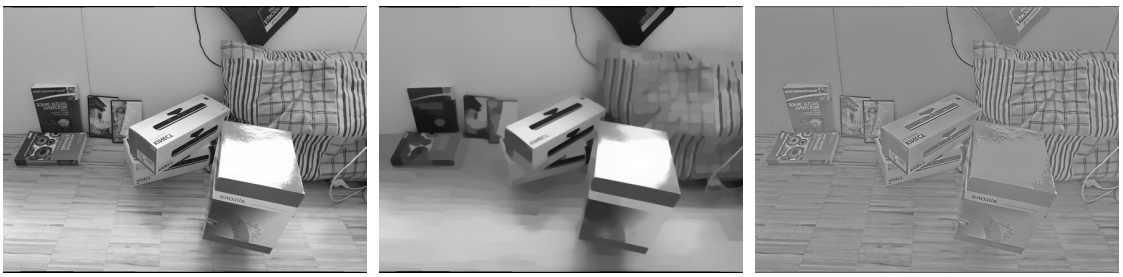
By an interpretation of the texture part as *noise*, the structure part can be defined as denoised version of the input image according to Eqn. (5.35). The structural image $\mathbf{I}_{\text{Struct}}$ can be either simply calculated as a blurred version of the input using a Gaussian filter or, as proposed in [127] as ROF denoised model of the input.

As shown in Figure 5.6, using only the texture part is beneficial to compensate illumination changes such as reflections and shadows but compared to the brightness compensation example in Figure 5.5 it is not able to compensate object occlusions in the consecutive frames.

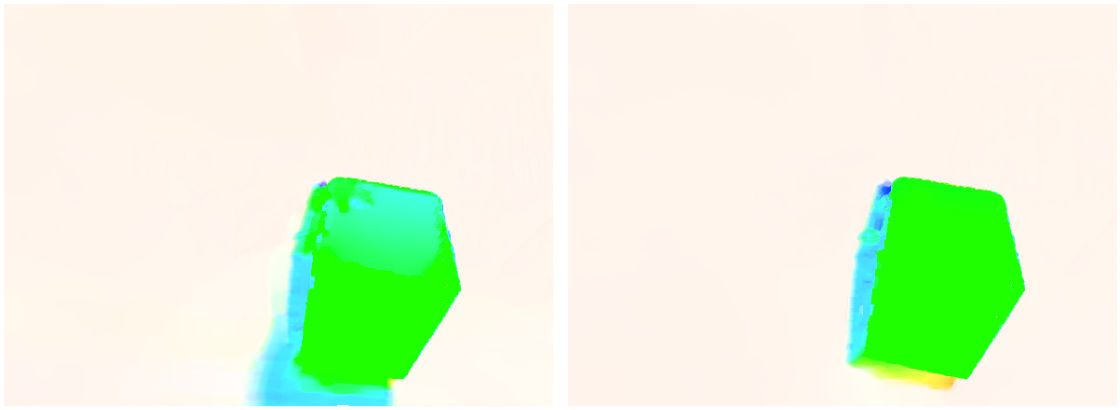
5.5.2 Patchwise Constancy

In this section we propose different approaches to model the constancy assumption on a patch-level. Hence the pixels of consecutive *RGB-D* frames are compared on a local neighborhood around the pixel. Hence, for each pixel a stream of intensity values is compared. Although, the patchwise matching has a higher computational complexity it has the big advantage that the data term gets more robust against intensity variations and noise since compared to a pixelwise term.

Traditional intensity data terms are calculated as pixelwise temporal derivatives by minimizing $I_{\Delta t}(\mathbf{x}, \mathbf{u}) = I_2(W(\mathbf{x}, \mathbf{u})) - I_1(\mathbf{x})$, as shown in Section 5.4. After linearization



(a) Intensity input $\mathbf{I}_1$ **(b)** Structure Part $\mathbf{I}_{\text{Struct}}$ **(c)** Texture Part $\mathbf{I}_{\text{Tex}}$



(d) Flow without decomposition **(e)** Flow from texture

Figure 5.6: Structure Texture decomposition. The input intensity images $\mathbf{I}_1^I$ / $\mathbf{I}_2^I$ are decomposed into their structure and texture component. The structural part is calculated using ROF denoising. The resulting textural part is used to estimate the *SF*. The scene is acquired with a Microsoft K4Wv2 sensor.

this model can be optimized with common convex optimzation frameworks. The downside of this model is that it restricts the data term to be differentiable and therefore to rather simple constraints. In our work we want to model a more robust data term my measuring the constancy in a local neighborhood such as the Sum of Squared Differences (SSD). For *SF*, the constancy for a pixel $\mathbf{x}$ measured around a local window $\mathcal{N}(\mathbf{x})$ is defined by a function

$$G(\mathbf{x}, \mathbf{u}) = \frac{1}{|\mathcal{N}|} \sum_{\mathbf{y} \in \mathcal{N}(\mathbf{x})} \left(I_2(W(\mathbf{y}, \mathbf{u})) - I_1(\mathbf{y}) \right)^2 , \tag{5.36}$$

which measures the normalized *SSD* of intensity values. This function alone is much more robust against noise compared to the pixelwise constancy as defined above. Although this matching function is better suited for real-world conditions a "simple" linearization in the image space is no longer possible. The principle idea to use more complex models in convex gradient descend models to directly linearize and derivate the matching function $G(\mathbf{x}, \mathbf{u})$ itself around the initial flow field $\mathbf{u}_0$. Although the direct linearization requires a

numerical derivation, it has the big advantage that the matching function can be arbitrary chosen.

Following [31] and [130], a second order Taylor expansion of the matching function is sufficient in most cases. This second order Taylor approximation of an arbitrary matching function $G(\mathbf{x}, \mathbf{u})$ is defined by

$$G(\mathbf{x}, \mathbf{u}) \approx \tilde{G}(\mathbf{x}, \mathbf{u}) = G(\mathbf{x}, \mathbf{u}_0) + \nabla G(\mathbf{x}, \mathbf{u}_0)^T (\mathbf{u} - \mathbf{u}_0) \tag{5.37}$$

$$+ \frac{1}{2} (\mathbf{u} - \mathbf{u}_0)^T \nabla^2 G(\mathbf{x}, \mathbf{u}_0)(\mathbf{u} - \mathbf{u}_0). \tag{5.38}$$

In this approximation the second derivative $\nabla^2 G(\mathbf{x}, \mathbf{u}_0)$ (Hessian) of the matching function is defined by

$$\nabla^2 G(\mathbf{x}, \mathbf{u}_0) = \begin{bmatrix} G_{xx}(\mathbf{x}, \mathbf{u}_0) & G_{xy}(\mathbf{x}, \mathbf{u}_0) & G_{xz}(\mathbf{x}, \mathbf{u}_0) \\ G_{yx}(\mathbf{x}, \mathbf{u}_0) & G_{yy}(\mathbf{x}, \mathbf{u}_0) & G_{yz}(\mathbf{x}, \mathbf{u}_0) \\ G_{zx}(\mathbf{x}, \mathbf{u}_0) & G_{zy}(\mathbf{x}, \mathbf{u}_0) & G_{zz}(\mathbf{x}, \mathbf{u}_0) \end{bmatrix}. \tag{5.39}$$

To be convex in $\tilde{G}(\mathbf{x}, \mathbf{u})$ this symmetric Hessian matrix *must*, by definition, be a positive semi-definite matrix, as shown in [11]. This is necessary to ensure convexity in the optimization. The positive semi-definiteness can be checked if all Eigenvalues of the matrix are positive. To transform the Hessian into a positive semi-definite matrix each entry with a negative Eigenvalue is set to zero, as shown in [129].

This transformation is not very efficient to compute because this requires the Eigenvalues of the 3×3 Hessian matrix for every pixel of the dense flow field after every update of $\mathbf{u}_0$. An alternative more efficient approximation is to neglect every mixed derivative in the matrix with $G_{ij} = 0, \ \forall i \neq j$. As shown in [31] this yields the positive semi-definite approximation of the Hessian defined by

$$\nabla^2_+ G(\mathbf{x}, \mathbf{u}_0) = \begin{bmatrix} G_{xx}(\mathbf{x}, \mathbf{u}_0)^+ & 0 & 0 \\ 0 & G_{yy}(\mathbf{x}, \mathbf{u}_0)^+ & 0 \\ 0 & 0 & G_{zz}(\mathbf{x}, \mathbf{u}_0)^+, \end{bmatrix}, \tag{5.40}$$

where only positive second order derivatives are allowed. In [129] it has been shown that this approximation does not harm the estimation accuracy.

With this generally applicable function approximation we are able to realize more complex data terms. In the following we will show and discuss multiple different matching terms which are especially suited for *SF* estimation. These matching functions are chosen to robustify the traditional *SF* constraints from Section 5.3 in terms of noise and illumination changes. In spite of this generalization our framework is still capable to deal with multiple data terms on the same input image, e.g. combining Absolute Differences (AD) with *NCC* and to use the quadfit term for the intensity as well as the depth frame. Therefore we will refer to $\mathbf{I}_1, \mathbf{I}_2$ as the handled input image pair irrespective if the function

G is applied on the depth or on the intensity frames.

5.5.2.1 Sum of Differences

The *sum of differences* measures the the deviation from the warped target image $\mathbf{I}_2$ to the source image $\mathbf{I}_1$ penalized by a norm and summed up in a local neighborhood $\mathcal{N}$.

As shortly shown above, the *SSD* at an image position $\mathbf{x}$ for a given flow field $\mathbf{u}$ is given by

$$G(\mathbf{x},\mathbf{u})_{\text{SSD}} = \frac{1}{|\mathcal{N}(\mathbf{x})|} \sum_{\mathbf{y}\in\mathcal{N}(\mathbf{x})} \|I_2(W(\mathbf{y},\mathbf{u})) - I_1(\mathbf{y})\|_2^2. \tag{5.41}$$

Compared to a pixel-wise error measure the *SSD* is more robust against noise but similar to the ℓ_2 penalization of the pixel-wise data term it is not robust against gross outliers in the input, such as Salt & Pepper noise.

Similar, the Sum of Absolute Differences (SAD) matching function is given by the sum of ℓ_1 penalizations between the first images and the warped second image on a local neighborhood $\mathcal{N}$ and is given by

$$G(\mathbf{x},\mathbf{u})_{\text{SAD}} = \frac{1}{|\mathcal{N}(\mathbf{x})|} \sum_{\mathbf{y}\in\mathcal{N}(\mathbf{x})} \|I_2(W(\mathbf{y},\mathbf{u})) - I_1(\mathbf{y})\|_1. \tag{5.42}$$

This data term has the same advantages as the ℓ_1 norm on the pixel-wise constraints combined with the robustness against noise because of the patch level. With a neighborhood of size $|\mathcal{N}| = 1$ this data term results in the traditional data term as shown above.

Similarly also any other sum of differences can be realized in our framework. But the major drawback of difference models is that their robustness to illumination changes, shadows or occlusions is rather limited. To be also capable of these local changes more complex methods have been proposed, which are described in the following.

5.5.2.2 Normalized Cross Correlation

Matching terms which are more robust to intensity variations in a local neighborhood between the images, such as changing illumination, are often designed to calculate only signs of the difference or the normalized differences. The *NCC* is one of those methods. As shown in [12, 59] it calculates the correlation of two patches which are normalized by mean and standard deviation.

The image patches are calculated by normalization to zero mean and standard deviation of one. The normalized image regions in a local neighborhood $\mathbf{y} \in \mathcal{N}$ around the pixel $\mathbf{x}$ are defined as

$$\mathbf{I}_1^n(\mathbf{y}) = \frac{\mathbf{I}_1(\mathbf{y}) - \bar{\mathbf{I}}_1(\mathbf{x})}{\sigma_{\mathbf{I}_1}} \quad \text{and} \quad \mathbf{I}_2^n(\mathbf{y}) = \frac{\mathbf{I}_2(W(\mathbf{y},\mathbf{u})) - \bar{\mathbf{I}}_2(W(\mathbf{x},\mathbf{u}))}{\sigma_{\mathbf{I}_2}}, \tag{5.43}$$

where $\bar{\mathbf{I}}(\mathbf{x})$ is defined as the mean value and $\sigma_{\mathbf{I}}$ is the standard deviation of all values in $\mathcal{N}(\mathbf{x})$. Mean and standard deviation are calculated by

$$\bar{\mathbf{I}}(\mathbf{x}) = \frac{1}{|\mathcal{N}(\mathbf{x})|} \sum_{\mathbf{y} \in \mathcal{N}(\mathbf{x})} \mathbf{I}(\mathbf{y}) \quad \text{and} \quad \sigma_{\mathbf{I}} = \sqrt{\frac{1}{|\mathcal{N}(\mathbf{x})|} \sum_{\mathbf{y} \in \mathcal{N}(\mathbf{x})} \big(\mathbf{I}(\mathbf{y}) - \bar{\mathbf{I}}(\mathbf{x})\big)^2}, \tag{5.44}$$

respectively. Hence, the cross correlation of the normalized patches is defined by the sum of correlation values between the normalized patches given by

$$\delta_{\text{NCC}}(\mathbf{x}) = \frac{1}{|\mathcal{N}(\mathbf{x})|} \sum_{\mathbf{y} \in \mathcal{N}(\mathbf{x})} \mathbf{I}_1^n(\mathbf{y}) \mathbf{I}_2^n(\mathbf{y}), \tag{5.45}$$

which can be interpreted as the cosine angle between the unit feature vectors as $\langle \mathbf{I}_1^n, \mathbf{I}_2^n \rangle$. This normalization by mean and standard deviation, as often used in template matching problem, has the advantage that the correlation calculates a relative difference independent of the overall multiplicative or additive intensity differences between the patches.

The *NCC* calculates a correlation value for each pixel in the image according to the normalization. This value ranges from -1 if the two patches do not match at all and $+1$ if they are matching perfectly. To implement this value into our primal minimization the constraint function is defined at

$$G(\mathbf{x}, \mathbf{u})_{\text{NCC}} = 1 - \max\left(0, \delta_{\text{NCC}}(\mathbf{x})\right) \tag{5.46}$$

where the matching value now is ranged by $G_{\text{NCC}} \in \mathbb{R}^{M \times N} \mapsto (0, 1)$.

The big advantage of the *NCC* is that it is still robust to noise and homogeneous regions similar to the *SAD* but above that it is robust to illumination changes. On the downside, the *NCC* is much more computationally expensive. In the following we show other matching approaches we used for *SF* estimation which have the same performance to illumination changes with a much lower computational cost.

5.5.2.3 Census Transform

The Census Transform (CT), first proposed by Zabih and Woodfill [135], is a technique to encode an intensity patch as a binary feature vector. Two of those binary feature vectors are efficiently matched with the other binary vectors by the Hamming distance [41]. This *CT* has been successfully used in other computer vision approaches such as Face Detection [33] and Stereo Estimation [92].

In the original *CT* the binary vector is defined as an ordered value comparison of every pixel in the patch to the center pixel. Hence, for an image $\mathbf{I}$ the Census signature at an

image position $\mathbf{x}$ is given by

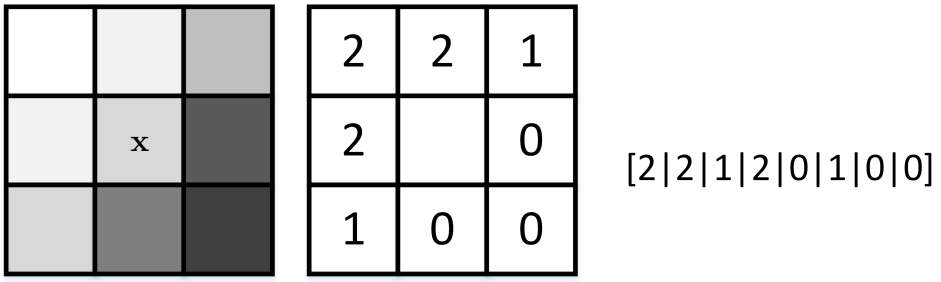
$$\mathbf{C}(\mathbf{I},\mathbf{x}) = \bigotimes_{\mathbf{y}\in\mathcal{N}'(\mathbf{x})} \{\xi(\mathbf{x},\mathbf{y})\}, \quad \text{where} \quad \xi(\mathbf{x},\mathbf{y}) = \begin{cases} 0, & \text{if} \quad \mathbf{I}(\mathbf{x}) < \mathbf{I}(\mathbf{y}) \\ 1, & \text{if} \quad \mathbf{I}(\mathbf{x}) \geq \mathbf{I}(\mathbf{y}) \end{cases} \tag{5.47}$$

where $\mathcal{N}'(\mathbf{x}) = \mathcal{N}(\mathbf{x}) \cup \mathbf{x}$ excluding the center pixel. The operator $\bigotimes$ denotes the concatenation operator. For an input image $\mathbf{I}$ this results in a Census image $C \in \mathbb{R}^{M\times N} \mapsto \mathbb{R}^{|\mathcal{N}'|}$ where each pixel encodes the local binary census signature of length $|\mathcal{N}'|$. The similarity between two patches is given by the Hamming distance. In our method the Census data term for one pixel is given as the normalized sum of differences between the census signatures of the warped image $\mathbf{I}_2(W(\mathbf{u},\mathbf{x}))$ and $\mathbf{I}_1(\mathbf{x})$ given by

$$G(\mathbf{x},\mathbf{u})_{\mathrm{CT}} = \frac{1}{|\mathcal{N}'(\mathbf{x})|} \sum_{i=1}^{|\mathcal{N}'|} [\mathbf{C}_i(I_2, W(\mathbf{x},\mathbf{u})) \neq \mathbf{C}_i(\mathbf{I}_1,\mathbf{x})]] \tag{5.48}$$

which measures the normalized sum of differences between $\mathbf{C}_i(I_2, W(\mathbf{x},\mathbf{u}))$ and $\mathbf{C}_i(\mathbf{I}_1,\mathbf{x})$.

Compared to the *NCC*, the Census signature and Hamming distance between Census signatures can be implemented very efficiently in terms of storage and computational effort. E.g. for a patch-size of 3×3 the census signature needs only 8Bit of storage and the hamming distance between patches can be easily calculated by the XOR operation between the signatures. The census has still the nice property that it is invariant to most changes in illumination.



(a) $\mathcal{N}'(\mathbf{x})$ **(b)** $\xi(\mathbf{I},\mathcal{N}(\mathbf{x}))_{\mathrm{TCT}}$ **(c)** $C(\mathbf{I};\mathbf{x})_{\mathrm{TCT}}$

Figure 5.7: Ternary Census Transform *TCT*. The *TCT* string (c) of pixel $\mathbf{x}$ is calculated by the concatenation of the ternary sign (b) from the local neighborhood $\mathcal{N}'$ (a) from center pixel $\mathbf{x}$.

Although the calculation of the binary Census signature is very simple it has some drawbacks dealing with noise around the center pixel. This sensitivity to noise can be avoided by a simple extension of the signature calculation. Instead of calculating a binary signature Stein [111] proposed a method to extend the Census signature to three distinctive cases. Hence, it is called the the Ternary Census Transform (TCT). Additionally to the two cases for lower and higher than the center pixel a third case is introduced for values

"similar" to the center pixel intensity. The pixel-wise *TCT* signature is given by

$$\xi(\mathbf{x},\mathbf{y})_{\mathrm{TCT}} = \begin{cases} 0, & \text{if } \mathbf{I}(\mathbf{y}) - \mathbf{I}(\mathbf{x}) < -\varepsilon \\ 1, & \text{if } |\mathbf{I}(\mathbf{y}) - \mathbf{I}(\mathbf{x})| \leq \varepsilon \\ 2, & \text{if } \mathbf{I}(\mathbf{y}) - \mathbf{I}(\mathbf{x}) > \varepsilon \end{cases}, \quad \text{where} \tag{5.49}$$

$$C(\mathbf{I},\mathbf{x})_{\mathrm{TCT}} = \bigotimes_{\mathbf{y}\in\mathcal{N}'(\mathbf{x})} \{\xi(\mathbf{x},\mathbf{y})_{\mathrm{TCT}}\} \tag{5.50}$$

where the scalar value $\varepsilon \in \mathbb{R}_{\geq 0}$ denotes the sensitivity to intensity differences. Note that for $\varepsilon = 0$ the *TCT* is equal to the standard *CT*. In Figure 5.7 the *TCT* transformation from patch intensity values to the Census signature is shown for one exemplar patch. The similarity between two patches is again given by the Hamming distance from Eqn. (5.48) and is denoted as $G(\mathbf{x},\mathbf{u})_{\mathrm{TCT}}$.

This *TCT* matching can be efficiently computed and combines a robustness to illumination changes such as global illumination, shadows and occlusions and being robust to acquisition noise.

