Convex Approaches for High Performance Video Processing

Dissertation

Manuel Werlberger

Graz, June 2012

Draft Copy: June 1, 2012

Institute for Computer Graphics and Vision
Graz University of Technology

TO MY FAMILY.

If we knew what it was we were doing, it would not be called research, would it?

Albert Einstein

ABSTRACT

Accurate and robust motion estimation in image sequences is essential for high quality video processing and digital film restoration. The ability to deal with strong outliers and large impaired regions is especially important for restoring historical film. Typical artifacts like brightness changes, noise, scratches or other forms of missing data may cause the algorithms to fail. Even in situations without disturbance and only changing illuminations the algorithms often have problems to compute the correct motion.

We provide a convex variational framework for motion estimation with different extensions to increase its accuracy as well as its robustness. We utilize efficient primal-dual methods for optimization that exhibit good parallelization capabilities. The algorithms are well suited for an efficient implementation on modern graphics hardware. Moreover, we evaluate all the different models on a benchmark sequence where ground-truth motion is known to quantitatively compare the proposed approaches.

To increase both, the robustness and the accuracy of the motion estimator we impose the following modifications into the motion estimation framework: At first, we extend existing approaches from the literature by anisotropic variants of the robust Huber norm to increase the accuracy near motion boundaries. Next, based on the Gestalt principles of grouping, we incorporate a low-level segmentation process to guide the regularization. We propose to use a non-local Huber regularization to impose this grouping into the variational framework. Constant intensity values over time along motion trajectories are a commonly made assumption to simplify motion estimation. For realistic scenes this requirement is rarely fulfilled. Hence, the ability to handle brightness variations is important and we show different approaches that are able to deal with such data. Furthermore, we propose a method to incorporate arbitrary data terms into the energy functional. This enables the usage of any desired data similarity measure.

In order to demonstrate the capabilities of robust motion estimation we incorporate the proposed motion estimation approach into video processing applications. At first, flow guidance is introduced into a variational TV-ℓ_1 denoising framework. Besides denoising this approach enables the restoration of coarse outliers. Even large areas of missing data can be filled with reasonable information. Most importantly, the given motion is continued within the unknown areas. In fact, the method is capable of recreating completely missing frames. Moreover, multiple frames can be generated which enables the approach to generate artificial slow-motion sequences. Second, we propose to use optical flow guidance also in a patch-based inpainting approach. This approach is mainly intended to remove static

artifacts that occur over a longer period of time. In addition, the other objects' motion is not affected by the restoration process. Including optical flow guidance into this patch-based approach demonstrates that the proposed enhancements are not limited to variational approaches.

Keywords. film, video processing, restoration, variational methods, motion, optical flow

CONTENTS

CHAPTER
ONE

MOTIVATION

The *UCLA Film and Television Archive* holds over 300000 films and television programs and over 8 million meters of newsreel (twice the distance Earth – Moon). The *Academy Film Archive* is in possession of over 140000 motion picture films and videos. Both archives emphasize the value of their collections and especially highlight the importance of preserving historical footage. The aim is to transfer all the material that resides on unstable film stock towards modern polyester film or to digital formats. In addition, the storage condition is optimized for minimizing the deterioration of the film bases. The amount of material and the difficulty of the restoration task justifies the need for automatic video processing algorithms.

With the rise of digitalization, the application of automatic algorithms for restoration tasks gains in importance. Modern algorithms are not only beneficial for restoration but also for video editing and motion picture post-production. More and more, digital technology is entering the domain of motion picture film production. While the digital format has quickly been accepted for broadcasting, it took longer for the digitalization process to enter the movie business. There, first the post-production stage has been concerned with the introduction of the digital intermediate. With the recent shift towards digital cinematography, digital media also outranked film as the medium for distribution companies. Subsequently, the current improvements in recording equipment completes the digitalization process. Still, the brief outline (for more information see Chapter 2) of this development shows, that digital video processing is a very relevant topic. Speaking in terms of money, faster and more accurate algorithms simplify the digital processing stages and may lower the production costs. In addition, the fast development of new technologies demands for high quality material and approaches are needed to process (older) material to meet the requirements.

For the restoration task, the major advantage of automation is obvious. The robust detection and restoration of film defects remains a very challenging task and is far from being solved for arbitrary impairments. In this thesis we do not push for a complete restoration pipeline but emphasize important elements that can be incorporated into such a pipeline to improve its usability, quality and robustness. For a description of a complete restoration pipeline, we refer to the approaches of Van Roosmalen [1999], and Kokaram [2003]. The succession of the pipeline's building blocks as in Figure 1.1 got well established [Van Roosmalen, 1999, Höller, 2012]. Notably, the position of "Image Stabilization" is different in the work of Van Roosmalen [1999] and Höller [2012]. In our opinion, stabilizing the video sequence at first simplifies the other processing steps. Furthermore, most of the

existing approaches do not utilize motion vectors other than for the retouching. The main reason is the lack of robust motion estimators and *e.g.* the potential occurrence of flicker artifacts makes it impossible for most approaches to compute reasonable motion vectors. Motion vectors are a rich source of information and are crucial for the restoration quality. Given a very robust motion estimator, the motion vectors could be beneficial in all stages of the restoration pipeline. The dashed arrows in Figure 1.1 highlight the desire to use motion information in every stage. Following, we give some examples how the single stages can benefit from an accurate and robust motion estimation:

- For the `stabilization` stage, the motion vectors can be used to compute the high frequency parts of the global motion or even stabilize unsteady camera work.
- If the approach is able to compute stable motion vectors in case of `flicker`, the illumination differences can be computed directly. Hence, the brightness variations are known and can be used for correcting them.
- The need for accurate motion vectors in the `retouching` is clear and if the motion vectors are perfectly reconstructed, this restoration stage is almost trivial.
- In the case of `noise` reduction the knowledge of the points' motion can be helpful as well. Mostly, removing noise is formulated as some smoothing process. When incorporating motion information the smoothing can also be performed temporally which can enhance the quality of single frames.

> *Motion estimation is the ideal vehicle for investigating the nature of image sequences ...* *[Kokaram, 2003]*

Actually, this statement from Kokaram [2003] is quite obvious. But in other words one could also say that if the motion within a sequence is perfectly understood and estimated, editing this sequence gets much easier. Unfortunately, there is no such algorithm that is able to compute the motion in every imaginable situation since motion estimation is an ill-posed problem. Consider the restoration task where parts of the image are destructed. There, the estimation of the pixels' movements between the single frames is anything but simple. Even slight changes in the illumination cause a lot of existing motion estimation algorithms to fail. Also the schematic overview of a restoration pipeline shows the need for a robust motion estimation. A robust motion estimation is crucial for all other steps in the restoration pipeline. Hence, motion estimation plays an essential role in this thesis. The emphasis lies on increasing its robustness towards outliers that occur during film restoration. In addition, we will present approaches that enhance the accuracy of such algorithms. Let us point out once more that this thesis will not introduce improvements of the restoration pipeline itself, but the presented algorithmic considerations improve the quality and robustness of the single steps, first and foremost, the motion estimation algorithm.