5.5.2.4 Closest Points

The depth constancy term defined in Section **??** is calculated by minimizing the pixelwise temporal derivatives in the depth image space. This is given by

$$\mathbf{I}^D_{\Delta t}(\mathbf{x}_1,\mathbf{u}) = \mathbf{I}^D_2(W(\mathbf{x}_1,\mathbf{u})) - \mathbf{I}^D_1(\mathbf{x}_1) - u_Z, \tag{5.51}$$

where the warping function defines the back-projection into 3D space and the projection into the image space according to the flow. Since the 3D data is directly available from the depth measurements we propose a data term where the matching is calculated directly in 3D space. This has the advantages that we do not need a projection into the image space. In this context, we propose a flow error metric based on the *ICP* algorithm [139], where the flow error is calculated as point differences directly in 3D space to match the local surface structure. Since we have two consecutive depth acquisitions $\mathbf{I}^D_1, \mathbf{I}^D_2$ with known camera intrinsics we calculate the 3D scene points $\mathbf{X}_1, \mathbf{X}_2$ by back-projection. To follow the principle of patchwise comparison we define the depth difference of a point $\mathbf{X}$ in a local neighborhood around the center point. Hence, we get more robust in the case of noise and low structural parts in the depth maps, similar to the patchwise terms in the image space. Including the flow vectors $\mathbf{u}$, the residual error between these two projected point clouds is defined by

$$G(\mathbf{x},\mathbf{u})_{\mathrm{CP}} = \frac{1}{|\mathcal{N}|} \sum_{\mathbf{y}\in\mathcal{N}(\mathbf{x})} \|\mathbf{X}_2(\mathbf{y}) - \mathbf{u}(\mathbf{y}) - \mathbf{X}_1(\mathbf{y}^*)\|_2, \in \Omega, \tag{5.52}$$

where the point $\mathbf{X}_1(\mathbf{y}^*)$ is denoted as the closest correspondence to $\mathbf{X}_2(\mathbf{y})$. In the context of *SF* the closest point to the transformed $\mathbf{X}_2(\mathbf{y})$, *i.e.*

$$\mathbf{y}^* = \underset{\mathbf{y}\in\Omega}{\operatorname{argmin}} \|\mathbf{X}_2(\mathbf{x}) - \mathbf{u}(\mathbf{x}) - \mathbf{X}_1(\mathbf{y})\|_2, \tag{5.53}$$

as shown in Figure 5.8. This correspondence can be calculated by a quadratic search or, more efficiently, by the k-Nearest Neighbor calculation. When matching 3D patches in-

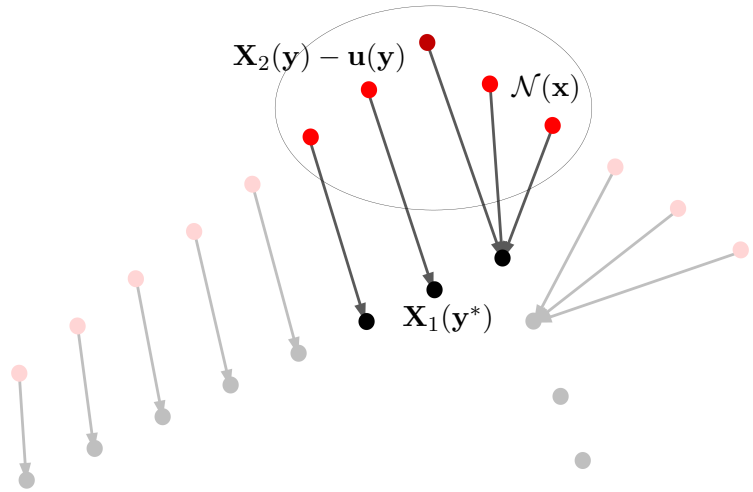


Figure 5.8: Patchwise closest point search. The depth fidelity at point $\mathbf{x}$ is measured as mean Euclidean distance of the all warped points $\mathbf{X}_2(\mathbf{y}) - \mathbf{u}(\mathbf{y})$ in the neighborhood $\mathcal{N}$ around $\mathbf{x}$ to the closest points in the first acquisition $\mathbf{X}_1(\mathbf{y}^*)$.

stead of depth pixel values the method becomes more robust against homogeneous depth regions and acquisition noise. Further, the direct matching in 3D does not lead to information loss due to back-projection and interpolation into the image space.

5.5.2.5 Multiscale Patchwise Terms

The above patchwise terms are calculated for a fixed neighborhood size $\mathcal{N}$ around each pixel during the optimization. In our experiments we figured out that the optimal size of this neighborhood often changes across datasets or even across regions in one image sequence. Hence, we propose to use a calculation of multiple neighborhoods for each pixel. Since all patchwise terms are normalized by the region size the best fitting neighborhood size is chosen by the minimal matching cost.

$$\mathcal{N}(\mathbf{x})^* = \arg\min_{\mathcal{N}} \left\{ G(\mathbf{x},\mathbf{u})^{\mathcal{N}^1}, G(\mathbf{x},\mathbf{u})^{\mathcal{N}^2}, \ldots, G(\mathbf{x},\mathbf{u})^{\mathcal{N}^M} \right\} \tag{5.54}$$

Hence the data fidelity cost of the multiscale term is defined as $G(\mathbf{x},\mathbf{u})^{\mathcal{N}^*}$.

5.5.2.6 Optimization of Patchwise Data Terms

When choosing one of the above patchwise terms for the depth image input and one term for the intensity input the new objective function results in

$$\min_{\mathbf{u}} \left\{ \lambda_I \int_\Omega w|\tilde{G}_I(\mathbf{x},\mathbf{u}_0)|\,\mathrm{d}x + \lambda_D \int_\Omega w|\tilde{G}_D(\mathbf{x},\mathbf{u}_0)|\,\mathrm{d}x + \int_\Omega |\nabla \mathbf{u}|\,\mathrm{d}x \right\}, \tag{5.55}$$

where $\tilde{G}_D$ is a patchwise data term from the depth data and $\tilde{G}_I$ is the patchwise data term based on the intensity data. The energy is optimized following the primal-dual scheme, where the primal objective from Eqn. (5.55) is transformed into a discretized convex-concave saddle-point problem defined as

$$\min_{\mathbf{u}} \max_{\mathbf{p},\mathbf{q}_D,\mathbf{q}_I} \langle \mathbf{p}, \nabla \mathbf{u}\rangle_{P_\mathbf{p}} + \lambda_I \langle \mathbf{q}_D, \tilde{G}_D(\mathbf{x},\mathbf{u}_0)\rangle_{Q_D} + \lambda_I \langle \mathbf{q}_I, \tilde{G}_I(\mathbf{x},\mathbf{u}_0)\rangle_{Q_I}. \tag{5.56}$$

With the formulation of the discrete saddle-point problem (5.56) we can apply the primal-dual algorithm. As shown in Section 3.4.3, the algorithm iteratively optimizes the primal and the dual variables for the individual pixels in three steps. First, the dual variables $\mathbf{p}$, $\mathbf{q}_D$ and $\mathbf{q}_I$ are updated using gradient ascend. Second, the primal optimizer $\mathbf{u}$ is updated using gradient descend. Third, $\mathbf{u}$ is refined in an over-relaxation step. For any iteration $n \geq 0$ and every level the primal dual steps are calculated according to

$$\begin{cases} \mathbf{p}_\mathbf{u}^{n+1} & = \mathcal{P}_{P_\mathbf{u}} \left\{ \mathbf{p}_\mathbf{u}^n + \sigma_{P_\mathbf{u}} \left(\nabla \bar{\mathbf{u}}^n \right) \right\} \\ q_I^{n+1} & = \mathcal{Q}_{Q_I} \left\{ q_I^n + \sigma_I \lambda_I w \left(\tilde{G}_I(\mathbf{x},\mathbf{u}_0) \right) \right\} \\ q_D^{n+1} & = \mathcal{Q}_{Q_D} \left\{ q_D^n + \sigma_D \lambda_D w \left(\tilde{G}_D(\mathbf{x},\mathbf{u}_0) \right) \right\} \\ \mathbf{u}^{n+1} & = \left(I + \tau_\mathbf{u} \left(\lambda_I w \nabla_I^2 + \lambda_D w \nabla_D^2 \right) \right)^{-1} \\ & \quad \Big\{ \mathbf{u}^n + \tau_\mathbf{u} \Big(\alpha_1 \nabla^\mathrm{T} {\mathbf{p}_\mathbf{u}}^{n+1} \\ & \quad + \lambda_I w \left(\nabla G_I(\mathbf{x},\mathbf{u}_0) - (\nabla_I^2)^T (\mathbf{u} - \mathbf{u}_0) \right) \\ & \quad + \lambda_D w \left(\nabla G_D(\mathbf{x},\mathbf{u}_0) - (\nabla_D^2)^T (\mathbf{u} - \mathbf{u}_0) \right) \Big) \Big\} \\ \bar{\mathbf{u}}^{n+1} & = 2\mathbf{u}^{n+1} - \bar{\mathbf{u}}^n, \end{cases} \tag{5.57}$$

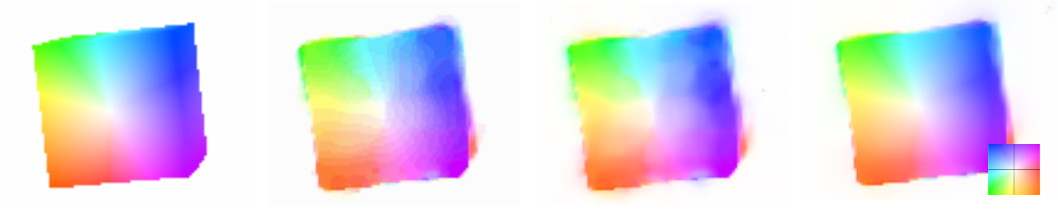
where $\nabla^2 G_I(\mathbf{x},\mathbf{u}_0)$ and $\nabla^2 G_D(\mathbf{x},\mathbf{u}_0)$ is denoted as ∇_I^2 and ∇_D^2 for brevity. The operator $\mathcal{P}_{P_\mathbf{u}}$ is defined as the point-wise Euclidean projections onto the convex set $P_\mathbf{u}$.

5.6 Advanced Regularization

In the previous section we presented multiple advances of the data term $\mathcal{D}$ to improve the robustness of the matching cost in the *SF* calculation. On the other side, there is also some room for improvement in the regularization term $\mathcal{R}$, which ensures a coherency between neighboring matches aims to generate a dense and hopefully noise-free motion estimation. In Section 5.4.1 we have already shown the basic *TV* regularization which penalizes the

flow gradient by a ℓ_1 norm. Compared to a ℓ_2 penalization the *TV* regularization already has the advantage that is does not smooth across sharp edges, but the main problem is that the *TV* favors piecewise constant solutions. This property is very useful for intensity image denoising but similar to the discussed depth superresolution approaches in the last chapter, this can be problematic for *SF* estimation especially in regions where a smooth motion transition exists, as shown in Figure 5.9.

In the following we show alternatives for convex regularization terms that are better suited for the multi-dimensional *SF* estimation. This regularization terms are similar to the regularization terms in Chapter 4. Second, we propose advanced weighing terms which use the information from the input sequences. In these terms the input images are used to guide the regularization which leads to an accuracy improvement in regions around object borders.



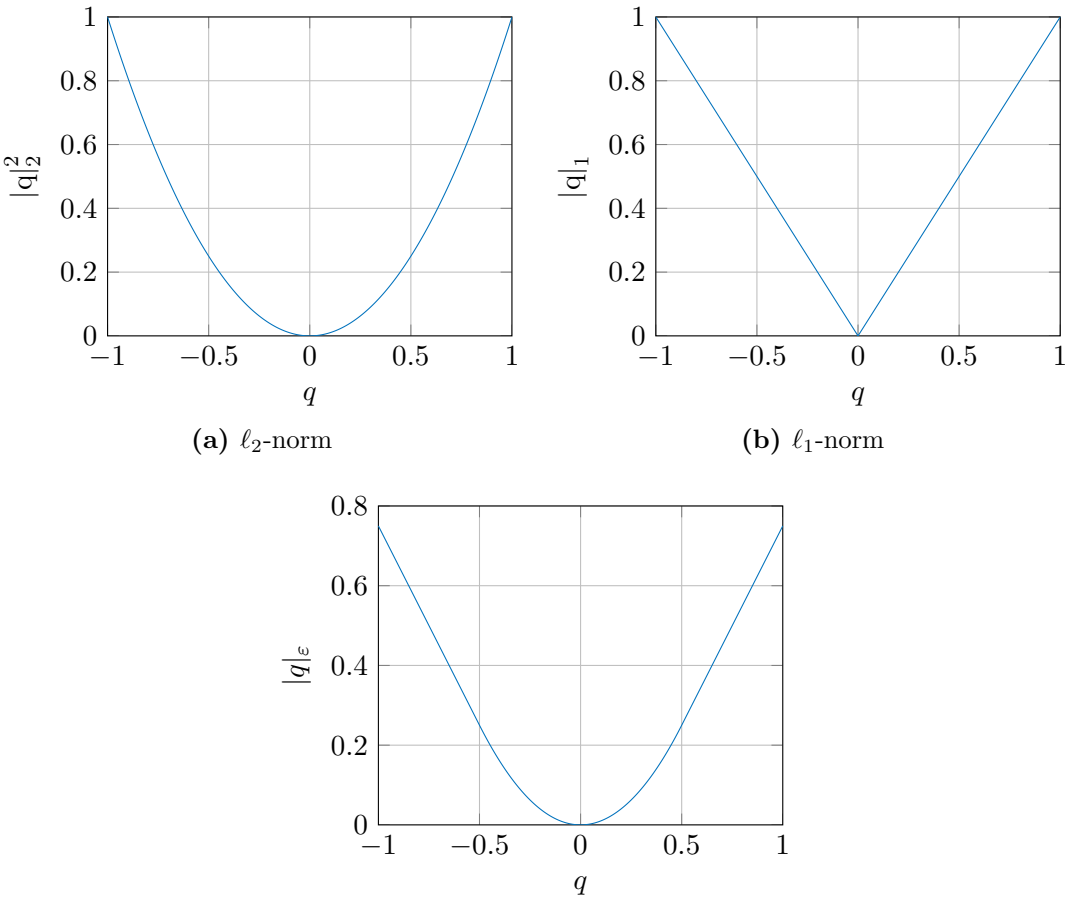
(a) Groundtruth **(b)** *TV* **(c)** *Huber* **(d)** *TGV*

Figure 5.9: Comparison of the regularization term for a simple box rotation. The *TV* regularization (b) optimizes for a piecewise constant solution. The *Huber* regularization (c) is more smooth but suffers from oversmoothing the boundaries. The Total Generalized Variation (TGV) regularization (d) optimizes for piecewise affine solution which fits best to the groundtruth flow. The flow is color coded in x and y direction. The color key is shown on the bottom right.

5.6.1 Huber Regularization

The *TV* regularization is used for many optimization problems such as denoising [99] or *OF* estimations [136]. In the case of *SF* estimation it is well suited to generate sharp motion boundaries but on the other hand it leads to erroneous results at smooth flow transitions such as object rotations since the *TV* favors piecewise constant solutions. This leads to a staircasing effect in the solution. As we have shortly discussed for the illumination compensation in Section 5.5.1, the Huber regularization is an alternative to the *TV*. Similar to the *TV* the Huber regularizer is a penalization on the motion gradient but in contrast to the *TV* it has a quadratic penalization on smaller gradients and an ℓ_1 penalization for larger values. To shortly recapitulate the definition shown in Section 3.1 the Huber-norm is defined as

$$|q|_\varepsilon = \begin{cases} \frac{|q|^2}{2\varepsilon} & \text{if} \quad |q| \leq \varepsilon \\ |q| - \frac{\varepsilon}{2} & \text{if} \quad |q| > \varepsilon \end{cases}. \tag{5.58}$$



(a) ℓ_2-norm

(b) ℓ_1-norm

(c) Huber-norm $\varepsilon = 0.5$

Figure 5.10: Different Norms for gradient penalization. While the penalization with the ℓ_2-norm leads to a smoothing of flow edges, the ℓ_1 penalization favors a piecewise constant solutions in the flow field. The Huber-norm is a tradeoff between ℓ_2-norm and ℓ_1-norm. Gradient values below ε are penalized with the ℓ_2-norm and values above ε are penalized with the ℓ_1-norm.

An illustration of the different norms is shown in Figure 5.10 for the 1D case. In the case of *SF* estimation the combination of ℓ_1 and ℓ_2 norms in the Huber penalization allows smooth flow transitions while the motion boundaries are only slightly smoothed. An example of the Huber norm is shown in Figure 5.9c.

5.6.2 Total Generalized Variation

A recently introduced generalization of the *TV* was proposed by Bredies et al. [13], namely the Total Generalized Variation (TGV). While the *TV* penalizes the first derivative of the optimizer, the *TGV* penalizes the derivative of arbitrary order. E.g. the *TGV* of second order penalizes the second derivative and hence, favors piecewise affine solutions instead

of piecewise constant solution as in th *TV* case. It has been shown that this property is very beneficial for depth denoising [28] and motion estimation [29].

Generally, a model that is regularized by a *TGV* term of order k favors a solution that is piecewise composed of polynomials of order $k-1$, as defined in Section 4.3.

For *SF* estimation it turns out that the second order *TGV* is a well defined trade-off between accuracy and computational complexity. This regularization penalizes the second derivatives of the flow field. Hence it approximates the *SF* by piecewise affine transformations. The primal definition of the second order *TGV* is formulated as

$$\mathrm{TGV}_{\alpha}^{2} = \min_{\mathbf{u},\mathbf{v}} \left\{ \alpha_1 \int_{\Omega} |\nabla \mathbf{u} - \mathbf{v}| \, \mathrm{d}x + \alpha_0 \int_{\Omega} |\nabla \mathbf{v}| \, \mathrm{d}x \right\}, \tag{5.59}$$

where additional to the first order smoothness of the flow field $\mathbf{u}$, the auxiliary variable $\mathbf{v}$ is introduced to enforce second order smoothness. The scalars α_0, $\alpha_1 \in \mathbb{R}_{\geq 0}$ are used to weight each order. Because the *TGV* regularizer is convex in $\mathbf{u}$ and $\mathbf{v}$ it allows to compute a global optimal solution.

Compared to the standard *TV* the *TGV* of second order avoids the piecewise constancy in the solution and in contrast to the Huber-norm it directly optimizes for piecewise affine flow transitions instead of an approximation. Hence, we directly can optimize for more complex movements while compared to the Huber-norm we can still preserve sharp motion objects.

In Figure 5.9 a comparison of the different regularizers is shown. As one can see in the experiments, the *TV* generates piecewise constant solutions which result in a staircasing in the output map. Although the Huber-norm is more accurate at smooth flow transitions than a *TV*-norm, it has problems to preserve sharp edges due to the ℓ_2 smoothing for at smaller gradients. The *TGV* regularization creates the best results among the proposed regularization terms where both sharp edges as well as smooth transitions are possible.

5.6.3 Regularizer Guidance

Building on the idea that moving objects and the flow field share the same edges we can use the input images to guide the optimization along these edges. Hence we can model a guidance image out of the depth image to weight the regularization term. Hence, the regularization force will be increased in homogeneous areas while it is decreased at object boundaries. This will help to estimate sharper edges while homogeneous areas are still smoothed. There are many ways to use the input data to aid the regularization term and in this section we show a few of them which are used throughout our methods.

The basic idea of the guidance term is to weight the regularization according to the input data where we build on the assumption that motion boundaries are more likely to appear at object boundaries such as high intensity and depth variations. To increase the estimation accuracy in those areas a simple approach is to use the image gradient information to weight the regularization (Huber, *TV* or *TGV*). Hence, the influence of the

regularization is increased at more homogeneous regions (more smoothing) and decreased at high gradient magnitudes (lesser smoothing). This simple scalar weighting is has already been used in other computer vision tasks such as denoising [58], superresolution [25] and image segmentation [15].

Scalar Weighting In our work the pointwise weighting function $g(\mathbf{x}) \in \mathbb{R}^{M \times N} \mapsto \mathbb{R}$ is defined by the image gradient as

$$g(\mathbf{x}) = \exp\left(\alpha_T \|\nabla \mathbf{I}^D(\mathbf{x})\|_2^{\beta_T}\right), \tag{5.60}$$

where the scalars $\alpha_T, \beta_T \in R$ are terms to weight the magnitude and sharpness of the function. The gradients of the guidance image are calculated by the Sobel operator to reduce the influence of acquisition noise on the weighting. In the case of *TV* regularization the complete term is given by

$$\mathcal{R}(\mathbf{u}) = \int_\Omega g(\mathbf{x}) \|\nabla \mathbf{u}(\mathbf{x})\|_1 \mathrm{d}\mathbf{x}. \tag{5.61}$$

This simple weighting term already delivers much better optimization results but it has problems because the magnitude weighting not only down-weights the regularization across motion boundaries but also along the boundaries. This can lead to high outliers in border regions.

As shown in previous works, by including the guidance term in the optimization model the overall numerical optimization scheme only changes with respect to the gradient ∇ and the divergence operator ∇^T in the dual and the primal step. When using the pointwise weighting the gradient and the divergence is simply multiplied by the gradient magnitude.

Gradient Weighting A very similar guidance approach is to account for the gradient difference in x and y direction and weight the motion gradient in each direction separately. Hence, the weighting term from Eqn. (5.60) is changed to the weighting vector

$$\mathbf{g}(\mathbf{x}) = \exp\left(\alpha_T |\nabla \mathbf{I}^D(\mathbf{x})|_2^{\beta_T}\right) \in \mathbb{R}^2. \tag{5.62}$$

The scalar parameters α_T and β_T are again weighting the influence on magnitude and sharpness. In the simple *TV* case, the gradient weighted regularization is given by

$$\mathcal{R}(\mathbf{u}) = \int_\Omega \|\mathrm{diag}\left(\mathbf{g}(\mathbf{x})\right) \nabla \mathbf{u}(\mathbf{x})\|_1 \mathrm{d}\mathbf{x}. \tag{5.63}$$

Hence, in the optimization for every image point $\mathbf{x}$ the weighted gradient and weighted divergence of the motion are given by $(\mathrm{diag}(\mathbf{g})\nabla)\,\mathbf{u}$ and $\left(\nabla^T \mathrm{diag}(\mathbf{g})\right)\mathbf{p}$, respectively. This gradient weighted term now accounts for each direction separately in hence yields better results than a simple scalar weighing.

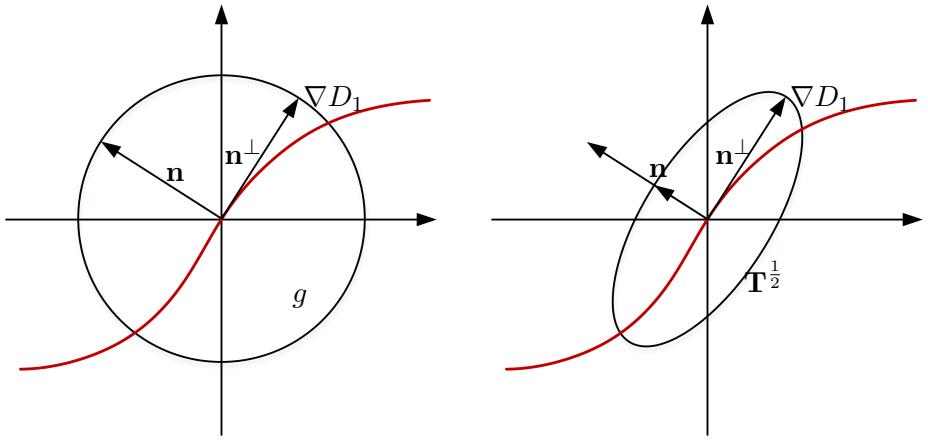
Anisotropic Weighting The anisotropic weighing term goes one step further by defining a rotation matrix that not only weights each gradient direction but also rotates the gradient vector by its direction. The idea is to integrate a full anisotropic diffusion matrix into the 2D gradient regularization based on the Nagel-Enkelmann operator [71]. Following our gradient definition from above, the anisotropic diffusion tensor $\mathbf{T}^{\frac{1}{2}}$ is defined as a symmetric 2×2 matrix given by

$$\mathbf{T}^{\frac{1}{2}}(\mathbf{x}) = \exp\left(\alpha_T |\nabla \mathbf{I}^D(\mathbf{x})|_2^{\beta_T}\right) \mathbf{n}\mathbf{n}^T + \mathbf{n}^{\perp}\mathbf{n}^{\perp T} = \begin{pmatrix} a & c \\ c & b \end{pmatrix}, \tag{5.64}$$

where $\mathbf{n}$ is the normalized direction of the guidance image gradient

$$\mathbf{n} = \frac{\nabla \mathbf{I}^D}{\|\nabla \mathbf{I}^D\|_2} \tag{5.65}$$

and $\mathbf{n}^{\perp}$ is the normal vector to the gradient. As in the previous terms the gradient is calculated by the Sobel operator. Note that at completely homogeneous guidance images the tensor becomes the identity matrix. In Figure 5.11 the difference between a simple isotropic weighting and the anisotropic diffusion tensor is shown. The anisotropic diffusion tensor not only weights the motion gradient but also orients the gradient direction during the optimization process. This regularization term has shown great success in stereo reconstruction [92] and depth image upsampling [28].



(a) isotropic weighting **(b)** anisotropic weighting

Figure 5.11: Anisotropic diffusion tensor. While an isotropic weighting of factor g penalizes each gradient direction equally, the anisotropic weighting $\mathbf{T}^{\frac{1}{2}}$ allows flow discontinuities along depth gradients while preventing discontinuities along depth edges [94].

Including the anisotropic tensor $\mathbf{T}^{\frac{1}{2}}$ from Eqn. (5.64) the regularization in our *TV*

example becomes

$$\mathcal{R}(\mathbf{u}) = \int_{\Omega} \|\mathbf{T}^{\frac{1}{2}} \nabla \mathbf{u}(\mathbf{x})\|_1 \mathrm{d}\mathbf{x}. \tag{5.66}$$

In the optimization the motion gradient and divergence operator is changed to

$$\mathbf{T}^{\frac{1}{2}} \nabla \mathbf{u} = \begin{pmatrix} a\nabla_x u_X + c\nabla_y u_X \\ c\nabla_x u_X + b\nabla_y u_X \\ a\nabla_x u_Y + c\nabla_y u_Y \\ c\nabla_x u_Y + b\nabla_y u_Y \\ a\nabla_x u_Z + c\nabla_y u_Z \\ c\nabla_x u_Z + b\nabla_y u_Z \end{pmatrix} \quad \text{, and} \tag{5.67}$$

$$\nabla^T \mathbf{T}^{\frac{1}{2}} \mathbf{p} = \begin{pmatrix} a\nabla_x p_X^1 + c\nabla_y p_X^1 + c\nabla_x p_X^2 + b\nabla_y p_X^2 \\ a\nabla_x p_Y^1 + c\nabla_y p_Y^1 + c\nabla_x p_Y^2 + b\nabla_y p_Y^2 \\ a\nabla_x p_Z^1 + c\nabla_y p_Z^1 + c\nabla_x p_Z^2 + b\nabla_y p_Z^2 \end{pmatrix} \tag{5.68}$$

respectively. Including the anisotropic tensor into the second order *TGV* the combined regularization term is given by

$$\mathrm{TGV}_{\alpha}^2 = \min_{\mathbf{u},\mathbf{v}} \left\{ \alpha_1 \int_{\Omega} |\mathbf{T}^{\frac{1}{2}} (\nabla \mathbf{u} - \mathbf{v})| \, \mathrm{d}x + \alpha_0 \int_{\Omega} |\nabla \mathbf{v}| \, \mathrm{d}x \right\}, \tag{5.69}$$

where the anisotropic tensor only influences the first order term but since all orders are connected by the auxiliary variables it has also influence on the higher orders.

With the combination of convex *TGV* regularization and anisotropic weighting we achieve smooth transitions between flows, typically occuring at object rotations and non-rigid movements, while sharp flow boundaries between moving objects can still be preserved.

5.7 Evaluation

In this section we provide and extensive qualitative and quantitative evaluation of the proposed methods for *SF* estimation.

First, we start with a very simple synthetic scene, where we evaluate on a variety of movements to point out the individual advantages and drawbacks of the proposed models with respect to specific scene motions. This synthetic evaluation is shown in Section 5.7.3.

Second, we show the quality and robustness of the model under different types of acquisition noise in the input data. This noise evaluation is shown in Section 5.7.4.

Third, we will focus on the quantitative evaluation of a commonly used benchmark based on the Middlebury dataset. In this evaluation we show how our method performs compared to *SoA* methods in terms of speed and accuracy. This evaluation is given in Section 5.7.5.

We also demonstrate the applicability to image data acquired in our laboratory. We show a qualitative evaluation of the *SF* models on off-the-shelf *RGB-D* cameras in 5.7.6. In this evaluation we use a PMD Nano and PMD Pico *ToF* camera [83], and a Microsoft Kinect for Windows v2 camera (K4Wv2). Finally we conclude the evaluation section with evaluations of the *SF* estimation for the applications of camera pose estimation and temporal image fusion for depth image superresolution in Section 5.8.

The proposed models are implemented in a C++/GPU framework. The speed of our models is measured as the average execution time over 100 runs computed on a Nvidia GTX680 GPU.

5.7.1 Nomenclature

In the following evaluation we abbreviate the method names according to the following scheme. The first part of the method name is the regularization term and its weighting, followed by an abbreviation of the depth data term and an abbreviation of the intensity data term:

$$\underbrace{\square\square\square\square}_{\text{Weight + Reg.}} - \overbrace{\square\square}^{\text{Depth Term}} - \underbrace{\square\square}_{\text{Intensity Term}}$$

In this context we define the abbreviations for the regularization weighting as

Weighting	Abbreviation
no weighting	-
Scalar Weighting	w
Gradient Weighting	g
Anisotropic Weighing	a

The abbreviations for the regularization term are given by

Regularization	Abbreviation
Total Variation	TV
Huber	H
Total Generalized Variation (2nd order)	TGV

The data term is abbreviated as

Data Term	Abbreviation
classical pixel-wise data term with ℓ_1 norm	*L1*
pixel-wise ℓ_2 term with illumination compensation	*cL1*
patchwise constancy with sum of squared differences	*SSD*
patchwise constancy with sum of absolute differences	*SAD*
patchwise constancy with normalized cross correlation	*NCC*
patchwise constancy with binary census transform	*BCT*
patchwise constancy with ternary census transform	*TCT*
patchwise constancy with closest points in 3D	*CP*

For example a possible combination of our *SF* model is

$$aTGV\text{-}CP\text{-}SAD,$$

which states that the model is composed of an anisotropic *TGV* regularization together with a closest point depth data term and the sum of absolute differences in the intensity data term. Furthermore if we use a patchwise dataterm with multiscale neighborhood we add *(ms)* to the end of the naming.

5.7.2 Evaluation Metrics

In literature there exists a variety of different error measures. In this work we follow the common error measurements of traditional *OF*, namely the End Point Error (EPE) and the Average Angular Error (AAE) as shown in [4], [112] and [129]. These measures can be easily extended to 3D space.

The generally applicable *EPE* is defined as follows:

$$\mathrm{EPE} = \sqrt{\frac{1}{|\Omega|} \sum_{\forall \mathbf{x} \in \Omega} \|\mathbf{u}(\mathbf{x}) - \mathbf{u}_{\mathrm{GT}}(\mathbf{x})\|_2^2}\,, \tag{5.70}$$

where $\mathbf{u}(\mathbf{x})$ and $\mathbf{u}(\mathbf{x})_{\mathrm{GT}}$ are the estimated and groundtruth flow vectors at point $\mathbf{x}$, respectively. The *EPE* measures the absolute flow error in motion magnitude.

The average angular error (*AAE*) measures the error in flow direction. This angular accuracy is defined as angle between the flow vectors by

$$\mathrm{AAE} = \frac{1}{|\Omega|} \sum_{\forall \mathbf{x} \in \Omega} \overset{-1}{\cos} \left(\frac{\tilde{\mathbf{u}}(\mathbf{x})^T \tilde{\mathbf{u}}_{\mathrm{GT}}(\mathbf{x})}{\|\tilde{\mathbf{u}}(\mathbf{x})\|_2 + \|\tilde{\mathbf{u}}_{\mathrm{GT}}(\mathbf{x})\|_2} \right). \tag{5.71}$$

We follow the angle measure proposed in [4], where the vectors are extended by one dimension to $\tilde{\mathbf{u}} = \left[\mathbf{u}^T, 1\right]^T$ and $\tilde{\mathbf{u}}_{\mathrm{GT}} = \left[\mathbf{u}_{\mathrm{GT}}^T, 1\right]^T$ to avoid the *division by zero* problem for zero flows.

The depth images in the Middlebury dataset are derived from given disparity maps.

The focal length and baseline can be freely defined. Since the *EPE* in 3D space is directly dependent on the chosen parameters the general 3D error measures are not directly applicable for this dataset. Hence, We follow Hornáček et al. [49] to calculate the 2D flow errors EPE_{2D} and AAE_{2D} together with the Root Mean Squared Error (RMSE) of disparity deviation, namely the RMS_{Vz}. This disparity deviation is defined as

$$\text{RMS}_{\text{Vz}} = \sqrt{\frac{1}{|\Omega|} \sum_{\forall \mathbf{x} \in \Omega} \left(\frac{bf}{D_1(\mathbf{x}) + \mathbf{u}_D(x)} - \frac{bf}{D_1(\mathbf{x})} \right)^2}, \tag{5.72}$$

where f and b are the defined focal length and the baseline. In other works such as [5, 39, 88, 90] the Normalized Root Mean Square (NRMS) is estimated as global flow error but since the evaluated numbers in their evaluation sections widely differ from each other we waived on using this measure in our evaluation.

5.7.3 Movement Evaluation on Synthetic Data

In this experiment we quantitatively evaluate the proposed *SF* algorithms and show the contributions of the individual terms in our objective functions. We further compare our method to several *SoA OF* and *SF* algorithms. In our experiment we create a synthetic dataset where a cube is rotated and translated in front of a static background. The scene is generated including depth and intensity image pairs as well as a groundtruth *SF*. The illumination is modeled by adding a depth dependent illumination model to the input intensity images. This model is composed of global Gaussian illumination factor and a depth depend illumination factor, where the illumination decreases with the distance to the camera. In Figure 5.14 a sample input sequence of the synthetic scene including the illumination model and the groundtruth motion is shown.

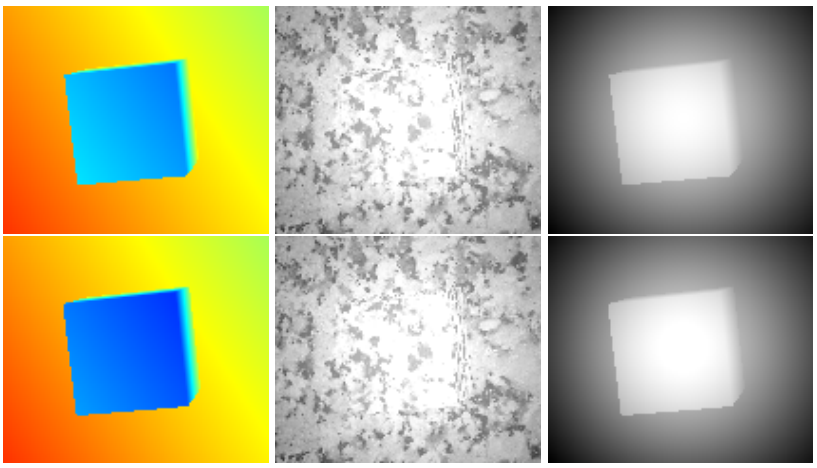
The acquisition noise is simulated with a constant Gaussian noise on the intensity input data and a depth dependent Gaussian noise on the depth data. To show the properties of the different data and regularization terms in our model we measure the accuracy for isolated object movements separately. In this sense, we evaluate the *SF* accuracy for a pure translation of 20% of the object size in X direction (T_X), a pure translation of 20% towards the camera (T_Z) and a rotation of 15 degrees about the Z axis (R_Z). Finally we evaluate the flow estimation for a combined movement and rotation in every direction.

The experiment on the synthetic dataset is divided into two main parts. First, we evaluate the basic data terms for different regularization and weighting terms, as shown in Figure 5.1. Second, we qualitatively evaluate different data term combinations used in our *SF* estimation and compare it to *SoA OF* techniques such as the *OF* estimation of Werlberger et al. [130] and the *Classic-NL-Full* methods of Sun et al. [112]. We further compare to the currently best performing *SF* method of Hornáček et al. [49]. Since the *OF* methods deliver no real 3D motion the 2D motion is projected into 3D space using

the noise-free depth maps. This 3D motion is calculated by

$$\mathbf{u} = \mathbf{K}^{-1}\mathbf{x}^h\mathbf{I}_1^D(\mathbf{x}) - \mathbf{K}^{-1}\left(\mathbf{x} + \mathbf{u}^{OF}\right)^h \mathbf{I}_2^D(\mathbf{x} + \mathbf{u}^{OF}) \tag{5.73}$$

The detailed evaluation is shown in Table 5.2 where the accuracy of the different



(a) Depth $\mathbf{I}_1^D$ / $\mathbf{I}_2^D$ **(b)** Intensity $\mathbf{I}_1^I$ / $\mathbf{I}_2^I$ **(c)** Illumination

Figure 5.12: Flow estimation on synthetic datasets. In (a) the color-coded input depth, in (b) the input intensity images and in (c) the illumination image is shown for the Z-translation example. The groundtruth *OF* and the results for this pure Z movement are shown in (c).

	$T_X = 20\%$		$T_Z = -20\%$		$R_Z = 15°$		$T = [19\%]^3, R = [15°, 15°, 15°]$		Avg.Time [s]
	EPE_{3D}	AAE_{3D}	EPE_{3D}	AAE_{3D}	EPE_{3D}	AAE_{3D}	EPE_{3D}	AAE_{3D}	
TV-L1-L1	0.2266	9.479	0.4884	8.576	0.2098	8.694	0.2498	7.970	3.02
H-L1-L1	0.2281	9.515	0.4853	8.394	0.2105	8.724	0.2472	7.921	3.00
TGV-L1-L1	0.2520	10.243	0.4674	7.756	0.2024	8.480	0.2417	7.973	3.12
wTV-L1-L1	0.2108	8.894	0.1950	7.166	0.1968	8.113	0.2066	8.136	3.09
wH-L1-L1	0.2157	8.997	0.1965	7.156	0.1936	8.044	0.2073	8.147	3.15
wTGV-L1-L1	0.2130	8.898	0.1940	7.180	0.1879	7.999	0.2037	8.198	3.32
gTV-L1-L1	0.2125	8.985	0.1884	6.943	0.1847	7.958	0.1984	8.016	3.22
gH-L1-L1	0.2163	9.106	0.1886	6.889	0.1910	8.056	0.2010	8.040	3.28
gTGV-L1-L1	0.2065	8.903	**0.1837**	6.908	0.1801	**7.777**	0.1913	7.908	3.36
aTV-L1-L1	0.2110	9.072	0.1878	6.876	0.1801	7.847	0.1926	7.940	3.36
aH-L1-L1	0.2099	9.030	0.1870	**6.846**	0.1808	7.877	0.1932	7.915	3.20
aTGV-L1-L1	**0.2026**	**8.846**	0.1848	6.882	**0.1783**	7.816	**0.1845**	**7.796**	3.55

Table 5.1: Scene Flow performance of different regularization terms on synthetic datasets. Comparison of different regularization/weighting terms at different object movements in terms of EPE and AAE in 3D. The best results for each error metric are highlighted.