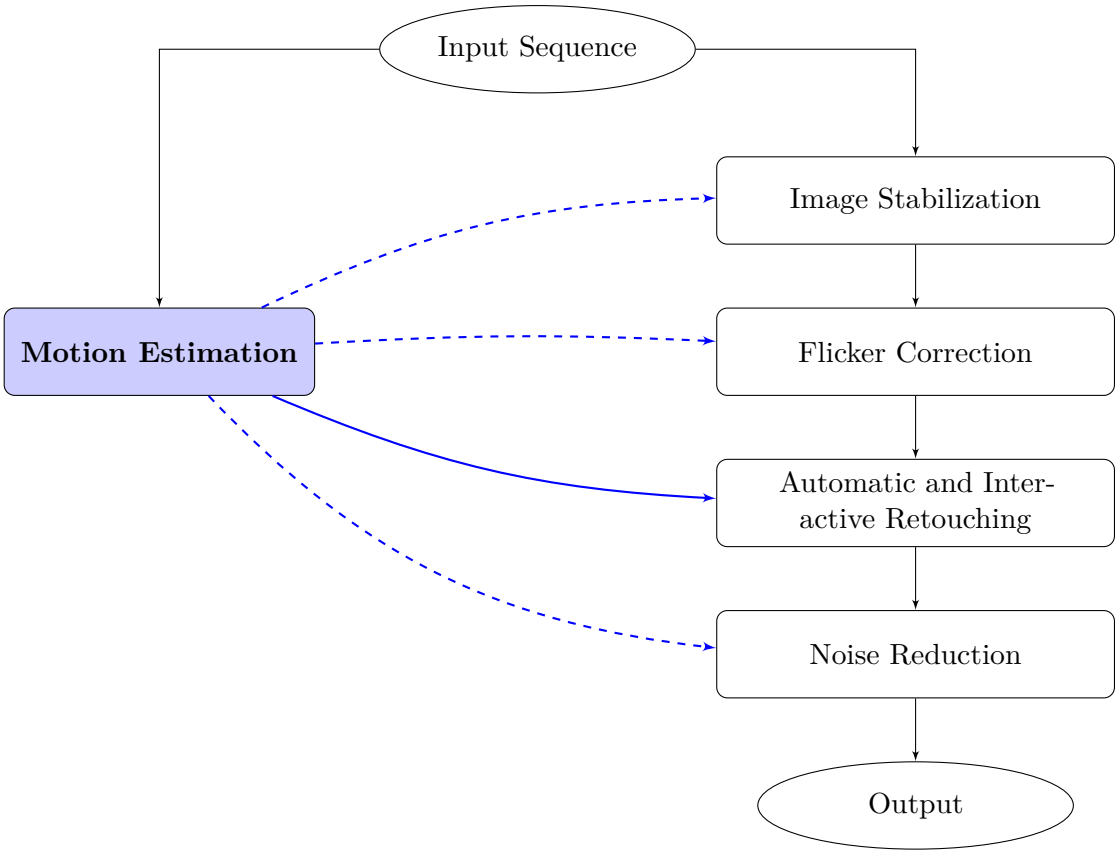


Figure 1.1.: Restoration pipeline.

Not only restoration tasks but also digital video processing and editing tasks attach great importance to the accuracy of the estimated motion. A common example is video stabilization that *e.g.* reduces the unsteadiness of the camera work by computing the motion of the camera and registering the moving frames onto each other. A second example which is of common interest is the generation of intermediate frames. Be it for restoration purposes, when dealing with unrecoverable frames, or for generating multiple intermediate frames for artificial slow motion sequences. To continue the occurring motion in the intermediate frames to get a natural viewing impression, again the estimation of the pixels' movement is crucial. Besides that, motion information improves a variety of algorithms where at a first glance the use of motion information is not obvious.

1.1. Organization and Contribution

The thesis comprises different parts of the publications [Werlberger et al., 2009, 2010, 2011a,b]. [Werlberger et al., 2011a] features the application of robust motion estimation for video processing tasks, whereas the other publications relate to methodical and algorithmic

improvements for motion estimation. When illustrating the organization of the rest of this thesis we mention parts where the contributions are incorporated. Appendix **??** lists all publications with a brief description of their content.

The robustness and quality of the algorithms is particularly crucial when dealing with corrupted input data. This is especially the case for film and video restoration tasks. Chapter 2 touches on this comprehensive topic. At first, we discuss the development of film and video for recording motion pictures and broadcasting. Furthermore, we introduce the taxonomy of video impairments and give a brief outline of relevant restoration approaches.

Chapter 3 provides an overview on the mathematical framework utilized throughout the thesis. Besides the mathematical definitions, an introduction on variational methods, convexity and related optimization algorithms is included in this chapter.

The schematic presentation of a typical restoration pipeline in Figure 1.1 reveals the importance of motion estimation. Chapter 4 introduces the concept of optical flow and its connection to motion estimation. In addition, this chapter comprises the origin for all our investigations in the field of motion estimation. The variational formulation to compute optical flow will be retained throughout the thesis. We mainly rely on the formulations of Weickert and Schnörr [2001], Brox et al. [2004], Zach et al. [2007]. In addition, we introduce the numerical scheme to optimize problems of the defined form. This will be used for the optimization of all the presented approaches for optical flow estimation.

Chapter 5 introduces several extensions of the optical flow approach to improve its robustness and quality. This expands the field of application for such algorithms. Especially for difficult data (*e.g.* artifacts afflicted video sequences) the computation of motion vectors is challenging. But for such data, an accurate and reliable motion estimate is crucial to improve the restoration quality. Besides the algorithmic improvements, an additional contribution is the definition of a versatile framework. All the extensions are incorporated into the same framework. This improves the ability to combine different enhancements and the general framework allows for a better comparison of the presented approaches. In the paper [Werlberger et al., 2009] we defined a variational optical flow model that we will use as the basis throughout the thesis. All extensions and investigations for motion estimation are incorporated into this framework. There, we also used a image-driven regularization that we will discuss in some variations in Chapter 5. Furthermore, for optimization we exchange the splitting approach by a more sophisticated primal-dual algorithm. The contributions of the paper [Werlberger et al., 2010] is then divided into improvements of the regularization and data term to better integrate into the thesis. The regularization is extended to incorporate larger regions in order to enforce the regularization within a certain object but prohibit the smoothing process across motion boundaries. The second contribution of this paper is the ability to utilize any arbitrary data term. This enables to also use more complex and also patch-based similarity measures which are difficult to include into the standard formulation. To compare the effects of the various extensions, we did an extensive evaluation of the various approaches on a benchmark dataset. Supplemental plots are depicted in Appendix A.

Chapter 6 deals with video processing applications and shows how motion vectors can be utilized to improve the quality in video post-production. The chapter is mainly divided into two parts:

- First, we define a variational restoration approach in form of an extended denoising model for both, film/video restoration and video post-processing [Werlberger et al., 2011a]. On the one hand, we show the model's ability to restore an input sequence from occurring outliers. On the other hand, the model can also be used for video post-production *e.g.* to generate intermediate frames to create an artificial slow-motion sequence. Here, the aim is to maintain the natural motion of objects throughout the whole sequence.