	$T_X = 20\%$		$T_Z = -20\%$		$R_Z = 15°$		$T = [10\%]^3, R = [5°]^3$		Avg.Time [s]
	EPE_{3D}	AAE_{3D}	EPE_{3D}	AAE_{3D}	EPE_{3D}	AAE_{3D}	EPE_{3D}	AAE_{3D}	
NL-TV-NCC (*OF*)	0.282	5.61	0.191	3.07	0.291	5.06			
Classic++ (*OF*)	0.353	6.16	0.176	3.27	0.322	5.54			
Classic+NL-Full (*OF*)	0.260	4.57	0.303	3.29	0.388	5.68			
*Horn*áček et al. [49]	0.089	3.85	0.090	3.30	0.056	2.43			
aTV-L1-L1	0.2110	9.072	0.1878	6.876	0.1801	7.847	0.1926	7.940	3.15
aH-L1-L1	0.2099	9.030	0.1870	6.846	0.1808	7.877	0.1932	7.915	3.24
aTGV-L1-L1	0.2026	8.846	0.1848	6.882	0.1783	7.816	0.1845	7.796	3.33
aTGV-L1-SSD	1.5902	49.261	0.1016	3.342	0.3216	13.551	0.9290	35.841	3.14
aTGV-L1-SAD	0.0494	1.977	0.0423	0.710	0.1992	6.834	0.1051	1.372	3.31
aTGV-L1-NCC	0.0444	1.726	0.0426	0.681	0.0570	1.358	0.1055	1.472	3.52
aTGV-L1-C	0.0405	1.574	0.0439	0.713	0.0743	1.843	0.1021	1.563	3.36
aTGV-L1-TCT	0.0425	1.663	0.0433	0.673	0.0705	1.745	0.1005	1.509	3.14
aTGV-CP-SSD	4.6599	65.905	10.2481	83.657	2.5805	55.457	6.5689	64.812	3.92
aTGV-CP-SAD	0.0445	1.802	0.0396	0.714	0.0429	1.182	0.0630	1.261	4.08
aTGV-CP-NCC	0.0418	1.590	0.0403	**0.658**	**0.0382**	**1.116**	0.0641	1.314	3.85
aTGV-CP-C	0.0330	1.217	0.0432	0.765	0.0432	1.279	0.0645	1.348	3.71
aTGV-CP-TCT	0.0333	1.241	0.0421	0.704	0.0452	1.357	0.0638	1.238	3.93
aTGV-CP-SAD(ms)	0.0470	1.926	**0.0395**	0.698	0.0397	1.177	0.0636	1.297	3.88
aTGV-CP-NCC(ms)	0.0356	1.348	0.0398	0.673	0.0457	1.157	**0.0627**	**1.157**	4.15
aTGV-CP-C(ms)	0.0362	1.331	0.0445	0.876	0.0496	1.544	0.0653	1.373	3.92
aTGV-CP-TCT(ms)	**0.0330**	**1.196**	0.0427	0.784	0.0492	1.454	0.0639	1.266	3.84

Table 5.2: Scene Flow evaluation on a synthetic dataset. Comparison of our method with *SOTA* *OF* and *SF* methods at different object movements in terms of EPE and AAE in 3D. Further, results of are shown, where different terms are turned off. The best result for each movement is highlighted and the second best is underlined.

Discussion

What can be clearly seen in the evaluation of the weighing terms in Table 5.1 is that the weighting clearly improves the optimization result. Further, gradient directed weighting and anisotropic weighing outperform the simple scalar weighting by far, while the anisotropic weighting is slightly more accurate than the gradient weighting. Furthermore, gradient and anisotropic tensors have a big impact on the quality since they bound the flow field directly at object boundaries.

In terms of regularization, the Huber and the *TGV* regularization deliver a much higher accuracy than the standard *TV* regularization. Further, it can be seen that the *TGV* performs best in this experiment due to the higher order regularization. While the first order approaches work well for pure translational movements they are not suitable to model smooth flow transitions such as rotations or non-rigid movements since these approaches enforce piecewise constant solutions in the flow field.

In Table 5.2 the comparison of the standard pixelwise data terms and the patchwise variants is shown. Using a patchwise data term brings a huge boost in performance, since the comparison on a larger region is more distinctive than a comparison on the pixel level.

The closest point depth data term improves the performance over a pixelwise *L1* depth data term in every variation. When comparing the variations of the depth data term, the *SSD* term has the worst performance due to quadratic penalization of the patch difference. Further, the illumination invariant Census and *NCC* models show a slightly better overall performance compared to the *SAD* term. However, in this experiment the multiscale patchwise terms do not improve over the single scale methods since the dataset includes enough measurable texture in every part of the image.

Compared to *SoA* approaches it can be seen that our *SF* estimation significantly outperforms traditional *OF* methods due to the additional depth information in our model. The *RGB-D* PatchMatch approach of Hornáček et al. [49] delivers comparable results for the flow magnitude (*EPE*) but lacks in angular precision (*AAE*).

5.7.4 Noise Evaluation

In this section we evaluate our method according to noise in the input data. Hence we generate a synthetic dataset and apply different levels of noise. We use the same moving object as in 5.7.3, which is randomly moved and rotated in all directions, as shown in Figure 5.17. On the synthetically generated dataset we add zero-mean Gaussian noise $N_I(0, \sigma)$ to the input intensity images. According to most depth sensor characteristics we also add depth dependent zero-mean Gaussian noise $N_D(0, \sigma(d_{\max} - d))$ to the depth input. The standard deviation σ for the added noise ranges from $0 - 50\%$ of the input intensity range.

To create a meaningful evaluation of the aforementioned methods we calculate the average error over 100 random translation/rotation combinations. These random translations range from -20% to $+20\%$ of the object size in X, Y and Z direction. The rotation

range is between -15 and $+15$ degrees about the X, Y and Z axis. As in the previous experiment the error is measured as *EPE* and *AAE* in 3D space. In all estimations over all methods we explicitly use the same initial flow parameters such as weightings and tensors. In Figure 5.17 the averaged results over all noise levels are shown.

Discussion

Overall, it can be seen that data terms with pixelwise penalization (marked as dashed lines) perform significantly worse than data terms with a more complex patchwise comparison. Obviously, the bounding anisotropic tensor has a big impact on the accuracy when comparing the anisotropic models (*aTGV-L1-L1*, *aTGV-CP-TCT*) to the unbounded models (*TGV-L1-L1*, *TGV-CP-TCT*). Further, the *TGV* regularization works slightly better than the *TV* regularization in terms of the overall error. A simple *SSD* as an intensity data term performs much worse than the more robust terms *NCC* and *TCT*, which have similar performance and only differ in runtime.

5.7.5 Middlebury Evaluation

In order to perform a quantitative comparison to other *SoA SF* methods, we evaluate our method using an existing scene flow benchmark dataset. We follow [5, 39, 49, 52, 87, 88, 90, 138], which use the rectified stereo intensity and disparity maps from the Middlebury *Cones*, *Teddy* and *Venus* datasets [102] to simulate scene flow. In this setting, two images are acquired with a pure horizontal camera movement. This allows to recover a ground truth scene motion at every point in a cluttered scene with pure X movement, where the 3D movement in Y and Z direction is zero and the movement in X-direction is given by the baseline. As in the compared methods, the disparity maps $\mathbf{H}_1$ and $\mathbf{H}_2$ are used to simulate the output of the depth sensor. The depth is calculated with a defined focal length f and a baseline b with

$$\mathbf{I}_i^D = \frac{fb}{\mathbf{H}_i}, \qquad \forall i = 1, 2. \tag{5.74}$$

Throughout all datasets we set $f = 150$ and $b = 50$. The calculated *SF* is backprojected into the image space for a direct comparison with the ground truth disparity maps. In Table 5.3 the quantitative accuracy results of our evaluation are shown. In the Figures 5.18, 5.19 and 5.20 the three Middlebury datasets are visually evaluated. A colored map is shown according to the flow error in a range of $[-1, +1]$ pixel. In our evaluation we compare the different methods to several *SoA* methods for *OF*, *SF* from stereo and *SF* from RGB-D data. We evaluate the different methods according to magnitude and angular error in 2D space as well as a disparity error in 3D space, as shown in section 5.7.2. We deliberately use the same parameters for all three datasets even though the Venus dataset has other lighting and surface conditions.

(a) *Groundtruth* **(b)** *TV-L1-L1* **(c)** *H-L1-L1* **(d)** *TGV-L1-L1*

(e) *wTV-L1-L1* **(f)** *wH-L1-L1* **(g)** *wTGV-L1-L1*

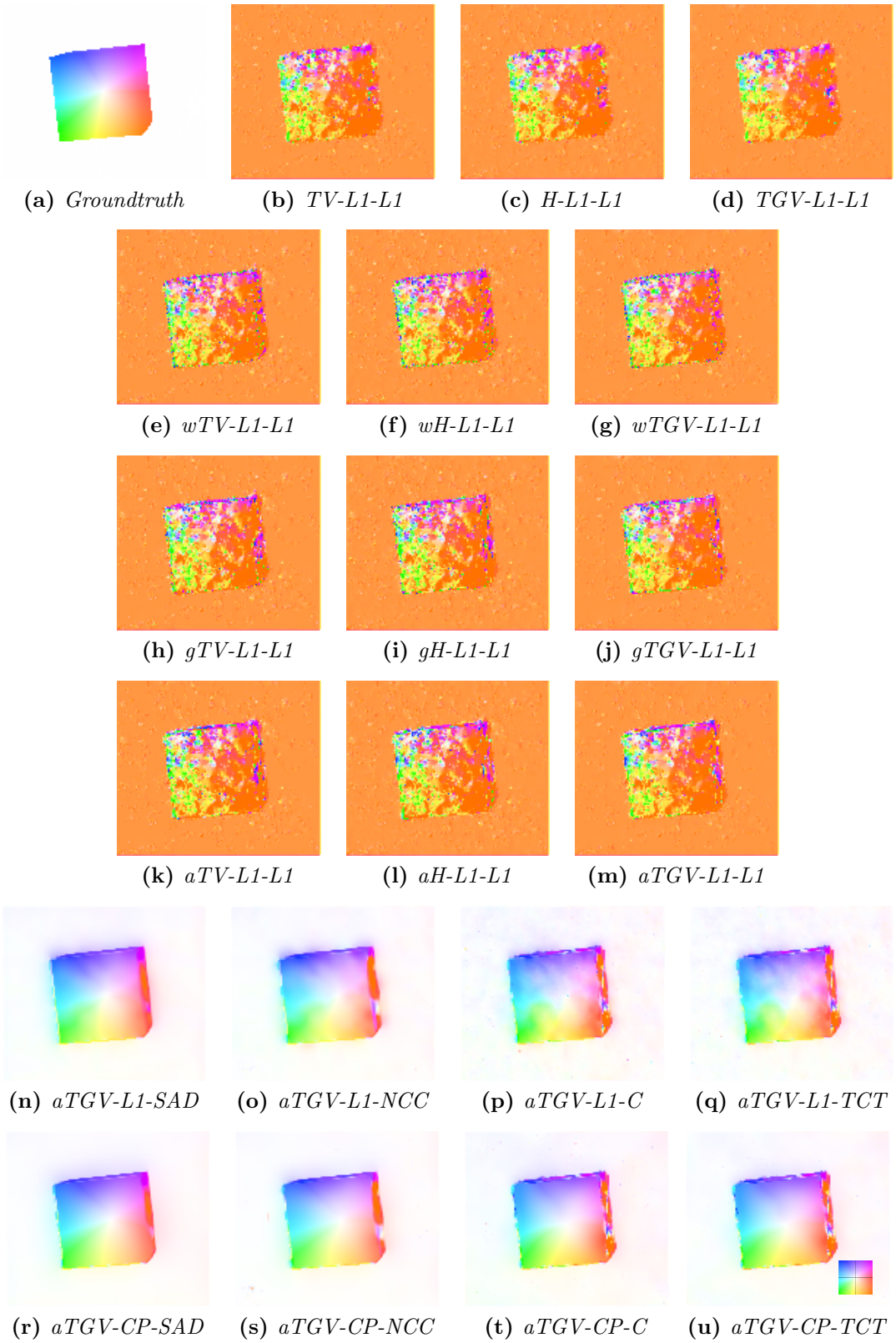
(h) *gTV-L1-L1* **(i)** *gH-L1-L1* **(j)** *gTGV-L1-L1*

(k) *aTV-L1-L1* **(l)** *aH-L1-L1* **(m)** *aTGV-L1-L1*

(n) *aTGV-L1-SAD* **(o)** *aTGV-L1-NCC* **(p)** *aTGV-L1-C* **(q)** *aTGV-L1-TCT*

(r) *aTGV-CP-SAD* **(s)** *aTGV-CP-NCC* **(t)** *aTGV-CP-C* **(u)** *aTGV-CP-TCT*

Figure 5.13: Flow estimation on a synthetic dataset for pure X translation. The color key is shown on the bottom right.



(a) *Groundtruth* **(b)** *TV-L1-L1* **(c)** *H-L1-L1* **(d)** *TGV-L1-L1*

(e) *wTV-L1-L1* **(f)** *wH-L1-L1* **(g)** *wTGV-L1-L1*

(h) *gTV-L1-L1* **(i)** *gH-L1-L1* **(j)** *gTGV-L1-L1*

(k) *aTV-L1-L1* **(l)** *aH-L1-L1* **(m)** *aTGV-L1-L1*

(n) *aTGV-L1-SAD* **(o)** *aTGV-L1-NCC* **(p)** *aTGV-L1-C* **(q)** *aTGV-L1-TCT*

(r) *aTGV-CP-SAD* **(s)** *aTGV-CP-NCC* **(t)** *aTGV-CP-C* **(u)** *aTGV-CP-TCT*

Figure 5.14: Flow estimation on a synthetic dataset for pure Z translation. The color key is shown on the bottom right.

(a) *Groundtruth* **(b)** *TV-L1-L1* **(c)** *H-L1-L1* **(d)** *TGV-L1-L1*

(e) *wTV-L1-L1* **(f)** *wH-L1-L1* **(g)** *wTGV-L1-L1*

(h) *gTV-L1-L1* **(i)** *gH-L1-L1* **(j)** *gTGV-L1-L1*

(k) *aTV-L1-L1* **(l)** *aH-L1-L1* **(m)** *aTGV-L1-L1*

(n) *aTGV-L1-SAD* **(o)** *aTGV-L1-NCC* **(p)** *aTGV-L1-C* **(q)** *aTGV-L1-TCT*

(r) *aTGV-CP-SAD* **(s)** *aTGV-CP-NCC* **(t)** *aTGV-CP-C* **(u)** *aTGV-CP-TCT*

Figure 5.15: Flow estimation on a synthetic dataset for a rotation about the Z axis. The color key is shown on the bottom right.

(a) *Groundtruth* **(b)** *TV-L1-L1* **(c)** *H-L1-L1* **(d)** *TGV-L1-L1*

(e) *wTV-L1-L1* **(f)** *wH-L1-L1* **(g)** *wTGV-L1-L1*

(h) *gTV-L1-L1* **(i)** *gH-L1-L1* **(j)** *gTGV-L1-L1*

(k) *aTV-L1-L1* **(l)** *aH-L1-L1* **(m)** *aTGV-L1-L1*

(n) *aTGV-L1-SAD* **(o)** *aTGV-L1-NCC* **(p)** *aTGV-L1-C* **(q)** *aTGV-L1-TCT*

(r) *aTGV-CP-SAD* **(s)** *aTGV-CP-NCC* **(t)** *aTGV-CP-C* **(u)** *aTGV-CP-TCT*

Figure 5.16: Flow estimation on a synthetic dataset for a rotation of 5 degrees about X, Y and Z and a translation of $t = [10\%, 10\%, -10\%]^T$ of the object size. The color key is shown on the bottom right.

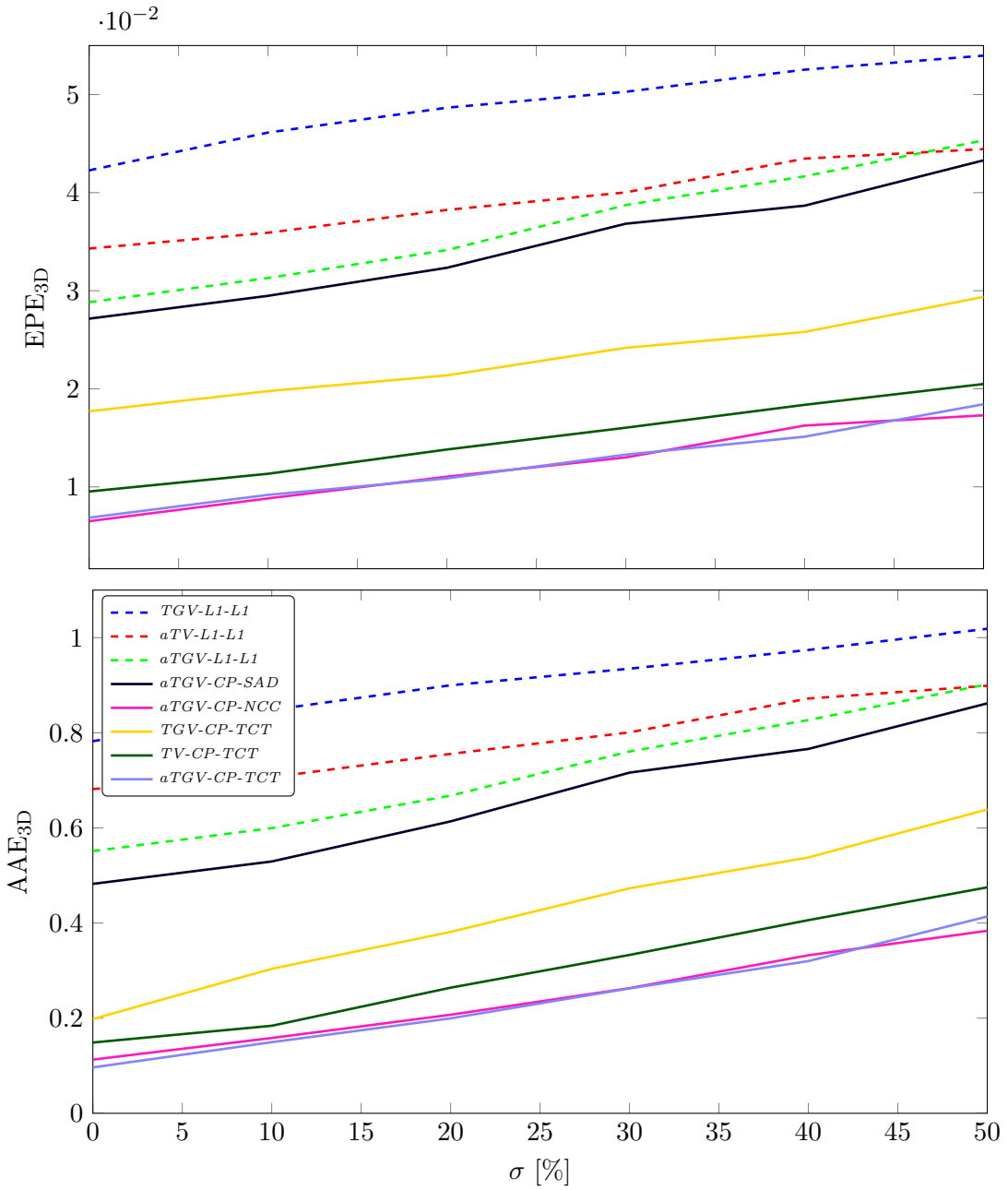
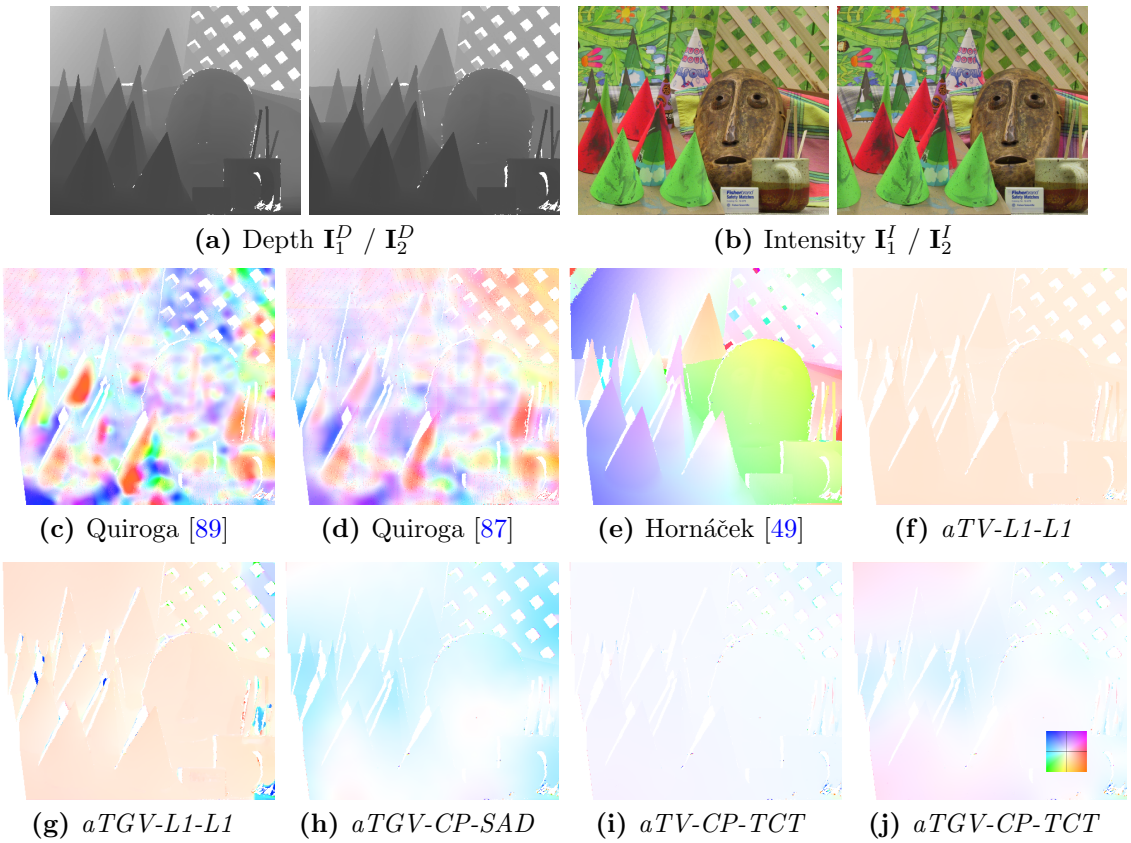


Figure 5.17: Noise Evaluation. Robustness evaluation of different *SF* models under increasing input noise. The error is measured as *EPE* and *AAE* in 3D space.