- Second, we also demonstrate the benefit of accurate motion vectors for an inpainting approach that is not formulated in the variational setting. This combination demonstrates the versatility of the presented optical flow approach.

Finally, Chapter **??** ends the thesis with a discussion and an outlook to further investigations.

CHAPTER
TWO

FILM AND VIDEO RESTORATION

With the growing demand on high quality broadcasting material, the grade of such available material gains in importance. Especially when screening historical film the demanded condition cannot be taken as granted. In addition, the gap between the available and the desired quality widens. This means that the restoration process gets more difficult, but also more important to obtain the desired quality for display. In the following we will give a brief introduction on the development of film and video and their differences in Section 2.1. Certainly, we do not consider this summary as complete in any kind but instead show some of the important milestones in the development of film and the relation to video and digital formats. Section 2.2 treats the copying process (telecine) from film to video or digital media. The taxonomy of considered artifacts is introduced in Section 2.4, and 2.5 gives an overview on approaches that cope with such artifacts to enhance the visual quality of image sequences. Finally, we discuss the findings and indicate the direction of the thesis with respect to the presented restoration methods.

Contents

2.1. From Film to Video

The terminus film derives from film stock, a photographic film with photosensitive layers, that was mainly used to capture motion pictures. Video is often used when talking about video tape but in general it refers to digitally stored motion pictures. Currently there is a trend towards tapeless setups, where the recorded motion pictures are directly written on *e.g.* hard disks, solid-state disks or flash memory. Despite all the technical facts and innovations in digital cinematography the choice between film and digital media has been a question of belief in the last decades. Nowadays motion pictures are still shot on film but with the increasing quality of the digital pipeline (recording equipment, *etc.*) the usage of film decreases. Currently already 50% of all motion pictures are filmed digitally.

The film stock is distinguished by its width (see Figure 2.1). The most accepted film gauges are 16 mm wide for broadcasting and 35 mm wide for motion picture production. The acceptance of 35 mm film as a universally film size is one of the reasons for its success. By contrast, several standards for video tapes defer its usage. This is also true in terms of distribution companies, where again the various video formats around the world made film the more flexible variant. Before outlining the processing of film and development towards digital media, let us first discuss the issues with historical film and why the digitalization and restoration of this material is important.

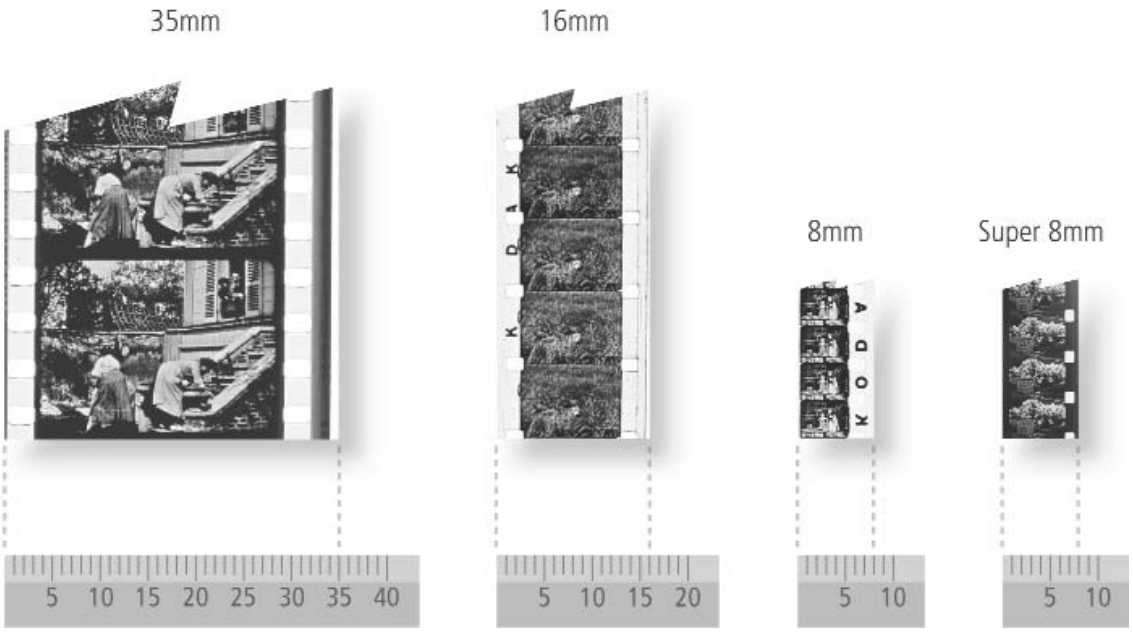


Figure 2.1.: Most common film gauges. The image is reproduced from the National Film Preservation Foundation [2004]. © [National Film Preservation Foundation, 2004]

Before the 1950s, motion picture films were shot on Nitrocellulose, which is chemically unstable and therefore unsuitable for long-term storage [Slide, 1992]. Improper storing

conditions, like wrong temperature or humidity, cause the film to decompose. Until the discovery that low temperatures stabilize the film base, the majority of early shot films were already severely damaged. It is said that *e.g.* more than 80% of American silent films are considered lost. The decomposition starts with color fading before the top layer of the film forms blisters and finally pulverizes completely. Around 1930 a safe alternative to nitrate film was developed by using cellulose triacetate or polyethylene terephthalate. But also those film bases are not spared from degradation. The acetate layer for example decomposes into acetic acid with the consequence of smelling like vinegar. This effect is typically known as vinegar syndrome (see Figure 2.2). In the 1950s an alternative to acetate film was introduced by using polyester as a film base. It is far more stable then the films based on nitrate or acetate, but as any plastic material it is still affected by a decay. Moreover, repeated playback causes mechanical abrasions. Those can range from single scratches to complete damage of multiple frames where e.g. the film reels get bonded together or when the film is torn apart.

Figure 2.2.: Combination of nitrate and triacetate film. In this case the more modern triacetate film decomposes (vinegar syndrome) and then degrades the nitrate film too.

Although the film has been the dominant medium in this industry, the process from the recording of a motion picture to broadcasting or distribution is by far more complex compared to digital cinematography. The image sequence is recorded on negative film, whereof a first positive is developed which is called the "master positive". This in turn is used to create a first duplicate ("dupe") in negative form and from that the release print is developed. Each development of a film negative may introduce subtle color changes and

decrease the quality. Due to this complex process, errors may slip in at different stages. In addition, the occurrence of noise, dirt, scratches, *etc.* accumulates over the copying processes.

This complex and costly procedure has been one of the main reasons for recording broadcasts to tape or lately to other digital media. However, for blockbusters the development took longer for several reasons. Firstly, after more than a century experience with 35 mm film, the production pipeline for a blockbuster movie is mature. The underlying industry is oriented towards film. Secondly, the flexibility of film according to achieved resolution and frame rate has been higher than with shooting video. Thirdly, the audience is used to the characteristics of the film material. We are accustomed to the soft characteristic of film and therefore material, that is entirely recorded digitally is often judged as oversharpened. Interestingly, this soft characteristic is actually due to quality loss of the copying processes. Finally, up to now a major difference of film and video was the contrast latitude of the medium. The graduation between white and black is continuous for film while it is discretized when recorded digitally. The range between shadows and high lights is measured in f-stops. For film the maximal possible dynamic range is between 8 and 12 f-stops. This value could not be reached by digital cameras so far. Only recently, state-of-the-art equipment has reached competitive values.