	Cones			*Teddy*			*Venus*			Avg.Time [s]
	EPE_{2D}	RMS_{Vz}	AAE_{2D}	EPE_{2D}	RMS_{Vz}	AAE_{2D}	EPE_{2D}	RMS_{Vz}	AAE_{2D}	
Brox and Malik [17] (*OF*)	2.83	1.75	1.75	3.20	0.47	0.39	0.72	0.14	1.28	-
Xu et al. [131] (*OF*)	1.66	1.15	0.21	1.70	0.50	0.28	0.30	0.22	1.43	-
Basha et al. [5] (2 views) (*ST*)	0.58	N/A	0.39	0.57	N/A	1.01	0.16	N/A	1.58	-
Huguet and Devernay [52] (*ST*)	1.10	N/A	0.69	1.25	N/A	0.51	0.31	N/A	0.98	18000
Hadfield and Bowden [40]	1.24	0.06	1.01	0.83	0.03	0.83	0.36	0.02	1.03	600
Zhang et al. [138]	1.04	0.69	0.73	0.69	0.15	1.15	-	-	-	-
Quiroga et al. [88] (local)	3.50	2.22	4.30	-	-	-	-	-	-	5s
Quiroga et al. [90]	0.81	0.04	0.36	0.90	0.04	0.69	0.20	**0.00**	0.98	10
Quiroga et al. [87]	0.75	0.02	0.30	0.65	0.02	0.37	0.16	**0.00**	0.77	60
*Horn*áček et al. [49]	0.54	0.02	0.52	0.35	0.01	0.16	0.26	0.02	0.64	-
TGV-L1-L1	0.48	0.09	0.27	0.23	0.03	0.38	0.18	**0.00**	0.85	1.48
aTV-L1-L1	0.20	**0.01**	0.27	0.18	**0.00**	0.30	0.16	**0.00**	0.70	**1.37**
aTGV-L1-L1	0.49	0.10	0.28	0.23	0.03	0.38	0.18	**0.00**	0.85	2.16
aTGV-L1-SAD	0.34	**0.01**	0.06	0.32	**0.00**	0.08	0.52	**0.00**	0.73	2.47
aTGV-CP-SAD	0.23	**0.01**	**0.02**	0.21	**0.00**	0.05	0.52	**0.00**	0.73	2.47
aTGV-CP-NCC	0.11	**0.01**	0.07	0.14	**0.00**	0.06	0.14	**0.00**	0.49	2.74
TGV-CP-TCT	0.11	**0.01**	0.07	0.13	**0.00**	0.06	0.15	**0.00**	0.49	2.20
aTV-CP-TCT	**0.08**	**0.01**	0.05	**0.09**	**0.00**	**0.01**	**0.11**	**0.00**	**0.15**	2.08
aTGV-CP-TCT	0.11	**0.01**	0.07	0.13	**0.00**	0.06	0.15	**0.00**	0.49	2.61

Table 5.3: Quantitative evaluation on the Middlebury dataset. The error is measured by EPE / AAE in 2D, and RMS in disparity change error. The best result for each dataset is highlighted and the second best is underlined. The topmost methods indicated with (*OF*) are *RGB* optical flow algorithms where the RMS_{Vz} is computed according to the mapped disparity given the 2D flow vector. Methods that calculate SF from stereo are marked with (*ST*). Note that the method [88] is a local method which only covers $\sim 50\%$ of the flow field.



(a) Depth $\mathbf{I}_1^D$ / $\mathbf{I}_2^D$ **(b)** Intensity $\mathbf{I}_1^I$ / $\mathbf{I}_2^I$

(c) Quiroga [89] **(d)** Quiroga [87] **(e)** Horňáček [49] **(f)** *aTV-L1-L1*

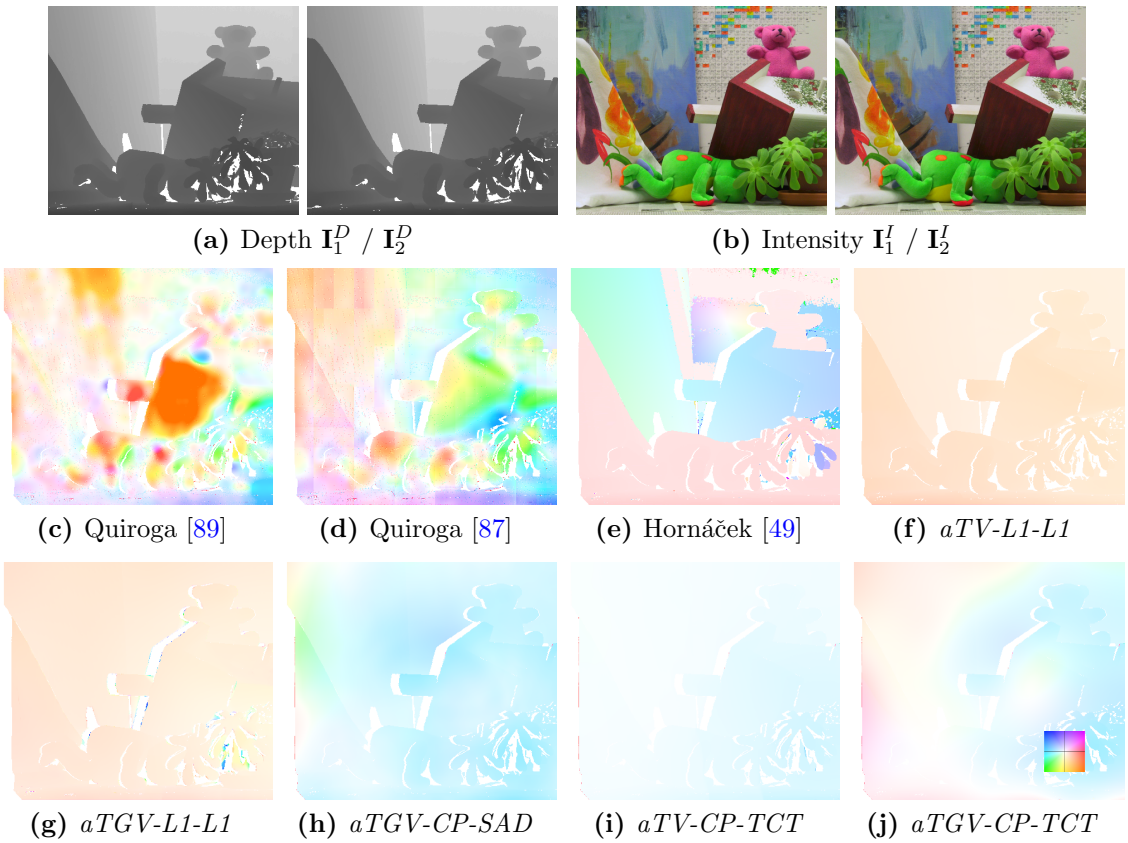
(g) *aTGV-L1-L1* **(h)** *aTGV-CP-SAD* **(i)** *aTV-CP-TCT* **(j)** *aTGV-CP-TCT*

Figure 5.18: Qualitative evaluation on the Middlebury Cones dataset. The 2D flow error is shown as color map in a range of $[-1, +1]$ pixel for all datasets and all methods. The color intensity depicts the amount of error. The error direction is encoded by the color itself. The color key is shown on the bottom right.

Discussion

What can be clearly seen is that the proposed models clearly improve over *SoA*. Similar to the last experiments, it can be seen that the patchwise dataterms get better results than methods with a pixelwise comparison in the data fidelity. In both qualitative and quantitative evaluations the anisotropic CP-TCT with simple TV regularization compared to *TGV* produces more accurate results. This is caused by the property of the Middlebury dataset. In this dataset only a constant translational movement in X direction appears without any flow transitions, which is perfectly suited for the TV case but probably will not appear in real world scenes.

Although we achieve superior results, we have to mention that the Middlebury sequence is not very well suited to evaluate *SF* algorithms, since it only incorporates a static movement in X direction without any movement in Y or Z direction. This does not



(a) Depth $\mathbf{I}_1^D$ / $\mathbf{I}_2^D$ **(b)** Intensity $\mathbf{I}_1^I$ / $\mathbf{I}_2^I$

(c) Quiroga [89] **(d)** Quiroga [87] **(e)** Horn\u00e1\u010dek [49] **(f)** *aTV-L1-L1*

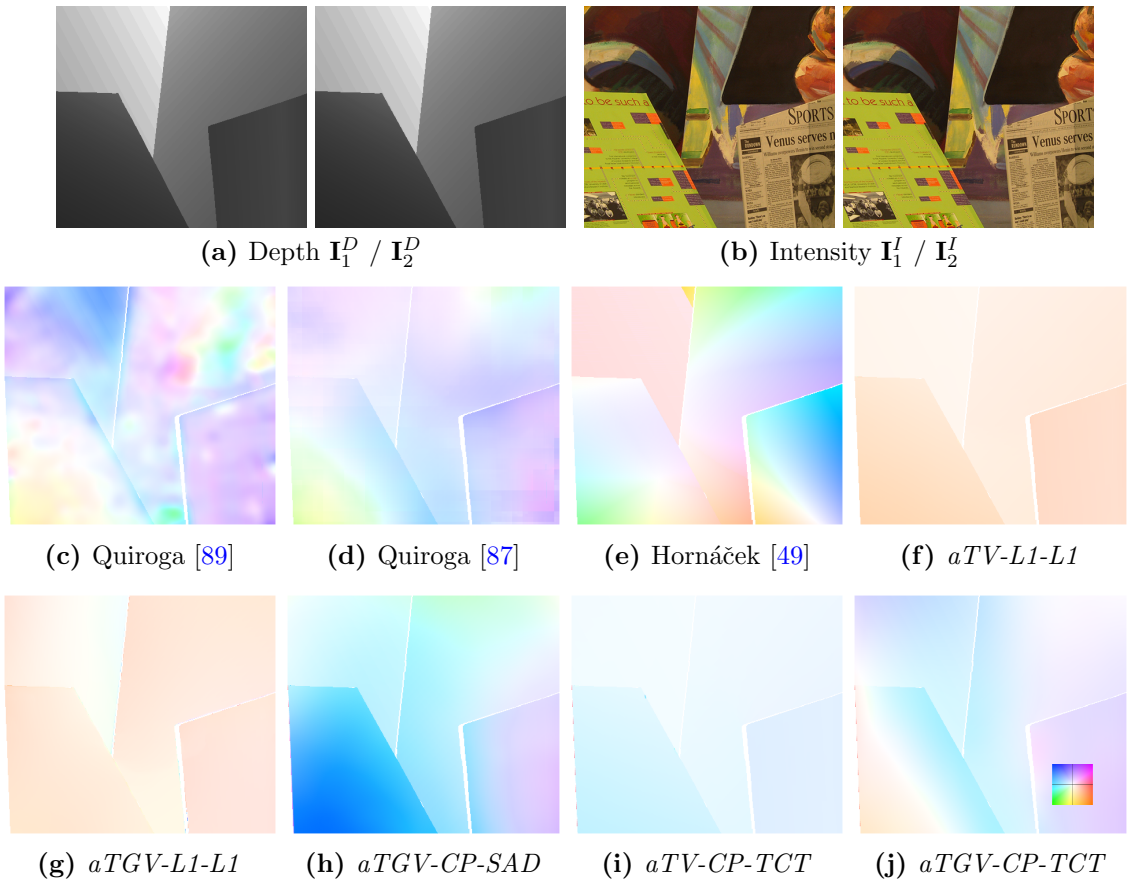
(g) *aTGV-L1-L1* **(h)** *aTGV-CP-SAD* **(i)** *aTV-CP-TCT* **(j)** *aTGV-CP-TCT*

Figure 5.19: Qualitative evaluation on the Middlebury Teddy dataset. The 2D flow error is shown as color map in a range of $[-1, +1]$ pixel for all datasets and all methods. The color intensity depicts the amount of error. The error direction is encoded by the color itself. The color key is shown on the bottom right.

show the advantages of higher order regularization or gradient weighting.

5.7.6 Real-World Evaluation

In this section we show a visual evaluation of *SF* estimation methods used for real *ToF* sensors. Hence, we apply the proposed methods to the measurement output of different sensors such as the a PMD Nano *ToF* camera and a Microsoft Kinect for Windows v2 camera (K4Wv2). In the acquisition setting we used rigid and non-rigid objects moving freely in the environment. The scenes are chosen to contain reflecting parts, changing illumination and shadows. The Microsoft K4Wv2 comes equipped with an additional intensity camera. The output of this sensor is used for the intensity data term in our model. Since the PMD Nano *ToF* camera does not offer a separate intensity image we use the amplitude image from the *ToF* sensor as intensity data input. The *ToF* sensor uses active infrared illumination. Hence the scene is not equally illuminated which causes a



(a) Depth $\mathbf{I}_1^D$ / $\mathbf{I}_2^D$ **(b)** Intensity $\mathbf{I}_1^I$ / $\mathbf{I}_2^I$

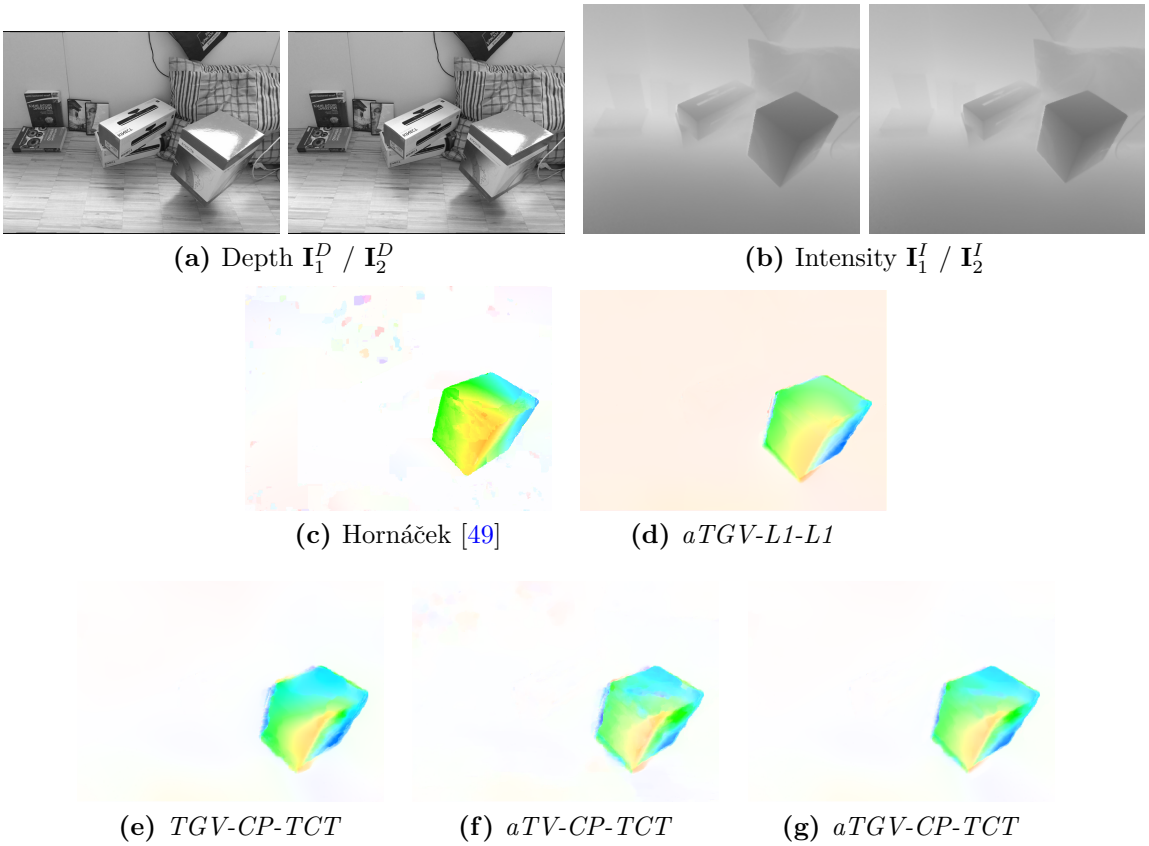
(c) Quiroga [89] **(d)** Quiroga [87] **(e)** Hornáček [49] **(f)** *aTV-L1-L1*

(g) *aTGV-L1-L1* **(h)** *aTGV-CP-SAD* **(i)** *aTV-CP-TCT* **(j)** *aTGV-CP-TCT*

Figure 5.20: Qualitative evaluation on the Middlebury Venus dataset. The 2D flow error is shown as color map in a range of $[-1, +1]$ pixel for all datasets and all methods. The color intensity depicts the amount of error. The error direction is encoded by the color itself. The color key is shown on the bottom right.

change in the reflected illumination from objects during a movement. To correctly estimate the metric flow we used our camera calibration toolbox from [30].

The visual evaluation for the three different datasets is shown in Figures 5.21, 5.22 and 5.23. In Figure 5.21 we show a scene with moving objects acquired with the Microsoft K4Wv2. The scene contains a global illumination with highly reflective object parts and shadows. In the second scene we acquired a moving box with the PMD Nano camera in Figure 5.22. One big challenge in this scene is that the depth acquisitions contain a high amount of noise. Furthermore, due to the direct illumination the reflected intensity decreases with the distance from the image center and the distance from the camera. In the third scene we acquired a moving hand with the PMD Nano camera to evaluate our methods under non-rigid movements.



(a) Depth $\mathbf{I}_1^D$ / $\mathbf{I}_2^D$ **(b)** Intensity $\mathbf{I}_1^I$ / $\mathbf{I}_2^I$

(c) Hornáček [49] **(d)** *aTGV-L1-L1*

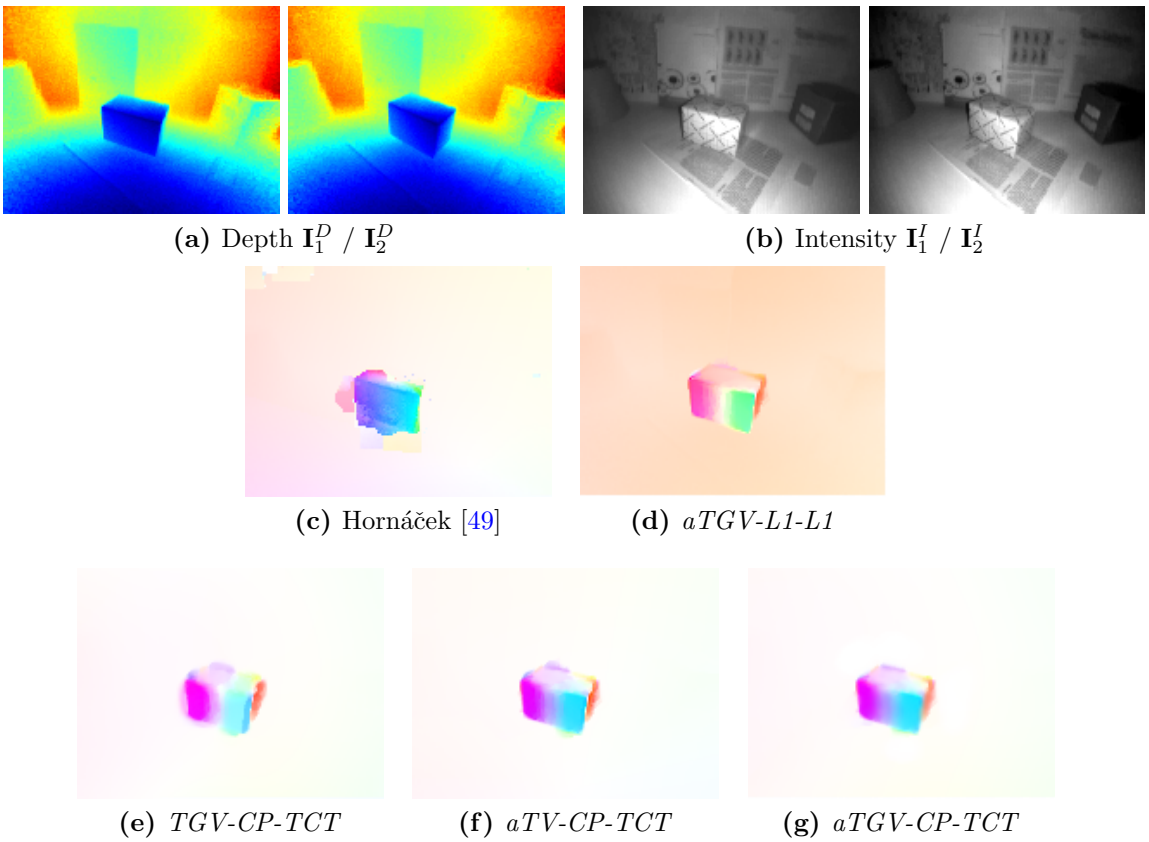
(e) *TGV-CP-TCT* **(f)** *aTV-CP-TCT* **(g)** *aTGV-CP-TCT*

Figure 5.21: Evaluation of different *SF* methods on real image sequences. The dataset contains a rotated and translated box acquired with the K4Wv2.

Discussion

This qualitative evaluation on real sensors clearly points out the properties of the different methods. Compared to the noise-free Middlebury datasets, real sensor acquisitions contain a high amount of noise. Furthermore, the intensity images contain specular reflections, shadows or illumination artifacts due to the active illumination. Due to these less than ideal properties an accurate *SF* estimation becomes more difficult. In the real-world evaluations it can be seen that pure pixelwise comparisons in the data terms such as *aTGV-L1-L1* clearly lack in precision. Methods based on a patch-wise comparison deliver much more accurate results due to the improved robustness to noise and illumination artifacts.

We further compared *TV* and *TGV* regularization. While the optimization result using *TV* regularization suffers from staircasing the optimization result from *TGV* models delivers more smooth results due to the higher order flow modeling. With the weighting of the regularization with an anisotropic tensor, the quality of the flow estimate further increases. This can be seen at flow boarders where models without edge weighting such



(a) Depth $\mathbf{I}_1^D$ / $\mathbf{I}_2^D$ **(b)** Intensity $\mathbf{I}_1^I$ / $\mathbf{I}_2^I$

(c) Hornáček [49] **(d)** *aTGV-L1-L1*

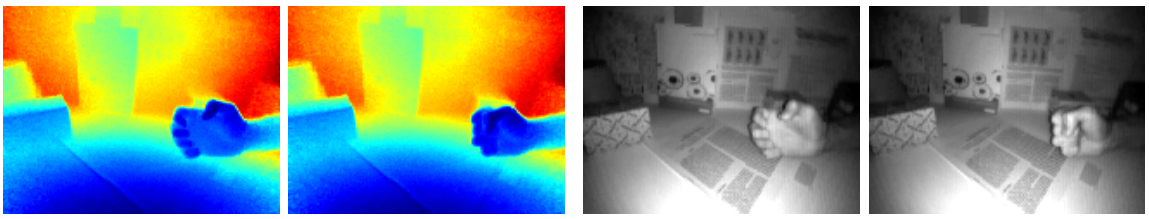
(e) *TGV-CP-TCT* **(f)** *aTV-CP-TCT* **(g)** *aTGV-CP-TCT*

Figure 5.22: Evaluation of different *SF* methods on real image sequences. The dataset contains a rotated box acquired with the PMD Nano Camera.

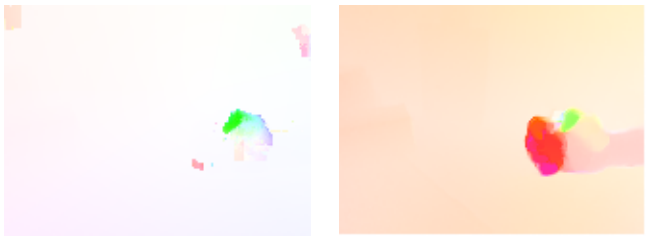
as *TGV-CP-TCT0* are blurry while models using an anisotropic weighting deliver much sharper motion boundaries. Beyond that, we compare our models with the *SoA* scene flow estimation *SphereFlow* proposed by Hornáček et al. [49]. This model delivers a high precision at noise-free datasets but has big problems to estimate the *SF* at larger noise levels or illumination changes, which appear at the PMD Nano sequences.

5.8 Applications

Fast and accurate *SF* estimation has many potential computer vision applications. In this section we will present two applications of our *SF* estimation method on real-world data. In 5.8.1 we show how to estimate the camera pose in a static scene without explicitly building a model of the scene. In 5.8.2 we use our model to increase the lateral resolution of a depth image by moving an object in front of an observing depth camera.



(a) Depth $\mathbf{I}_1^D$ / $\mathbf{I}_2^D$ **(b)** Intensity $\mathbf{I}_1^I$ / $\mathbf{I}_2^I$



(c) Hornáček [49] **(d)** *aTGV-L1-L1*

(e) *TGV-CP-TCT* **(f)** *aTV-CP-TCT* **(g)** *aTGV-CP-TCT*

Figure 5.23: Evaluation of different *SF* methods on real image sequences. The dataset contains a hand movement acquired with the PMD Nano Camera.

5.8.1 Camera Pose Estimation

An accurate *SF* allows to estimate the pose of a moving camera in space. Since our method calculates metric *SF* we can directly estimate the movement of each scene point from one frame to the next. Compared to traditional camera pose estimation this approach has the big advantage that a feature calculation as well as a robust feature matching is no longer necessary. The point correspondences are directly taken from the estimated flow field. This reduces the computational effort and therefore leads to higher frame rates.

Given the estimated flow field $\mathbf{u}$ between two consecutive frames $t = \{1, 2\}$, we can establish corresponding point sets $\mathbf{X}_1$ and $\tilde{\mathbf{X}}_2 = \mathbf{X}_1 + \mathbf{u}$, according to (5.6). As in traditional pose estimation, the general rotation $R_1 \in \mathrm{SO}(3)$ and translation $T_1 \in \mathbb{R}^3$ is calculated by Euclidean motion estimation as

$$\min_{R_1, T_1} \left(R_1 \mathbf{X}_1 + T_1 - \tilde{\mathbf{X}}_2 \right)^2. \tag{5.75}$$

The camera pose is updated by $P_2 = P_1 [R_1 | T_1]^{-1}$, where P_1 and P_2 are the camera poses.

For multiple frames this pose estimation can be propagated by

$$P_{t+1} = P_{t_1} \bigcap_{i=t_1}^{t} [R_i|T_i]^{-1}, \tag{5.76}$$

where t_1 denotes the first frame.

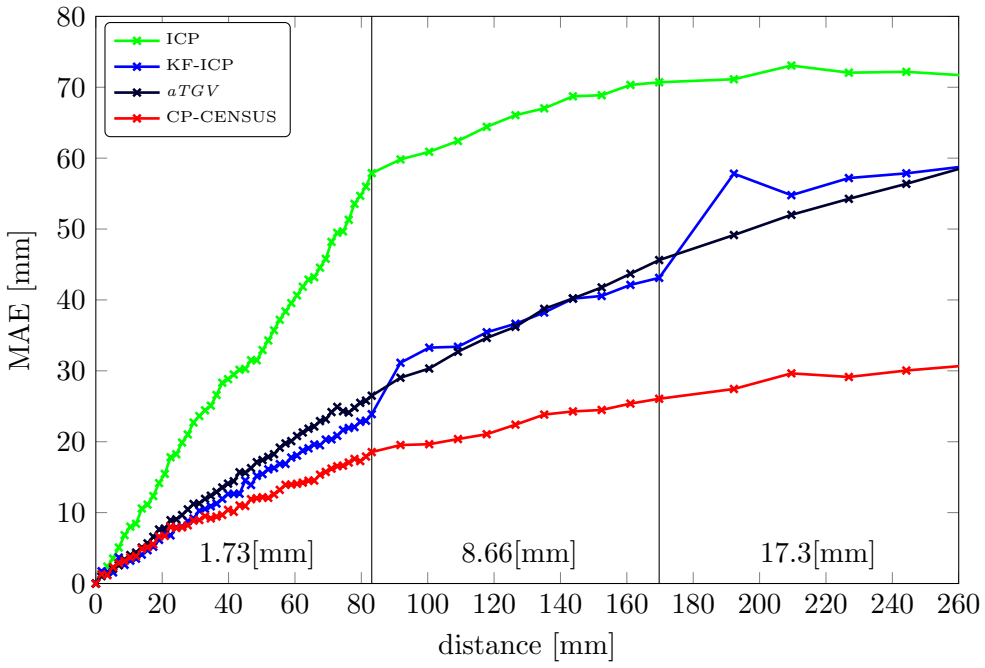


Figure 5.24: Evaluation of camera pose estimation using *aTGV-CP-TCT* and the method using only pixelwise dataterms *aTGV* compared to standard *ICP* and the model based *ICP* (Kinect Fusion (KFusion)). The error is given by means of *RMSE* between real and estimated camera pose in [mm] for a relative distance between two consecutive frames of 1.73, 8.66 and 17.32mm.

For the numerical evaluation of the camera pose estimation, we use the PMD Nano *ToF* camera mounted on the head of an industrial robot. We use the real scene from Fig. 5.22 (2^{nd} row) acquired by different, known camera poses instead of moving the objects. The robot moves 260mm in a linear movement in positive X, Y and Z direction with a distance of 1.73, 8.66 and 17.32mm between consecutive acquisitions. The estimation accuracy is compared to standard *ICP* with 100 iterations and to a model based multi-scale *ICP* as proposed in the *KFusion* framework [76].

The camera position of the first frame P_{t_1} is defined as the world coordinate center defined as the initial position of the robot head. The error is quantified as the metric difference between the accumulated camera poses P_t and the known robot poses is calculated in terms of *RMSE*. The propagated average movement error is shown in Fig. 5.24 and the *RMSE* per mm movement separated in $X/Y/Z$ direction is shown in Table 5.4.

	X [mm]	Y [mm]	Z [mm]
RMSE - *ICP*	0.203	0.109	0.176
RMSE - *KFusion*	0.049	0.135	0.085
RMSE - *aTGV*	**0.031**	0.074	0.149
RMSE - aTGV-CP-TCT	0.049	**0.062**	**0.069**

Table 5.4: Camera Pose Estimation using different methods. The positioning error of the different methods used for camera pose estimation is measured as *RMSE* in [mm]. The best result for each axis is highlighted.

The relative rotation error in the pose estimation is below 0.8 degrees for all three methods. Because we assume a static scene, the *SF* computation is accelerated by reducing the number of levels and iterations per level resulting in an average frame rate of 12.2fps.

Discussion

In the error statistics of mean and relative error it can be seen that CP-TCT clearly outperforms the compared methods *ICP*, *KFusion* and the pixelwise flow estimation *aTGV* used for camera pose estimation. Further, the relative error of *aTGV-CP-TCT* is not dependent on the movement magnitude compared to *ICP*, which has a higher error for smaller steps. The *KFusion* method builds a model of the scene, which evolves over time. In contrast, a big advantage of camera pose estimation based on *SF* methods is to estimate a frame-to-frame pose propagation over the whole sequence without the need of an explicit model, which makes them also applicable on dynamic scenes.

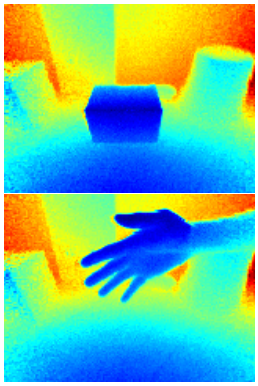
5.8.2 Superresolution

Similar to the camera pose estimation, *SF* can also be used for depth superresolution of a scene. each acquisition back into the first frame, according to the estimated flow field. In this experiment we show how our *SF* estimation is used for depth superresolution of freely moving objects in a scene. Therefore, we compute the *SF* for consecutive depth and intensity image pairs in a sequence of T frames. The point set of each acquisition is then back propagated into the first frame solely through the *SF* vectors at each point by

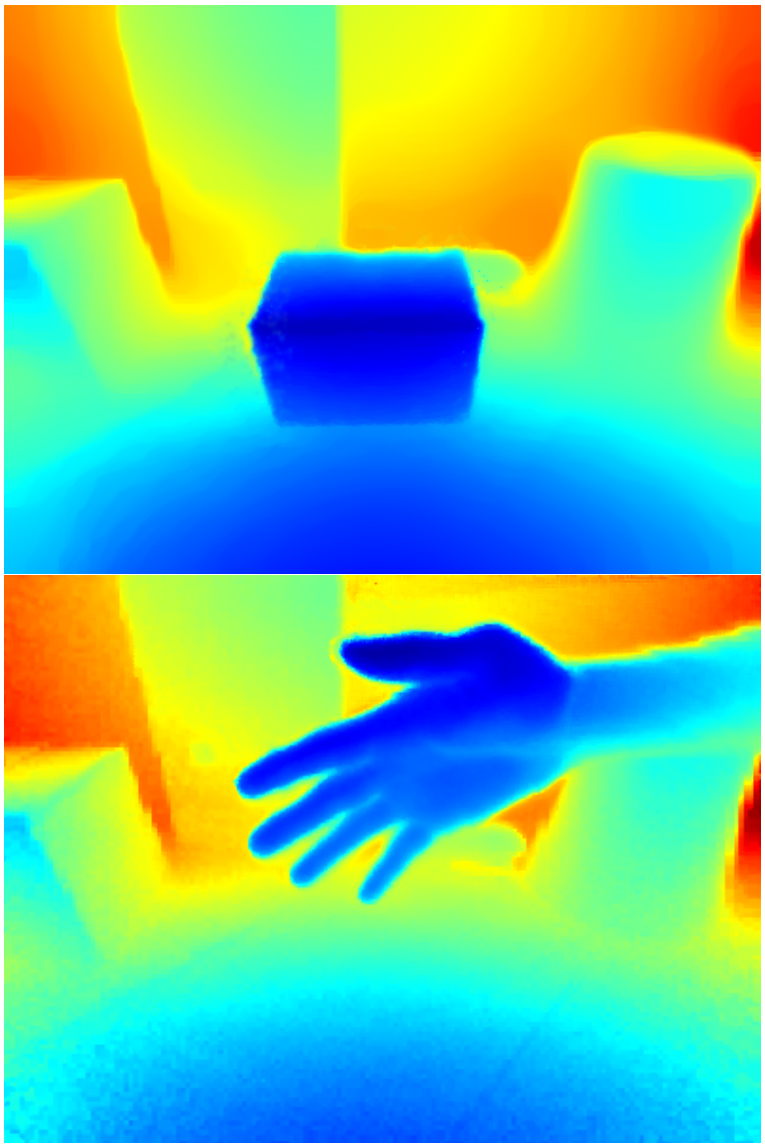
$$\mathbf{X}_1(t) = \mathbf{X}_t - \sum_{i=1}^{t} \mathbf{u}_i, \; \forall t = T...1. \tag{5.77}$$

space, the superresolution result. The superresolved depth image results by back-projecting all point sets $\mathbf{X}_1(t)$ into a higher resolution image space Ω_H and applying our variational superresolution approach [28] without the use of an intensity guidance. If multiple 3D points map in the same image pixel, the median depth of these points is

taken. For this experiment we use $T = 10$ consecutive images from the real world scene shown in Fig. 5.22 (2^{nd} and 3^{rd} row). The resulting depth map has $3\times$ the size of the original input images. The visual results for a rigid and a non-rigid movements of our superresolution approach compared to the first input depth map are shown in Fig. 5.25.



(a) Low resolution input



(b) Superresolved output

Figure 5.25: Temporal depth image superresolution from *SF* on real image sequences. In (a) the object snippet of the first input depth map is shown. In (b) the corresponding superresolution result is shown with a lateral resolution of $3\times$ the input size.

6 Conclusion

6.1 Conclusion

In this thesis we have proposed methods to improve the quality of 3D information from Time of Flight (ToF) depth cameras. These cameras create dense depth maps at very high frame rates. However, due to the active measurement principle and the physical limitations of the sensor chip, the downsides of this technology are a high acquisition noise with gross outliers and a low image resolution. The main focus of this thesis was to improve the depth measurements of *ToF* cameras in terms of image quality. In this context we first have shown different methods to increase the lateral and spatial resolution of single image acquisition. Second, we have presented approaches to enhance the depth measurements by the temporal motion of each scene point in an image sequence.

6.1.1 Depth superresolution

Given a noisy depth acquisition the goal of the superresolution is to estimate a depth image with reduced noise and higher resolution. The estimation is formulated as a global energy optimization problem. To deal with the problems of current depth acquisitions we define a higher order regularization term in the model. This higher order regularization is able reconstruct the depth map out of piecewise planar parts while reducing the noise and preserving sharp discontinuities. For a fast numerical optimization we use a first order primal-dual algorithm.

Further, we show how the superresolution quality can be improved by adding information from other sensor modalities or a learned model into the optimization. First, we show how the superresolution quality can be improved by adding an intensity image of higher resolution into the model. This approach builds on the assumption that textural edges in the intensity image are more likely to appear at high depth discontinuities. Hence, we use this image as a guidance during the optimization. The guidance in this model is formulated as a anisotropic diffusion tensor. Second, we show how we can combine

the superresolution from a single depth image by sparse coding. The idea comes from intensity image superresolution where sparse coding is directly used for superresolving an image. In depth image superresolution this method fails since it is not applicable for higher levels of noise. Therefore, we estimated a high resolution guidance from the low resolution input depth by the proposed variational sparse coding. This approach is more robust to noise since it penalizes not only the number of code samples used but also the difference of neighboring patches. The result of the variational sparse coding is used as an anisotropic guidance in our superresolution approach. With this combination we are able to get more robust against noise than State-of-the-Art (SoA) sparse coding approaches and more accurate than variational approaches.

6.1.2 3D Scene Flow

In our work on 3D motion estimation the main goal was to calculate the scene flow from a sequence of depth and intensity acquisitions. The intensity image in this sequence comes either from an additional intensity camera or by using the infrared image from the *ToF* camera. The main problems we have faced in this estimation were the high acquisition noise and homogeneous parts in the input images as well as illumination changes in the intensity images between acquisitions. In our work, we first proposed the general scene flow framework where we defined the 3D flow through a projection and back-projection using the calibrated camera parameters. To account for the main problems we formulated a model which is more robust to noise and illumination changes. The model is additionally capable of low textured and low structured areas in the input images. The estimation of the scene flow is formulated as a convex energy minimization problem. We have shown different models for the regularization which are capable of smooth flow transitions, which occur at rotations and non-rigid movements, while sharp boundaries of the flow field are preserved. In our evaluation we discussed the properties of the proposed models and gave an qualitative and quantitative comparison to *SoA* approaches. In this context we gave exemplar applications for the scene flow estimation for temporal image superresolution, 3D reconstruction and camera pose estimation.

6.2 Outlook and Future Work

We proposed methods to drastically increase the quality and information from *ToF* depth acquisitions. Despite their success, there is always room for improvement. On the one side we plan to further improve the proposed methods and on the other side we want to apply the current methods to improve other applications in computer vision.

With the recent success of deep learning and Convolutional Neural Networks (CNNs) in computer vision we want to combine their output with our variational models to further increase the quality. We plan to work on novel algorithms which combine a *CNN* together with the proposed superresolution, where the primal-dual scheme is formulated as a neural

network. The whole combination of *CNN* and variational optimization is trained end to end. But the usage of deep learning is not limited to image superresolution. In future work it would be interesting how we can train a scene flow estimation in a similar fashion. In this way, we could train the patchbased comparison operators in our model to get illumination invariant features from a trained *CNN* instead of the existing handcrafted features.

Another direction will be to include the proposed methods in *SoA* applications. The superresolution in combination with the scene flow could be easily used for full 3D reconstructions and structure from motion of non-rigid scenes. The performance of other applications such as gesture recognition, object detection or pose estimation can greatly benefit from the success of our methods.

A

List of Acronyms

AAE	Average Angular Error
AD	Absolute Differences
ANR	Anchored Neighborhood Regression
ARF	Alternating Regression Forest
CNN	Convolutional Neural Network
CP	Closest Point
CT	Census Transform
EPE	End Point Error
GD	Gradient Descend
GR	Global Regression
HR	high resolution
ICP	Iterative Closest Point
KFusion	Kinect Fusion
LED	Light-emitting Diode
LF	Legendre-Fenchel transform
LK	Lukas Kanade
LR	low resolution
MAE	Mean Absolute Error
MRF	Markov Random Field
NCC	Normalized Cross Correlation
NIR	near infrared range
NRMS	Normalized Root Mean Square
OF	Optical Flow
OMP	Orthogonal Matching Pursuit
PMD	Photonic Mixing Device
PSF	Point Spread Function
RF	Random Forest

RMSE	Root Mean Squared Error
SAD	Sum of Absolute Differences
SF	Scene Flow
SfM	Structure from Motion
SISR	single image superresolution
SoA	State-of-the-Art
SR	superresolution
SSD	Sum of Squared Differences
SVD	Singular Value Decomposition
TCT	Ternary Census Transform
TGV	Total Generalized Variation
ToF	Time of Flight
TV	Total Variation

B

List of Publications

My work at the Institute for Computer Graphics and Vision led to the following peer-reviewed publications. For the sake of completeness of this Thesis, they are listed in chronological order along with the respective abstracts.