To the obvious question for the superior technology no general answer could be given for a long time. It was strongly dependent on the operational area whether to choose between film and video. Especially for Hollywood blockbusters shooting film dominated and is still in use. However, the trend is going towards shooting motion pictures with digital cameras. The first film that was entirely recorded with a digital camera was "Russian Ark" released in 2002. The interesting thing is that the whole movie was captured as a single shot with a digital camera and a Steadicam* setup. In 2008, two milestones of digital cinematography were released with the films "Slumdog Millionaire" and "The Curious Case of Benjamin Button". The first, "Slumdog Millionaire" was shot on multiple formats and was the first film shot partly with digital cameras winning the Academy Award for Best Cinematography. For "The Curious Case of Benjamin Button" most of the footage (approximately 95%) was recorded with digital cameras. The rather late switch towards digital technology demonstrates that not only for historic material but also for modern footage the digitalization process is of high importance.

Since the first decade of the 20^{th} century the so-called digital intermediate has had a strong influence. Although the majority of film footage has been shot on film it has been digitalized for post-production, broadcasting or distribution. More details on the digitalization process itself will be given in the next section about the Telecine. When transferring motion pictures from film to a digital media the quality should be as good as possible and occurring artifacts should be removed. Clearly, when coping with historic material this step is more important compared to modern material. Nevertheless, ToDomu:

* An apparatus for canceling the camera mans movement and stabilizing the scene.

outliers hier erklaerenoutliers can also occur when dealing with modern film material and due to the demand towards high quality the tolerated impairments are minuscule. Due to the fact that lately digital media has been coming to the fore in combination with the demand of high quality, the restoration process of film is an active field of research.

The recent trend of digitalization is a step towards the possibility to store motion pictures not exposed to typical degradation caused by long-term storage. Clearly, before (or during) digitalization is a good moment to clean the material from all existing artifacts and distortions and assure the high quality that is demanded nowadays. The downsides of digitalization like *e.g.* compression artifacts should not be kept under cover, but within this thesis digital artifacts are not explicitly treated.

Despite the improvements in quality of visually impaired material, also the reduction in bit rate can be an essential argument for broadcasting companies when it comes to video restoration. Clean material can be compressed more efficiently and therefore a restorated film can either be stored at lower bit rates with equal visual quality or when keeping the bit rate constant, the quality of the compressed material can be improved. Therefore, the restoration process goes beyond the mere sake of improving the visual quality but can also reduce broadcasting costs by achieving better compression rates [Van Roosmalen, 1999].

2.2. Telecine

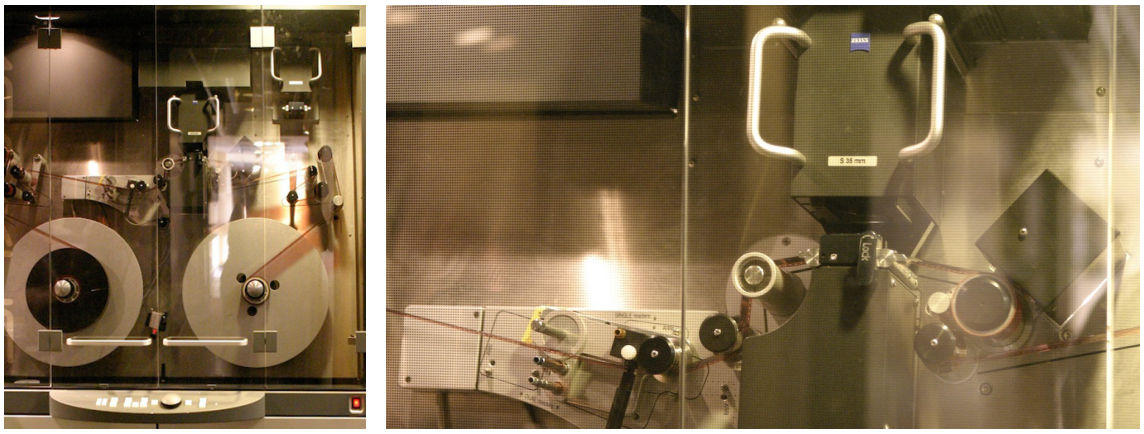


Figure 2.3.: Philips Shadow telecine released in 2000 capable of scanning film to SDTV and HDTV formats.

As stated before, motion pictures that are captured on film need to be transferred to video for *e.g.* broadcasting or digitized for post-production (*cf.* digital intermediate) or archiving. For this task a telecine is used to scan a film and convert it into an analog or digital signal which is then stored on video tape or is available for further processing. Besides transferring the material onto a different medium, often a restoration step is incorporated into the pipeline. In case of impaired material the telecine process is a

good point to improve the visual quality for further actions. Therefore, when dealing with historical or visually impaired material, a restoration stage can be integrated into the telecine process. In the era of NTSC, PAL and SECAM* an additional frame rate conversion has been integrated into the system: Film is typically shot at 24 frames per second (fps) whereas the NTSC standard broadcasts at 30 fps and PAL/SECAM at 25 fps. For the latter, the difference of a single frame per second is often neglected as the difference does not attract attention when the original 24 fps are played back with 25 fps. Ignoring the framerate differences between the 24 fps of the recorded material and the 30 fps of NTSC would result into prominent flicker. Therefore, a frame rate conversion is integrated into the telecine process.

Figure 2.3 shows a Philips Shadow telecine from the year 2000 with a line array CCD scanner. The exposed film image is lighted and a prism separates the light into the primary colors (red, green and blue) which are separately recorded by the CCD array. With the rise of digital intermediate systems, motion picture film scanners have been introduced. Two different types of scanners are available: Evolved from the telecine process the scanning process is performed while the film moves by. A possibility to scan at higher resolution is enabled by scanning each frame of the film stock individually. Resolution up to 8K† can be achieved. The emerging data is stored on a frame-by-frame basis and is then available for further processing. Today, scanning is more important than the telecine process.

2.3. Restoration

The previous Section has already highlighted the necessity of the restoration task when *e.g.* film is digitalized or historical movies are processed. In the following we will give some more details on restoration, introduce the taxonomy for occurring artifacts and review some relevant restoration approaches.

With the introduction of film scanners and the possibility to digitize impaired film material, the emphasis gradually shifted towards digital film restoration. The restoration of motion pictures ideally is a fully-automatic procedure, because manually repairing frame by frame would be a cumbersome task. Considering a frame rate of 24 fps, the restoration of every single frame is an intractable process. Unfortunately, the methods for recovering from defects in a video sequence are mostly not a trivial extension of 1D signal processing or 2D image restoration methods towards 3D video signals [Kokaram, 2003]. Hence, utilizing spatio-temporal information of the motion pictures [Biemond et al., 1987, Efstratiadis and Katsaggelos, 1990, Erdem et al., 1992, Van Roosmalen, 1999, Kokaram, 2003] and imposing motion compensation before applying temporal filters [Dennis, 1980, Dubois and Sabri, 1984, Martinez and Lim, 1985, 1989] has already been a valuable improvement when it comes to restoration quality. For further details on relevant restoration approaches see

* Different encoding systems for broadcasting television. † 8K video format has a resolution of 8192×4320 pixels. In todays theaters mostly 2K (2048×1080) or 4K (4196×2160) resolution is used and for the home environment 1080p (1920×1080) is the current standard.