B.1 2016

A Deep Primal-Dual Network for Guided Depth Super-Resolution

Gernot Riegler, David Ferstl, Matthias Rüther, and Horst Bischof
In: *Proceedings of British Machine Vision Conference (BMVC)*
September 2016, York, Great Britain
(Accepted for oral presentation)

Abstract: In this paper we present a novel method to increase the spatial resolution of depth images. We combine a deep fully convolutional network with a non-local variational method in a deep primal-dual network. The joint network computes a noise-free, high-resolution estimate from a noisy, low-resolution input depth map. Additionally, a high-resolution intensity image is used to guide the reconstruction in the network. By unrolling the optimization steps of a first-order primal-dual algorithm and formulating it as a network, we can train our joint method end-to-end. This not only enables us to learn the weights of the fully convolutional network, but also to optimize all parameters of the variational method and its optimization procedure. The training of such a deep network requires a large dataset for supervision. Therefore, we generate high-quality depth maps and corresponding color images with a physically based renderer. In an exhaustive evaluation we show that our method outperforms the state-of-the-art on multiple benchmarks.

Depth Data Fusion for Simultaneous Localization and Mapping – RGB-DD SLAM

Krzysztof Tadeusz Walas, Michal Nowicki, David Ferstl and Piotr Skrzypczynski
In: *Proceedings of IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*
September 2016, Baden-Baden, Germany
(Accepted for oral presentation)

Abstract: This paper presents an approach to data fusion from multiple depth sensors with different principles of range measurements. This concept is motivated by the observation that depth sensors exploiting different range measurement techniques have also different characteristics of the uncertainty and artifacts in the obtained depth images. Thus, fusing the information from two or more measurement channels allows us to mutually compensate for some of the unwanted effects. The target application of our combined sensor is Simultaneous Localization and Mapping (SLAM). We demonstrated that fusing depth data from two sources in the convex optimization framework yields better results in feature-based 3-D SLAM, than the use of individual sensors for this task. The experimental part is based on data registered with a calibrated rig comprising ASUS Xtion Pro Live and MESA SwissRanger SR-4000 sensors, and ground truth trajectories obtained from a motion capture system. The results of sensor trajectory estimation are demonstrated in terms of the ATE and RPE metrics, widely adopted by the SLAM community.

B.2 2015

Variational Depth Superresolution using Example-Based Edge Representations

David Ferstl, Matthias Rüther, and Horst Bischof
In: *Proceedings of IEEE International Conference on Computer Vision (ICCV)*
December 2015, Santiago de Chile, Chile
(Accepted for poster presentation)

Abstract: In this paper we propose a novel method for depth image superresolution which combines recent advances in example based upsampling with variational superresolution based on a known blur kernel. Most traditional depth superresolution approaches try to use additional high resolution intensity images as guidance for superresolution. In our method we learn a dictionary of edge priors from an external database of high and low resolution examples. In a novel variational sparse coding approach this dictionary is used to infer strong edge priors. Additionally to the traditional sparse coding constraints the difference in the overlap of neighboring edge patches is minimized in our optimization. These edge priors are used in a novel variational superresolution as anisotropic guidance of

a higher order regularization. Both the sparse coding and the variational superresolution of the depth are solved based on the primal-dual formulation. In an exhaustive numerical and visual evaluation we show that our method clearly outperforms existing approaches on multiple real and synthetic datasets.

Learning Depth Calibration of Time-of-Flight Cameras

David Ferstl, Christian Reinbacher, Gernot Riegler, Matthias Rüther, and Horst Bischof
In: *Proceedings of British Machine Vision Conference (BMVC)*
September 2015, Swansea, Great Britain
(Accepted for poster presentation)

Abstract: We present a novel method for an automatic calibration of modern consumer Time-of-Flight cameras. Usually, these sensors come equipped with an integrated color camera. Albeit they deliver acquisitions at high frame rates they usually suffer from incorrect calibration and low accuracy due to multiple error sources. Using information from both cameras together with a simple planar target, we will show how to accurately calibrate both color and depth camera and tackle most error sources inherent to Time-of-Flight technology in a unified calibration framework. Automatic feature detection minimizes user interaction during calibration. We utilize a Random Regression Forest to optimize the manufacturer supplied depth measurements. We show the improvements to commonly used depth calibration methods in a qualitative and quantitative evaluation on multiple scenes acquired by an accurate reference system for the application of dense 3D reconstruction.

A Framework for Articulated Hand Pose Estimation and Evaluation

Gernot Riegler, David Ferstl, Matthias Rüther, and Horst Bischof
In: *Proceedings of Scandinavian Conference on Image Analysis (SCIA)*
june 2015, Copenhagen, Denmark
(Accepted for oral presentation)

Abstract: We present in this paper a framework for articulated hand pose estimation and evaluation. Within this framework we implemented recently published methods for hand segmentation and inference of hand postures. We further propose a new approach for the segmentation and extend existing convolutional network based inference methods. Additionally, we created a new dataset that consists of a synthetically generated training set and accurately annotated test sequences captured with two different consumer depth cameras. The evaluation shows that we can improve with our methods the state-of-the-art. To foster further research, we will make all sources and the complete dataset used in this work publicly available.

B.3 2014

aTGV-SF: Dense Variational Scene Flow through Projective Warping and Higher Order Regularization

David Ferstl, Christian Reinbacher, Gernot Riegler, Matthias Rüther, and Horst Bischof
In: *Proceedings of IEEE International Conference on 3D Vision (3DV)*
December 2014, Tokio, Japan
(**Won best Runner Up Paper Award** - Accepted for oral presentation)

Abstract: In this paper we present a novel method to accurately estimate the dense 3D motion field, known as scene flow, from depth and intensity acquisitions. The method is formulated as a convex energy optimization, where the motion warping of each scene point is estimated through a projection and back-projection directly in 3D space. We utilize higher order regularization which is weighted and directed according to the input data by an anisotropic diffusion tensor. Our formulation enables the calculation of a dense flow field which does not penalize smooth and non-rigid movements while aligning motion boundaries with strong depth boundaries. An efficient parallelization of the numerical algorithm leads to runtimes in the order of 1s and therefore enables the method to be used in a variety of applications. We show that this novel scene flow calculation outperforms existing approaches in terms of speed and accuracy. Furthermore, we demonstrate applications such as camera pose estimation and depth image superresolution, which are enabled by the high accuracy of the proposed method. We show these applications using modern depth sensors such as Microsoft Kinect or the PMD Nano Time-of-Flight sensor.

CP-Census: A Novel Model for Dense Variational Scene Flow from RGB-D Data

David Ferstl, Gernot Riegler, Matthias Rüther, and Horst Bischof
In: *Proceedings of British Machine Vision Conference (BMVC)*
September 2014, Nottingham, Great Britain
(Accepted for oral presentation)

Abstract: We present a novel method for dense variational scene flow estimation based a multiscale Ternary Census Transform in combination with a patchwise Closest Points depth data term. On the one hand, the Ternary Census Transform in the intensity data term is capable of handling illumination changes, low texture and noise. On the other hand, the patchwise Closest Points search in the depth data term increases the robustness in low structured regions. Further, we utilize higher order regularization which is weighted and directed according to the input data by an anisotropic diffusion tensor. This allows to calculate a dense and accurate flow field which supports smooth as well as non-rigid movements while preserving flow boundaries. The numerical algorithm is solved based

on a primal-dual formulation and is efficiently parallelized to run at high frame rates. In an extensive qualitative and quantitative evaluation we show that this novel method for scene flow calculation outperforms existing approaches. The method is applicable to any sensor delivering dense depth and intensity data such as Microsoft Kinect or Intel Gesture Camera.

Hough Networks for Head Pose Estimation and Facial Feature Localization

Gernot Riegler, David Ferstl, Matthias Rüther, and Horst Bischof
In: *Proceedings of British Machine Vision Conference (BMVC)*
September 2014, Nottingham, Great Britain
(Accepted for poster presentation)

Abstract: We present Hough Networks (HNs): a novel method that combines the idea of Hough Forests (HFs) with Convolutional Neural Networks (CNNs). Similar to HFs, we perform a simultaneous classification and regression on densely-extracted image patches. But instead of a Random Forest we utilize a CNN which is capable of learning higher-order feature representations and does not rely on any handcrafted features. Applying a CNN at patch level allows the segmentation of the image into foreground and background. Furthermore, the structure of a CNN supports efficient inference of patches extracted from a regular grid. We evaluate HNs on two computer vision tasks: head pose estimation and facial feature localization. Our method achieves at least state-of-the-art performance without sacrificing versatility which allows extension to many other applications.

MedEyeTrac-a System for optical Monitoring of Eye Positions in tomographic Imaging and Radiotherapy of Eye Tumors

Winkler, P, Ruether, M, Ferstl, D, Huepf, T, Flitsch, R, Konrad, T, Wackernagel, W, Tarmann, L, Langmann, G, Doeller, C, and others
In: *Strahlentherapie und Onkologie*
May 2014

B.4 2013

Image Guided Depth Upsampling using Anisotropic Total Generalized Variation

David Ferstl, Christian Reinbacher, Rene Ranftl, Matthias Rüther, and Horst Bischof
In: *Proceedings of IEEE International Conference on Computer Vision (ICCV)*
December 2013, Sydney, Australia
(Accepted for poster presentation)

Abstract: In this work we present a novel method for the challenging problem of depth image upsampling. Modern depth cameras such as Kinect or Time of Flight cameras deliver dense, high quality depth measurements but are limited in their lateral resolution. To overcome this limitation we formulate a convex optimization problem using higher order regularization for depth image upsampling. In this optimization an anisotropic diffusion tensor, calculated from a high resolution intensity image, is used to guide the upsampling. We derive a numerical algorithm based on a primal-dual formulation that is efficiently parallelized and runs at multiple frames per second. We show that this novel upsampling clearly outperforms state of the art approaches in terms of speed and accuracy on the widely used Middlebury 2007 datasets. Furthermore, we introduce novel datasets with highly accurate groundtruth, which, for the first time, enable to benchmark depth upsampling methods using real sensor data.

Multi-Modality Depth Map Fusion using Primal-Dual Optimization

David Ferstl, Rene Ranftl, Matthias Rüther, and Horst Bischof
In: *Proceedings of IEEE International Conference on Computational Photography (ICCP)*
December 2013, Boston, USA
(Accepted for oral presentation)

Abstract: We present a novel fusion method that combines complementary 3D and 2D imaging techniques. Consider a Time-of-Flight sensor that acquires a dense depth map on a wide depth range but with a comparably small resolution. Complementary, a stereo sensor generates a disparity map in high resolution but with occlusions and outliers. In our method, we fuse depth data, and optionally also intensity data using a primal-dual optimization, with an energy functional that is designed to compensate for missing parts, filter strong outliers and reduce the acquisition noise. The numerical algorithm is efficiently implemented on a GPU to achieve a processing speed of 10 to 15 frames per second. Experiments on synthetic, real and benchmark datasets show that the results are superior compared to each sensor alone and to competing optimization techniques. In a practical example, we are able to fuse a Kinect triangulation sensor and a small size Time-of-Flight camera to create a gaming sensor with superior resolution, acquisition range and accuracy.

B.5 2012

Real-Time Hand Gesture Recognition in a Virtual 3D Environment

David Ferstl, Matthias Rüther, and Horst Bischof
In: *In Proceedings German and Austrian Association for Pattern Recognition (DAGM)*
September 2012, Graz, Austria
(Accepted for oral presentation)

Abstract: We present a novel technique using a range camera for real-time recognition of the hand gesture and position in 3D. Simultaneously the user's hand and head pose are tracked and used for interaction in a virtual 3D desktop environment. As human gestures provide a natural way of communications between humans, gesture recognition is a major field of research used also for human-computer communication. Most existing hand interaction systems are restricted to a 2D touch-sensitive plane, or track and recognize the hand gesture on color images. Due to the lack of depth information, these systems are limited to the 2D space only, provide little depth invariance and are sensitive to rotation and segmentation errors, whereas we built a system for bare-hand 3D interaction, independent from hand rotation and depth.

Depth Coded Shape from Focus

Martin Lenz, David Ferstl, Matthias Rüther, and Horst Bischof
In: *In Proceedings IEEE International Conference on Computational Photography (ICCP)*
April 2012, Seattle, USA
(Accepted for oral presentation)

Abstract: We present a novel shape from focus method for high- speed shape reconstruction in optical microscopy. While the traditional shape from focus approach heavily depends on presence of surface texture, and requires a considerable amount of measurement time, our method is able to perform reconstruction from only two images. Our method relies the rapid projection of a binary pattern sequence, while object is continuously moved through the camera focus range and a single image is continuously exposed. Deconvolution of the integral image allows a direct decoding of binary pattern and its associated depth. Experiments a synthetic dataset and on real scenes show that a depth map can be reconstructed at only 3% of memory costs and fraction of the computational effort compared with traditional shape from focus.

Integral Shape from Focus

David Ferstl, Martin Lenz, Matthias Rüther, and Horst Bischof
In: *In Proceedings Computer Vision Winter Workshop (CVWW)*
February 2012, Mala Nedelja, Slovenia
(Accepted for oral presentation)

Abstract: A disadvantage of traditional Shape From Focus methods is its huge computational cost, which is directly dependent on the size and amount of the acquired images. In this paper we propose a novel method to reduce the measurement time of traditional Shape From Focus to a minimum. This reduction is achieved with a novel focus measure, which calculates the focus by comparison of an estimated all-in-focus image with the stack images using a normalized similarity measure. The all-in-focus image is calculated by a

deconvolution of the integrated image stack with an calibrated integral point spread function. Because of the normalized result, the depth estimation of each surface point stops if a local focus maximum beyond a predefined threshold is found, which leads to a reduction of 45 percent in computation time compared to traditional Shape From Focus. Experiments show, that our method outperforms the traditional Shape Form Focus algorithm in terms of speed with a comparable accuracy.

Bibliography

[1] Adelson, E. H. and Bergen, J. R. (1985). Spatiotemporal energy models for the perception of motion. *Journal of the Optical Society of America A*, 2(2):284–299. (page 70)

[2] Aharon, M., Elad, M., and Bruckstein, A. (2006). K-svd: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Transactions on Signal Processing*, 54(11):4311–4322. (page 34, 46)

[3] Amiaz, T., Lubetzky, E., and Kiryati, N. (2007). Coarse to over-fine optical flow estimation. *Pattern Recognition (PR)*, 40(9):2496 –2503. (page 83)

[4] Baker, S., Scharstein, D., Lewis, J., Roth, S., Black, M., and Szeliski, R. (2011). A database and evaluation methodology for optical flow. *International Journal of Computer Vision (IJCV)*, 92(1):1–31. (page 104)

[5] Basha, T., Moses, Y., and Kiryati, N. (2010). Multi-view scene flow estimation: A view centered variational approach. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (page 70, 73, 105, 110, 116)

[6] Beck, A. and Teboulle, M. (2009). A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, 2(1):183–202. (page 25)

[7] Belhedi, A., Bourgeois, S., Gay-Bellile, V., Sayd, P., Bartoli, A., and Hamrouni, K. (2012). Non-parametric depth calibration of a tof camera. In *Proceedings of the IEEE International Conference on Image Processing*, pages 549 –552. (page 32)

[8] Bevilacqua, M., Roumy, A., Guillemot, C., and line Alberi Morel, M. (2012). Low-complexity single-image super-resolution based on nonnegative neighbor embedding. In *Proceedings of British Machine Vision Conference (BMVC)*. (page 51, 52)

[9] Black, M. J. and Jepson, A. D. (1996). Estimating optical flow in segmented images using variable-order parametric models with local deformations. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 18(10):972–986. (page 83)

[10] Bleyer, M., Rhemann, C., and Rother, C. (2011). Patchmatch stereo - stereo matching with slanted support windows. In *Proceedings of British Machine Vision Conference (BMVC)*. (page 74)

[11] Boyd, S. and Vandenberghe, L. (2004). *Convex Optimization*. Cambridge University Press, New York, NY, USA. (page 16, 24, 90)

[12] Bracewell, R. N. (2000). *The Fourier Transform and Its Applications*. Electrical engineering series. McGraw Hill, New York, NY, USA, 3rd edition. (page 91)

[13] Bredies, K., Kunisch, K., and Pock, T. (2010). Total generalized variation. *SIAM Journal on Imaging Sciences*, 3(3):492– 526. (page 37, 98)

[14] Breiman, L. (2001). Random forests. *Machine Learning (ML)*, 45(1):5–32. (page 32)

[15] Bresson, X., Esedolu, S., Vandergheynst, P., Thiran, J.-P., and Osher, S. (2007). Fast global minimization of the active contour/snake model. *Journal of Mathematical Imaging and Vision*, 28(2):151–167. (page 100)

[16] Brox, T., Bruhn, A., Papenberg, N., and Weickert, J. (2004). High accuracy optical flow estimation based on a theory for warping. In *Proceedings of European Conference on Computer Vision (ECCV)*. (page 72, 83)

[17] Brox, T. and Malik, J. (2011). Large displacement optical flow: Descriptor matching in variational motion estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 33(3):500–513. (page 116)

[18] Burrus, N. (2012). Kinect calibration. `http://nicolas.burrus.name/index.php/Research/KinectCalibration/`. Accessed Nov., 2012. (page 31)

[19] Cech, J., Sanchez-Riera, J., and Horaud, R. (2011). Scene flow estimation by growing correspondence seeds. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (page 73)

[20] Chambolle, A. (2004). An algorithm for total variation minimization and applications. *Journal of Mathematical Imaging and Vision*, 20(1):89 –97. (page 25, 26)

[21] Chambolle, A. and Pock, T. (2011). A first-order primal-dual algorithm for convex problems with applications to imaging. *Journal of Mathematical Imaging and Vision*, 40(1):120 –145. (page 24, 25, 27, 49, 81)

[22] Chan, D., Buisman, H., Theobalt, C., and Thrun, S. (2008). A noise-aware filter for real-time depth upsampling. In *Proceedings of European Conference on Computer Vision Workshops (ECCVW)*. (page 33, 51, 52)

[23] Chang, H., Yeung, D.-Y., and Xiong, Y. (2004). Super-resolution through neighbor embedding. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (page 51, 52)

[24] Cui, Y., Schuon, S., Chan, D., Thrun, S., and Theobalt, C. (2010). 3d shape scanning with a time-of-flight camera. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (page 33)

[25] Diebel, J. and Thrun, S. (2005). An application of markov random fields to range sensing. In *Proceedings of the Conference on Neural Information Processing Systems (NIPS)*. (page 33, 34, 42, 51, 52, 100)

[26] Esser, E., Zhang, X., and Chan, T. (2010). A general framework for a class of first order primal-dual algorithms for convex optimization in imaging science. *SIAM Journal on Imaging Sciences*, 3(4):1015 –1046. (page 25)

[27] Ferstl, D., Ranftl, R., Ruether, M., and Bischof, H. (2013a). Multi-modality depth map fusion using primal-dual optimization. In *Proceedings of the IEEE International Conference on Computational Photography (ICCP)*. (page 33)

[28] Ferstl, D., Reinbacher, C., Ranftl, R., Rüther, M., and Bischof, H. (2013b). Image guided depth upsampling using anisotropic total generalized variation. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. (page 50, 60, 99, 101, 124)

[29] Ferstl, D., Reinbacher, C., Riegler, G., Ruether, M., and Bischof, H. (2014a). atgv-sf: Dense variational scene flow through projective warping and higher order regularization. In *Proceedings of the IEEE International Conference on 3D Vision (3DV)*. (page 99)

[30] Ferstl, D., Reinbacher, C., Riegler, G., Ruether, M., and Bischof, H. (2015). Learning depth calibration of time-of-flight cameras. In *Proceedings of British Machine Vision Conference (BMVC)*. (page 32, 41, 60, 119)

[31] Ferstl, D., Riegler, G., Ruether, M., and Bischof, H. (2014b). Cp-census: A novel model for dense variational scene flow from rgb-d data. In *Proceedings of British Machine Vision Conference (BMVC)*. (page 90)

[32] Freeman, B. and Liu, C. (2011). *Markov random fields for super-resolution and texture synthesis*, chapter 10. MIT Press, Cambridge, MA, USA. (page 34)

[33] Froba, B. and Ernst, A. (2004). Face detection with the modified census transform. In *Proceedings of the IEEE International Conference on Automatic Face and Gesture Recognition (FG)*, pages 91–96. (page 92)

[34] Fuchs, S. (2010). Multipath interference compensation in time-of-flight camera images. In *Proceedings of the IEEE International Conference on Pattern Recognition (ICPR)*. (page 30)