Section 2.5.

In addition the restoration process is a balancing act of removing all unwanted impairments without destroying the film characteristics. The discussion of the restored film *Metropolis* (1927) is a famous example for such a controversy. In the optimal case, all impairments are removed but the movie is still recognized as historical with all its characteristics. As restoration almost always includes some kind of filtering operation the results are often over-smoothed and this degrades the typical film characteristics, especially the film grain. In order to distinguish film artifacts from the film's characteristic and to establish a common formulation we now introduce the used taxonomy of some relevant impairments.

2.4. Artifacts

Degradation of film and video material can have different causes. Kokaram [2003] distinguished between impairments that are caused by film degradation and mechanical abrasion from playback or the telecine process. During the copying process also more complex degradations like synchronization errors or the loss of contrast can occur. However, the artifacts are visually similar whatever they are caused by. Hence, we do not make this distinction and treat the family of artifacts as a whole and neglect the cause of the impairments. In the sense of Kokaram [2003] and Van Roosmalen [1999] we divide the artifacts into four major impairment groups:

- Missing data
- Noise
- Flicker
- Image vibration

While the distinction of artifacts could be done at a finer scale, we refer to this classification because we are mainly interested in the overall characteristics of these impairment families and the resultant effects on the presented algorithms. In the following we give a rough description of the impairment groups. Figure 2.4 shows some typical film artifacts from the impairment families *missing data* and *noise*. At the current point, *flicker* and *image vibration* is difficult to visualize with still images as those artifacts are mainly pronounced over the temporal component. In the following we give some details on how those impairment families can be characterized:

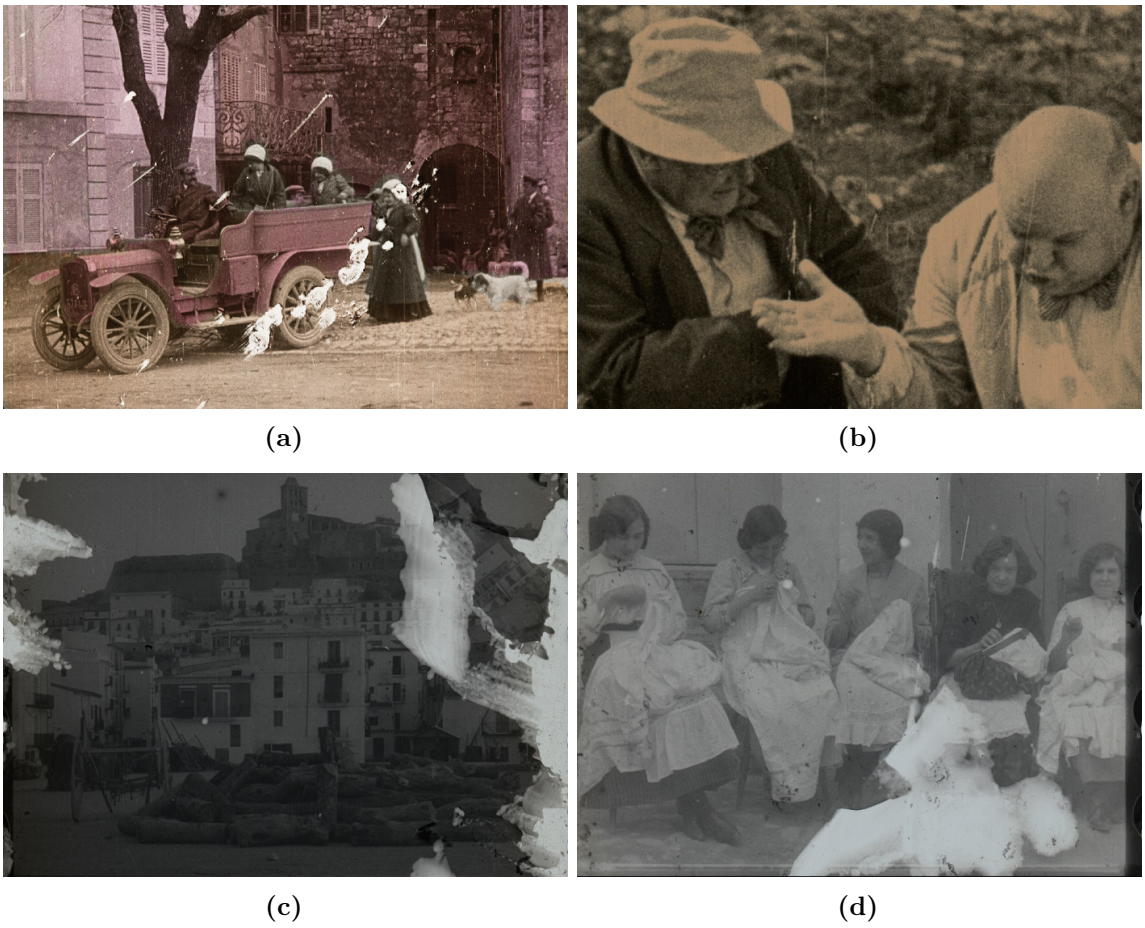
Missing data describes areas where no image information can be deduced. This can be due to scratches or speckles, destructed areas, and completely lost frames for the worst case. Such impairments can emerge from physical abrasion of the film layer caused by the mechanical transportation in the playback apparatus. More severe deterioration can

occur in areas where film reels get bonded together while played back which can result into completely destructed areas. Besides mechanical abrasion, dust and dirt on the lens or film while recording can lead to typical black and white spots in the recorded material. Also pollution when storing or copying the film can produce such speckle artifacts. As discussed in the Motivation, another cause for destructed regions is film degradation due to chemical decomposition of the film base material. This can lead to small cracks but can also escalate into big destructed regions and even unrecoverable frames.

Noise is difficult to judge for historic film material. Due to film grain the presence of a certain amount of granularity is an essential characteristic. Simply removing all the noise will lead to the impression of over-smoothed material destroying the typical film characteristic. Though, noise can become too severe especially for very old material that was copied multiple times. For every replication the noise level increases as the copying process can add some more impairments when no restoration is performed. A widely used technique to maintain the film grain is to smooth the video sequence and afterwards superimpose some artificial grain. Clearly, a better approach is to only remove the severe artifacts and leave some noise/grain in the image to maintain the typical film characteristic.

Flicker is an effect mainly confined to material recorded on film. It describes brightness changes from one frame to the next and is mostly caused by different exposure times. This is a very common impairment for the black and white era, and was caused by the cameras exhibiting unconstant film exposures. Nevertheless, it is also possible that flicker has been introduced by degrading film layers. The main difference between the two variants is the duration where the flickering occurs. If caused by film degradation, the brightness changes are often restricted to parts of film and often only parts of the frame are afflicted. In contrast, inconsistent exposure times of the camera equipment effects the whole movie.

Image vibration describes the effect of an unsteady picture over time. This can be caused directly by unsteadiness in the camera work by *e.g.* filming with a handheld camera. On a smaller scale such shaking effects can be induced by inaccurate transportation of the film while telecined or played back. Degraded transport perforations can cause unsteady pictures during playback although the movie is technically fine.



(a) **(b)**

(c) **(d)**

Figure 2.4.: Examples for film defects. (a) shows typical blob artifacts where no data information is available. In addition it is a good example for noise and scratches which do also occur in (b). (c) and (d) have large destructed areas.

2.5. Restoration Methods

In this section we will present an assortment of approaches to visually improve degraded motion pictures. At first, we distinguish between algorithms that are applied to single frames and such that explicitly use the temporal information available from video sequences.

Standard image denoising methods can be applied on a frame-by-frame basis to a video sequence. However, this neglects the temporal component and such methods obviously do not profit from this additional information in the data. In addition, good looking restoration results for a single image do not necessarily satisfy when seen in the context of a whole sequence. A straightforward possibility to incorporate the temporal component is frame averaging. This is a simple and fast approach to reduce noise from image sequences by taking *e.g.* the mean of the color values over a certain temporal window. The results are pleasant on static objects, however, when motion is present it introduces additional

blur in these areas.

For this reason, McMann et al. [1978], Dennis [1980], Dubois and Sabri [1984] narrowed the application of frame averaging towards non-moving objects. Hence, the filtering operation is only applied to objects where no motion is detected in order to avoid introducing additional blur. The major drawback of this approach is the limitation of denoising towards static parts of the scene. When non-moving objects begin to move, the noise increases in those areas as the filtering is then prohibited. In addition, the moving elements attract the observers attention and are therefore particularly noticed. For this reason it is essential to handle the moving parts of the scene and also apply some kind of filtering there. This enables the restoration of the whole sequence and this certainly is a desirable goal.

Especially Huang [1981], Dennis [1980], Dubois and Sabri [1984], Martinez and Lim [1985] depicted the importance of motion estimation when applying temporal filters. In terms of different filters, a linear and a median filter [Tukey, 1977] is applied on the motion compensated frames and it is shown that especially the median filter performs well for removing defects in presence of moving objects. For small velocities even moving edges are preserved by the median filter as it is the case for the 2D variant of the filter for discontinuities. This variant works nicely if the motion compensation yields correct results. However, for larger motion and complex sequences it often occurs that motion estimation fails in certain areas. In such areas it is better to adapt the smoothing to the quality of the motion estimator. If one is certain that the motion is valid, the smoothing can be performed, but when it fails the temporal denoising should be reduced. In this sense, Dubois and Sabri [1984] introduced an additional weighting parameter that adapts the denoising towards the accuracy of the estimated motion vectors.

Incorporating motion detectors and estimators to guide temporal filtering improves the sharpness of the denoised sequences. Still a combination of both, temporal and the classical spatial filtering might even improve over both approaches. Efstratiadis and Katsaggelos [1990], Ozkan [1991, 1992], Erdem et al. [1992], Kokaram [2003] proposed 3D operators to combine the benefits of noise reducing spatial filters and the detail preserving tendency of temporal filters. Furthermore, spatio-temporal Wiener filtering has been used in Martinez and Lim [1985], Martinez [1987], Erdem et al. [1992], Katsaggelos et al. [1989], Kokaram [2003] incorporating motion estimation.

2.6. Motion Estimation

A robust and accurate motion estimation technique is essential for the resultant quality of the restoration approach. Despite the long history of research motion estimation remains an unsolved problem and there exists no algorithm that achieves perfect results. Already the schematic presentation of a restoration pipeline in the first chapter in Figure 1.1 emphasized the important of an accurate algorithm for motion estimation. Often, optical flow is utilized to estimate the pixels' movement in an image sequence. Optical flow and motion estimation can only be equated under certain circumstances. While optical flow

describes the motion of the image brightness pattern while motion estimation itself refers to the actual movement of specific points in the image. In Chapter 4 we will discuss the characteristics of optical flow in more detail and give some more details on using optical flow for motion estimation. Furthermore we will present a variational approach that has been heavily used throughout the last decades for computing optical flow vectors in sequent images. As the restoration quality is heavily dependent on the quality of the motion estimator we will then discuss various extensions in Chapter 5. But before discussing the framework for optical flow estimation, we will introduce the mathematical foundations in Chapter 3 and give an introduction to variational approaches.

CHAPTER
THREE

VARIATIONAL METHODS IN IMAGING

The following chapter is devoted to a brief review on variational methods in the field of image processing. The aim is to introduce the mathematical framework for the presented models and to put together the common notation and operations. First, we introduce some details on the mathematical notation, then give an overview on variational methods for imaging problems and present some basic concepts on convex analysis and optimization.

Contents

3.1. Prerequisites (mathematical notation)

Before we start revisiting the mathematical foundations of variational methods, we first introduce some of the used notation and operations that will be utilized throughout the following sections. Furthermore, we give some details on the discretization of the defined operators for the usage in numerical schemes applied to functions that are defined in the continuous domain.

In order to use a spatio-temporal intensity function $I(x,t)$ in numerical schemes we define single images of this space-time volume as $I_1(x), \ldots, I_K(x)$ which are defined at discrete time positions $t = \{1, \ldots, K\}$. x denotes the spatial position as the coordinate vector in an image that is defined as

$$x = \begin{pmatrix} x^1 \\ x^2 \end{pmatrix}.$$

The sequent input images $I_k(x), k = 1, \ldots, K$ are defined on a regular Cartesian grid of size $M \times N$,

$$\Omega^h = \{(ih, jh) : 1 \leq i \leq M, 1 \leq j \leq N\}\,, \tag{3.1}$$

where h denotes the size of the pixels (pixel spacing). For all further considerations we assume a regular and homogeneous pixel grid with pixels of size $h = 1$. This simplifies the definition of the pixel grid to

$$\Omega = \{(i,j) : 1 \leq i \leq M, 1 \leq j \leq N\}\,. \tag{3.2}$$

Hence, the discrete pixel position within an image is given by (i,j). Next, we define the finite dimensional vector spaces

$$\begin{aligned} X &= \mathbb{R}^{MN} \\ Y &= X \times X = \mathbb{R}^{2MN} \\ Z &= Y \times Y = \mathbb{R}^{4MN}\,. \end{aligned} \tag{3.3}$$

With the vectors

$$\begin{aligned} a &\in X && \text{and} && \hat{a} \in X\,, \\ b &= \left(b^1, b^2\right) \in Y && \text{and} && \hat{b} = \left(\hat{b}^1, \hat{b}^2\right) \in Y\,, \\ c &= \left(c^1, c^2, c^3, c^4\right) \in Z && \text{and} && \hat{c} = \left(\hat{c}^1, \hat{c}^2, \hat{c}^3, \hat{c}^4\right) \in Z\,. \end{aligned}$$

the standard scalar products are defined as

$$\langle a, \hat{a} \rangle_X = \sum_{j=1}^{N} \sum_{i=1}^{M} a_{i,j} \hat{a}_{i,j} \,, \tag{3.4}$$

$$\left\langle b, \hat{b} \right\rangle_Y = \sum_{j=1}^{N} \sum_{i=1}^{M} \left(b^1_{i,j} \hat{b}^1_{i,j} + b^2_{i,j} \hat{b}^2_{i,j} \right) \,, \qquad \text{and} \tag{3.5}$$

$$\langle c, \hat{c} \rangle_Z = \sum_{j=1}^{N} \sum_{i=1}^{M} \left(c^1_{i,j} \hat{c}^1_{i,j} + c^2_{i,j} \hat{c}^2_{i,j} + c^3_{i,j} \hat{c}^3_{i,j} + c^4_{i,j} \hat{c}^4_{i,j} \right) \,. \tag{3.6}$$