[35] Fuchs, S. and Hirzinger, G. (2008). Extrinsic and depth calibration of tof-cameras. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (page 32)

[36] Gottfried, J.-M., Fehr, J., and Garbe, C. (2011). Computing range flow from multi-modal kinect data. In *Proceedigns of the International Symposium on Visual Computing*. (page 73, 74)

[37] Gudmundsson, S. A., Aanaes, H., and Larsen, R. (2008). Fusion of stereo vision and time-of-flight imaging for improved 3d estimation. *International Journal of Intelligent Systems Technologies and Applications*, 5(3/4):425 –433. (page 33)

[38] Guomundsson, S., Aanaes, H., and Larsen, R. (2007). Environmental effects on measurement uncertainties of time-of-flight cameras. In *Proceedings of the IEEE International Symposium on Signals Circuits and Systems (ISSCS)*. (page 30, 32)

[39] Hadfield, S. and Bowden, R. (2011). Kinecting the dots: Particle based scene flow from depth sensors. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. (page 73, 74, 105, 110)

[40] Hadfield, S. and Bowden, R. (2014). Scene particles: Unregularized particle-based scene flow estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 36(3):564–576. (page 73, 74, 116)

[41] Hamming, R. W. (1950). Error detecting and error correcting codes. *Bell System Technical Journal*, 29(2):147 –160. (page 92)

[42] Hartley, R. and Zisserman, A. (2003). *Multiple View Geometry*. Cambridge University Press, Cambridge, UK, 2nd edition. (page 12)

[43] He, K., Sun, J., and Tang, X. (2010). Guided image filtering. In *Proceedings of European Conference on Computer Vision (ECCV)*. (page xxiv, 51, 52, 60, 61)

[44] Herbst, E., Ren, X., and Fox, D. (2013). Rgb-d flow: Dense 3-d motion estimation using color and depth. In *Proceedings of the International Conference for Robotics and Automation (ICRA)*. (page 74)

[45] Herrera C., D., Kannala, J., and Heikkilä, J. (2012). Joint depth and color camera calibration with distortion correction. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 34(10):2058 –2064. (page 31)

[46] Hess, R., Baker, C., and Zihl, J. (1989). The "motion-blind" patient: low-level spatial and temporal filters. *The Journal of Neuroscience*, 9(5):1628–1640. (page 70)

[47] Hirschmuller, H. and Scharstein, D. (2007). Evaluation of cost functions for stereo matching. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (page 50)

[48] Horn, B. K. and Schunck, B. G. (1981). Determining optical flow. *Artificial Intelligence*, 17(1-3):185 –203. (page 70, 72, 75, 79, 80)

[49] Hornáček, M., Fitzgibbon, A., and Rother, C. (2014). Sphereflow: 6dof scene flow from rgb-d pairs. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (page 74, 105, 108, 109, 110, 116, 117, 118, 119, 120, 121, 122)

[50] Hornáček, M., Rhemann, C., Gelautz, M., and Rother, C. (2013). Depth super resolution by rigid body self-similarity in 3d. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (page 34, 51, 52, 54, 55, 74)

[51] Huber, P. J. (1973). Robust regression: Asymptotics, conjectures and monte carlo. *The Annals of Statistics*, 1(5):799 –821. (page 9)

[52] Huguet, F. and Devernay, F. (2007). A variational method for scene flow estimation from stereo sequences. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. (page 70, 72, 110, 116)

[53] Kahlmann, T., Remondino, F., and Ingensand, H. (2006). Calibration for increased accuracy of the range imaging camera swissranger. *Proceedings of the International Symposium of the Photogrammetry, Remote Sensing and Spatial Information Sciences (ISPRS)*, 36(5):136 –141. (page 32)

[54] Kearney, J. K. and Thompson, W. B. (1988). Bounding constraint propagation for optical flow estimation. In *Motion Understanding: Robot and Human Vision*, pages 1 –21. Springer US, Boston, MA, USA. (page 70)

[55] Kopf, J., Cohen, M. F., Lischinski, D., and Uyttendaele, M. (2007). Joint bilateral upsampling. *ACM Transactions on Graphics (TOG)*, 26(3):961 –965. (page xxiv, 33, 51, 52, 60, 61)

[56] Lange, R. (2000). *3D Time-of-Flight distance measurement with custom solid-state image sensors in CMOS/CCD technology*. PhD thesis, Department of Electrical Engineering and Computer Science at University of Siegen, Germany. (page 30)

[57] Letouzey, A., Petit, B., and Boyer, E. (2011). Scene flow from depth and color images. In *Proceedings of British Machine Vision Conference (BMVC)*. (page 73, 74, 84)

[58] Leung, S. and Osher, S. (2005). Global minimization of the active contour model with tv-inpainting and two-phase denoising. In *Proceedings on Variational, Geometric, and Level Set Methods in Computer Vision (VLSM)*, volume 3752 of *Lecture Notes in Computer Science*, pages 67 –79. Springer US. (page 100)

[59] Lewis, J. P. (1995). Fast normalized cross-correlation. In *Proceedings of Vision Interface*, pages 120 –123. (page 91)

[60] Li, J., Zeng, G., Gan, R., Zha, H., and Wang, L. (2012). A bayesian approach to uncertainty-based depth map super resolution. In *Proceedigns of the Asian Conference on Computer Vision*. (page 34)

[61] Lindner, M. and Kolb, A. (2006). Lateral and depth calibration of pmd-distance sensors. In *Proceedigns of the International Symposium on Visual Computing*, pages 524 –533. Springer-Verlag Berlin Heidelberg. (page 32)

[62] Lindner, M. and Kolb, A. (2007). Calibration of the intensity-related distance error of the pmd tof-camera. *Intelligent Robots and Computer Vision: Algorithms, Techniques, and Active Vision*, 6764(1):1 –8. (page 30, 32)

[63] Lions, P. L. and Mercier, B. (1979). Splitting algorithms for the sum of two nonlinear operators. *SIAM Journal on Numerical Analysis*, 16(6):964 –979. (page 25)

[64] Lucas, B. D. and Kanade, T. (1981). An iterative image registration technique with an application to stereo vision. In *International Joint Conference on Artificial Intelligence*. (page 72, 75, 79)

[65] Lukins, T. C. and Fisher, B. (2005). Colour constrained 4d flow. In *Proceedings of British Machine Vision Conference (BMVC)*. (page 73)

[66] Mac Aodha, O., Campbell, N. D., Nair, A., and Brostow, G. J. (2012). Patch based synthesis for single depth image super-resolution. In *Proceedings of European Conference on Computer Vision (ECCV)*. (page xvii, 4, 34, 47, 50, 51, 52, 53, 54, 55)

[67] Mandal, S., Bhavsar, A., and Sao, A. (2014). Hierarchical example-based range-image super-resolution with edge-preservation. In *Proceedings of the IEEE International Conference on Image Processing*. (page 34)

[68] May, S., Droeschel, D., Fuchs, S., Holz, D., and Nuchter, A. (2009). Robust 3d-mapping with time-of-flight cameras. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*. (page 30)

[69] Michaeli, T. and Irani, M. (2013). Nonparametric blind super-resolution. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. (page 35, 36)

[70] Mitzel, D., Pock, T., Schoenemann, T., and Cremers, D. (2009). Video super resolution using duality based tv-l1 optical flow. In *Proceedings of the German Conference on Pattern Recognition (GCPR*. (page 34)

[71] Nagel, H.-H. (1989). On a constraint equation for the estimation of displacement rates in image sequences. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 11(1):13–30. (page 85, 101)

[72] Nagel, H.-H. and Enkelmann, W. (1986). An investigation of smoothness constraints for the estimation of displacement vector fields from image sequences. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 8(5):565 –593. (page 42, 48)

[73] Nasrollahi, K. and Moeslund, T. (2014). Super-resolution: a comprehensive survey. *Machine Vision and Applications*, 25(6):1423 –1468. (page 34)

[74] Nesterov, Y. (2004). *Introductory lectures on convex optimization: a basic course*, volume 87 of *Applied Optimization*. Springer US, Boston, MA, USA. (page 16, 24, 25)

[75] Newcombe, R. A., Fox, D., and Seitz, S. M. (2015). Dynamicfusion: Reconstruction and tracking of non-rigid scenes in real-time. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (page 33)

[76] Newcombe, R. A., Izadi, S., Hilliges, O., Molyneaux, D., Kim, D., Davison, A. J., Kohli, P., Shotton, J., Hodges, S., and Fitzgibbon, A. (2011). Kinectfusion: Real-time dense surface mapping and tracking. In *Proceedings of the International Symposium on Mixed and Augmented Reality (ISMAR)*. (page 33, 70, 123)

[77] Nikolova, M. (2004). A variational approach to remove outliers and impulse noise. *Journal of Mathematical Imaging and Vision*, 20(1-2):99 –120. (page 37)

[78] Ochs, P., Dosovitskiy, A., Brox, T., and Pock, T. (2013). An iterated ℓ_1 algorithm for non-smooth non-convex optimization in computer vision. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (page 80)

[79] Olshausen, B. A. and Field, D. J. (1997). Sparse coding with an overcomplete basis set: A strategy employed by v1? *Vision Research*, 37(23):3311 –3325. (page 34)

[80] Papenberg, N., Bruhn, A., Brox, T., Didas, S., and Weickert, J. (2006). Highly accurate optic flow computation with theoretically justified warping. *International Journal of Computer Vision (IJCV)*, 67(2):141 –158. (page 88)

[81] Parikh, N. and Boyd, S. (2014). Proximal algorithms. *Foundations and Trends in Optimization*, 1(3):127 –239. (page xviii, 23)

[82] Park, J., Kim, H., Tai, Y.-W., Brown, M., and Kweon, I. (2011). High quality depth map upsampling for 3d-tof cameras. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. (page 34, 42, 50, 51, 52, 53, 54)

[83] PMD Technologies, Siegen, G. (2004). Camboard nano tof camera. (page 60, 103)

[84] Pock, T. and Chambolle, A. (2011). Diagonal preconditioning for first order primal-dual algorithms in convex optimization. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. (page 27)

[85] Pollefeys, M., Koch, R., and Gool, L. V. (1999). Self-calibration and metric reconstruction in spite of varying and unknown internal camera parameters. In *International Journal of Computer Vision (IJCV)*, pages 7–25. (page 70)

[86] Polyak, B. (1964). Some methods of speeding up the convergence of iteration methods. *USSR Computational Mathematics and Mathematical Physics*, 4(5):1–17. (page 24)

[87] Quiroga, J., Brox, T., Devernay, F., and Crowley, J. (2014). Dense semi-rigid scene flow estimation from rgbd images. In *Proceedings of European Conference on Computer Vision (ECCV)*. (page 74, 110, 116, 117, 118, 119)

[88] Quiroga, J., Devernay, F., and Crowley, J. L. (2012). Scene flow by tracking in intensity and depth data. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. (page xxiv, 73, 105, 110, 116)

[89] Quiroga, J., Devernay, F., and Crowley, J. L. (2013a). Local scene flow by tracking in intensity and depth. *Journal of Visual Communication Image Representation*, 25(1):98 –107. (page 117, 118, 119)

[90] Quiroga, J., Devernay, F., and Crowley, J. L. (2013b). Local/global scene flow estimation. In *Proceedings of the IEEE International Conference on Image Processing*. (page 73, 74, 105, 110, 116)

[91] Radmer, J., Fuste, P., Schmidt, H., and Kruger, J. (2008). Incident light related distance error study and calibration of the pmd-range imaging camera. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1 –6. (page 32)

[92] Ranftl, R., Gehrig, S., Pock, T., and Bischof, H. (2012). Pushing the limits of stereo using variational stereo estimation. In *IEEE Intelligent Vehicles Symposium*. (page 92, 101)

[93] Ranftl, R., Pock, T., and Bischof, H. (2013). Minimizing tgv-based variational models with non-convex data terms. In *Proceedings of the International Conference on Scale Space and Variational Methods in Computer Vision (SSVM)*. (page 80)

[94] Reinbacher, C., Pock, T., Bauer, C., and Bischof, H. (2010). Variational segmentation of elongated volumetric structures. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (page xx, 101)

[95] Reynolds, M., Dobos, J., Peel, L., Weyrich, T., and Brostow, G. (2011). Capturing time-of-flight data with confidence. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (page 32)

[96] Rockafellar, R. T. (1976). Monotone operators and the proximal point algorithm. *SIAM Journal on Control and Optimization*, 14(5):877 –898. (page 22)

[97] Rockafellar, R. T. (1997). *Convex Analysis*. Princeton University Press, Princeton, MA, USA. (page 16, 24, 79)

[98] Rockafellar, R. T. and Wets, R. J.-B. (1998). *Variational Analysis*, volume 317. Springer-Verlag Berlin Heidelberg, Berlin Heidelberg, Germany. (page 16, 24)

[99] Rudin, L. I., Osher, S., and Fatemi, E. (1992). Nonlinear total variation based noise removal algorithms. *Physica D: Nonlinear Phenomena*, 60(1-4):259 –268. (page 25, 97)

[100] Scharstein, D. and Pal, C. (2007). Learning conditional random fields for stereo. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (page 50)

[101] Scharstein, D. and Szeliski, R. (2002). A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International Journal of Computer Vision*, 47(1-3):7 –42. (page)

[102] Scharstein, D. and Szeliski, R. (2003). High-accuracy stereo depth maps using structured light. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (page 50, 110)

[103] Schmidt, M. (2011). *Analysis, Modeling and Dynamic Optimization of 3D Time-of-Flight Imaging Systems*. PhD thesis, Ruperto-Carola University of Heidelberg, Germany. (page 30)

[104] Schulter, S., Leistner, C., Wohlhart, P., Roth, P., and Bischof, H. (2013). Alternating regression forests for object detection and pose estimation. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. (page 32)

[105] Schuon, S., Theobalt, C., Davis, J., and Thrun, S. (2008). High-quality scanning using time-of-flight depth superresolution. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. (page 32)

[106] Schuon, S., Theobalt, C., Davis, J., and Thrun, S. (2009). Lidarboost: Depth superresolution for tof 3d shape scanning. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (page 32)

[107] Shotton, J., Fitzgibbon, A., Cook, M., Sharp, T., Finocchio, M., Moore, R., Kipman, A., and Blake, A. (2011). Real-time human pose recognition in parts from a single depth image. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (page 70)

[108] Smisek, J., Jancosek, M., and Pajdla, T. (2011). 3d with kinect. In *Proceedings of the IEEE International Conference on Computer Vision Workshops (ICCVW)*. (page 31)

[109] Song, S. and Xiao, J. (2014). Sliding shapes for 3d object detection in depth images. In *Proceedings of European Conference on Computer Vision (ECCV)*. (page 70)

[110] Spies, H., Jahne, B., and Barron, J. (2000). Dense range flow from depth and intensity data. In *Proceedings of the IEEE International Conference on Pattern Recognition (ICPR)*. (page 73)

[111] Stein, F. (2004). Efficient Computation of Optical Flow Using the Census Transform. In *Proceedings of the German Conference on Pattern Recognition (GCPR*. (page 93)

[112] Sun, D., Roth, S., and Black, M. J. (2014). A quantitative analysis of current practices in optical flow estimation and the principles behind them. *International Journal of Computer Vision (IJCV)*, 106(2):115 –137. (page 72, 104, 105)

[113] Sun, D., Sudderth, E. B., and Pfister, H. (2015). Layered rgbd scene flow estimation. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 548 –556. (page 74)

[114] Timofte, R., De Smet, V., and Van Gool, L. (2013). Anchored neighborhood regression for fast example-based super-resolution. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. (page xviii, 34, 51, 52, 53, 54)

[115] Tomasi, C. and Manduchi, R. (1998). Bilateral filtering for gray and color images. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. (page 33)

[116] Tropp, J. and Gilbert, A. (2007). Signal recovery from random measurements via orthogonal matching pursuit. *IEEE Transactions on Information Theory*, 53(12):4655 –4666. (page 34, 46)

[117] Unger, M., Pock, T., Werlberger, M., and Bischof, H. (2010). A convex approach for variational super-resolution. In *Proceedings of the German Conference on Pattern Recognition (GCPR*. (page 35, 36, 53)

[118] Valgaerts, L., Bruhn, A., Zimmer, H., Weickert, J., Stoll, C., and Theobalt, C. (2010). Joint estimation of motion, structure and geometry from stereo sequences. In *Proceedings of European Conference on Computer Vision (ECCV)*. (page 72)

[119] Vedula, S., Baker, S., Rander, P., Collins, R., and Kanade, T. (1999). Three-dimensional scene flow. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. (page 70, 72, 80)

[120] Vedula, S., Rander, P., Collins, R., and Kanade, T. (2005). Three-dimensional scene flow. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 27(3):475–480. (page 70, 72)

[121] Verri, A. and Poggio, T. (1989). Motion field and optical flow: qualitative properties. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 11(5):490–498. (page 70, 85)

[122] Vogel, C., Roth, S., and Schindler, K. (2014). View-consistent 3d scene flow estimation over multiple frames. In *Proceedings of European Conference on Computer Vision (ECCV)*. (page 73)

[123] Vogel, C., Schindler, K., and Roth, S. (2011). 3d scene flow estimation with a rigid motion prior. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. (page 73)

[124] Vogel, C., Schindler, K., and Roth, S. (2013). Piecewise rigid scene flow. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. (page 70, 73, 74)

[125] Warren, W., Kay, B., Zosh, W., Duchon, A., and Sahuc, S. (2001). Optic flow is used to control human walking. *Nature: Neuroscience*, 4(2):213–216. (page 70)

[126] Wedel, A., Brox, T., Vaudrey, T., Rabe, C., Franke, U., and Cremers, D. (2011). Stereoscopic scene flow computation for 3d motion understanding. *International Journal of Computer Vision (IJCV)*, 95(1):29 –51. (page 70, 72)

[127] Wedel, A., Pock, T., Zach, C., Bischof, H., and Cremers, D. (2009). An improved algorithm for tv-l1 optical flow. In *Proceedings of the International Dragstuhl Seminar on Statistical and Geometrical Approaches to Visual Motion Analysis*, pages 23 –45. (page 88)

[128] Wedel, A., Rabe, C., Vaudrey, T., Brox, T., Franke, U., and Cremers, D. (2008). Efficient dense scene flow from sparse or dense stereo data. In *Proceedings of European Conference on Computer Vision (ECCV)*. (page 72)

[129] Werlberger, M. (2012). *Convex Approaches for High Performance Video Processing*. PhD thesis, Institute for Computer Graphics and Vision, Graz University of Technology, Graz, Austria. (page 75, 85, 90, 104)

[130] Werlberger, M., Pock, T., and Bischof, H. (2010). Motion estimation with non-local total variation regularization. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (page 90, 105)

[131] Xu, L., Jia, J., and Matsushita, Y. (2012). Motion detail preserving optical flow estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 34(9):1744–1757. (page 116)

[132] Yang, J., Wright, J., Huang, T., and Ma, Y. (2010). Image super-resolution via sparse representation. *IEEE Transactions on Image Processing*, 19(11):2861–2873. (page 34, 47, 51, 52)

[133] Yang, Q., Yang, R., Davis, J., and Nister, D. (2007). Spatial-depth super resolution for range images. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (page 33, 51, 52)

[134] Yuan, Q., Zhang, L., and Shen, H. (2012). Multiframe super-resolution employing a spatially weighted total variation model. *IEEE Transactions on Circuits and Systems for Video Technology*, 22(3):379–392. (page 35)

[135] Zabih, R. and Woodfill, J. (1994). Non-parametric local transforms for computing visual correspondence. In *Proceedings of European Conference on Computer Vision (ECCV)*. (page 92)

[136] Zach, C., Pock, T., and Bischof, H. (2007). A duality based approach for realtime tv-l1 optical flow. In *Proceedings of the German Conference on Pattern Recognition (GCPR*. (page 81, 83, 97)

[137] Zeyde, R., Elad, M., and Protter, M. (2012). On single image scale-up using sparse-representations. In *International Conference on Curves and Surfaces*, volume 6920, pages 711 –730. (page xxiv, 34, 45, 46, 47, 51, 52, 55, 59)

[138] Zhang, X., Chen, D., Yuan, Z., and Zheng, N. (2013). Dense scene flow based on depth and multi-channel bilateral filter. In *Proceedigns of the Asian Conference on Computer Vision*. (page 73, 74, 110, 116)

[139] Zhang, Z. (1994). Iterative point matching for registration of free-form curves and surfaces. *International Journal of Computer Vision (IJCV)*, 13(2):119–152. (page 94)

[140] Zhu, J., Wang, L., Yang, R., Davis, J., and Pan, Z. (2011). Reliability fusion of time-of-flight depth and stereo geometry for high quality depth maps. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 33(7):1400 –1414. (page 33)