Moreover we define the ℓ_p-norm as

$$||a||_p = \left(\sum_{j=1}^{N} \sum_{i=1}^{M} |a_{i,j}|^p \right)^{1/p} \,. \tag{3.7}$$

Frequently used norms are the ℓ_1, the ℓ_2 (or Euclidean) norm and the ℓ_∞ (or maximum) norm which are defined as

$$||a||_1 = \sum_{j=1}^{N} \sum_{i=1}^{M} |a_{i,j}| \,, \tag{3.8}$$

$$||a||_2 = \sqrt{\langle a, a \rangle_X} \qquad \text{and} \tag{3.9}$$

$$||a||_\infty = \max\left(a_{i,j}\right) \,. \tag{3.10}$$

The gradient operator $\nabla : X \to Y$ is defined as

$$(\nabla a)_{i,j} = \begin{pmatrix} \left(\delta^+_{x^1} a\right)_{i,j} \\ \left(\delta^+_{x^2} a\right)_{i,j} \end{pmatrix} , \tag{3.11}$$

When used in a numerical scheme $\delta^+_{x^1/x^2}$ are standard forward differences computed on the discrete lattice with Neumann boundary conditions defined as

$$\begin{aligned} (\delta^+_{x^1} a)_{i,j} &= \begin{cases} a_{i+1,j} - a_{i,j} & \text{if} \quad i < M \\ 0 & \text{else} \,, \end{cases} \\ (\delta^+_{x^2} a)_{i,j} &= \begin{cases} a_{i,j+1} - a_{i,j} & \text{if} \quad j < N \\ 0 & \text{else} \,. \end{cases} \end{aligned} \tag{3.12}$$

Furthermore, we introduce the adjoint operator (denoted as *) of the gradient as $\nabla^* : Y \to X$ being the negative divergence operator through the identity

$$\langle \nabla a, b \rangle_Y \equiv \langle a, \nabla^* b \rangle_X = \langle a, -\mathrm{div}\, b \rangle_X \,. \tag{3.13}$$

To compute the divergence in a numerical scheme we use standard finite backward differences

$$-\left(\operatorname{div} b\right)_{i,j} = \left(\delta^-_{x^1} b^1\right)_{i,j} + \left(\delta^-_{x^2} b^2\right)_{i,j} . \tag{3.14}$$

With the backward differences $\delta^-_{x^1/x^2}$ defined as

$$\begin{aligned}
(\delta^-_{x^1} b^1)_{i,j} &= \begin{cases} 0 & \text{if} \quad m = 1 \\ b^1_{i,j} - b^1_{i-1,i} & \text{if} \quad 1 < i < M \\ b^1_{i-1,i} & \text{else} \ , \end{cases} \\
(\delta^-_{x^2} b^2)_{i,j} &= \begin{cases} 0 & \text{if} \quad n = 1 \\ b^2_{i,j} - b^2_{i,j-1} & \text{if} \quad 1 < j < N \\ b^2_{i,j-1} & \text{else} \ . \end{cases}
\end{aligned} \tag{3.15}$$

Similarly to (3.11), the gradient in Y, $\nabla : Y \to Z$ is defined as

$$\left(\nabla b\right)_{i,j} = \left(\left(\delta^+_{x^1} b^1\right)_{i,j}, \left(\delta^+_{x^2} b^1\right)_{i,j}, \left(\delta^+_{x^1} b^2\right)_{i,j}, \left(\delta^+_{x^2} b^2\right)_{i,j}\right)^T . \tag{3.16}$$

Subsequently, the adjoint operator of the gradient $-\operatorname{div} : Z \to Y$ is defined as

$$-\left(\operatorname{div} c\right)_{i,j} = \left(\left(\delta^-_{x^1} c^1\right)_{i,j} + \left(\delta^-_{x^2} c^2\right)_{i,j}, \left(\delta^-_{x^1} c^3\right)_{i,j} + \left(\delta^-_{x^2} c^4\right)_{i,j}\right)^T . \tag{3.17}$$

3.2. Variational Methods

Most of the problems in imaging and computer vision are inverse problems. According to Keller [1976] inverse problems are defined as follows:

> *We call two problems inverses of one another if the formulation of each involves all or part of the solution of the other. Often, for historical reasons, one of the two problems has been studied extensively for some time, while the other is newer and not so well understood. In such cases, the former problem is called the direct problem, while the latter is called the inverse problem. [Keller, 1976]*

The direct problem is often called forward problem. In other words, the forward problem defines *e.g.* generating data with a given parametric model, whereas the inverse problem would be the reconstruction of the model parameters from the given data. Such data can be acquired by *e.g.* some measurements, observations, or as in the case of computer vision, images or video streams. Applications in computer vision demonstrate the difficulty and ambiguity of inverse problems. Considering a given image, it is difficult to recover the original data which could be demanded in the form of appearance or 3D structure.

Typically, one of the two problems is ill-posed in the sense of Hadamard [1902] who defines problems to be well-posed if

1. a solution exists (existence),

2. it is the only solution (uniqueness),

3. and the solution depends continuously on the data of the problem (stability).

If one out of the two inverse problems is an ill-posed problem, that one is denoted as inverse, whereas the second one is then the direct problem.

As implied in the description of forward and inverse problems the usual tasks we have to deal with in computer vision and image processing are ill-posed. In order to make such problems well-posed, additional information is incorporated into the model. This is often referred to as prior assumption. A common form of formulating an imaging problem is to define an energy functional of the form

$$\min_{v} E(v(x)) . \tag{3.18}$$

Its stationary point (in (3.18) defined as its minimum) should describe the desired solution. In order to apply sophisticated optimization schemes, the formulated functional should be convex. The next question is about the assembly of such an optimization problem. Usually some kind of data fidelity ($\mathcal{D}$) is defined to describe the desired objective. As mentioned, mostly this will form an ill-posed problem and some prior knowledge must be added. This can be done in form of an additional regularization term ($\mathcal{R}$) (a.k.a. prior or smoothness term) which results in the overall optimization problem

$$\begin{aligned} E(v(x)) &= \mathcal{R}(v(x)) + \lambda \mathcal{D}(v(x), f(x)) \\ &= \int_{\Omega} \psi_{\mathcal{R}} \left(x, v(x), Dv(x), \ldots, D^{l} v(x) \right) \mathrm{d}x + \lambda \int_{\Omega} \psi_{\mathcal{D}} \left(\delta(v(x), f(x)) \right) \mathrm{d}x . \end{aligned} \tag{3.19}$$

On the one hand, the data term $\mathcal{D}(\cdot)$ measures the data fidelity of the desired signal $v(x)$ to the given data $f(x)$ by optimizing a penalized similarity measure $\delta(\cdot)$. On the other hand, the regularizer $\mathcal{R}(\cdot)$ imposes some additional knowledge (*e.g.* enforcing the signal to be smooth) to make the optimization problem a well-posed problem. The order l of the regularizer represents the highest order of derivative of $v(x)$ used in the prior term. Note that with a sufficiently smooth $v(x)$ the linear operator D can be replaced by the gradient operator ∇.

As the specifications of such a variational formulation suggest, solving a specific problem within this framework becomes a twofold task:

- At first, an appropriate energy functional (model) must be defined, where the stationary point yields the desired solution. Mostly it is a combination of a data term, defining the problem and the relation to the given data, and a regularization term to obtain a well-posed problem. Section 3.3 gives an example on utilizing such an approach for the image denoising task.

- Second, to obtain this solution, the optimization problem must be solved. An appropriate algorithm is applied and in the best case a global optimum of the function is computed. More details on convex function and respective optimization techniques are given in Section 3.5 and 3.6.

3.3. Variational Image Denoising

As an illustrative example to model a variational formulation, we have a closer look on the image denoising task which is well studied and nicely suits to a variational formulation. By means of commonly used formulations we explain the constellation of such a problem.

The task is to extract the clean image $\check{v}(x)$ from the given data corrupted by noise η

$$f(x) = \check{v}(x) + \eta\,. \tag{3.20}$$

As the noise is usually unknown, the first step is to model an appropriate data term. A common approach is to minimize the linear least squares fit of the residual

$$\mathcal{D}(v(x), f(x)) = \int_\Omega (v(x) - f(x))^2 \,\mathrm{d}x\,. \tag{3.21}$$

Next, to make the problem well-posed, additional prior knowledge has to be added to (3.21). An often used variant is to enforce the fluctuations of high frequencies in the image to be small by incorporating a quadratic penalization of the first-order derivatives of $v(x)$:

$$\mathcal{R}(v(x)) = \int_\Omega |\nabla v|^2 \,\mathrm{d}x \tag{3.22}$$

The first articles using such quadratic regularizers in the context of solving inverse problems date back to Tikhonov [1963] and are therefore often referred to as Tikhonov regularization. Combining the data term (3.21) and the quadratic regularizer (3.22) yields the so-called Tikhonov model, an energy optimization problem of the form

$$\min_v \int_\Omega \frac{1}{2} |\nabla v|^2 \,\mathrm{d}x + \frac{\lambda}{2} \int_\Omega (v(x) - f(x))^2 \,\mathrm{d}x\,. \tag{3.23}$$

Figure 3.1 shows an exemplar result of the Tikhonov model. At first, synthetic Gaussian noise is superimposed on a clean image (Figure 3.1b) to obtain the noisy input image (Figure 3.1a) that is used as input for the algorithm. Although the noise is removed, the oversmoothing effect in the result (Figure 3.1c) immediately attracts attention.

The smoothing is accredited with the regularization term. By design, the used regularization penalizes the first-order derivatives of the result v in a quadratic way. Considering statistical evaluation on the distribution of a linear operator applied to natural images (*e.g.* a gradient operator), Huang and Mumford [1999], and Weiss and Freeman [2007] showed that the quadratic smoothness assumption (Figure 3.2b) does not fit properly

(a) Noisy input image. **(b)** Original image. **(c)** Tikhonov

Figure 3.1.: Variational image denoising with the Tikhonov model (3.23) with $\lambda = 0.05$.

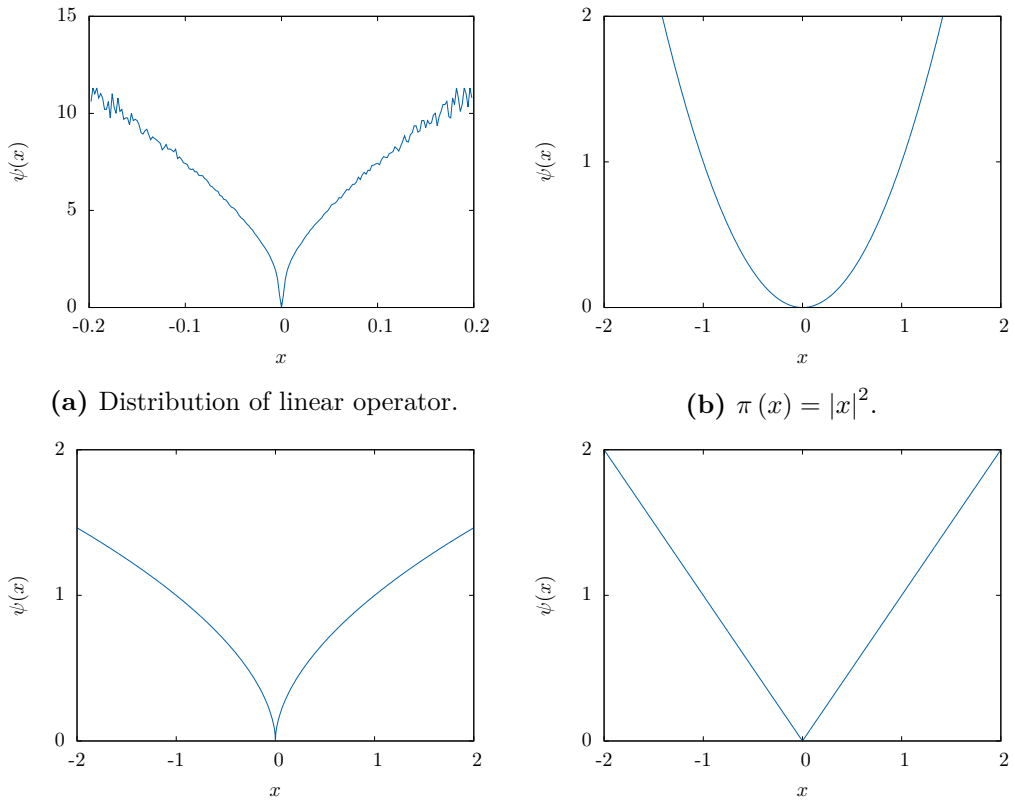
to the learned distribution shown in Figure 3.2a. A regularization that would fit such a given data is depicted in Figure 3.2c with the penalty function $\Psi_{\mathcal{R}}(s) = |s|^{0.55}$. Unlike the quadratic regularization this penalty function is non-convex and therefore not desirable for integration into the variational framework.

Already the seminal work by Rudin et al. [1992] has drawn the attention to use an ℓ_1-norm for regularizing the variational denoising problem. This yields the so-called ROF model with the objective function

$$\min_v \int_\Omega |\nabla v| \, \mathrm{d}x + \frac{\lambda}{2} \int_\Omega ||v(x) - f(x)||_2^2 \, \mathrm{d}x \,, \tag{3.24}$$

assuming that v is suffiently smooth. The utilized regularization $\mathcal{R}(\nabla v)$ is often referred to as total variation. The ℓ_1-norm, shown in Figure 3.2d bridges the gap between convexity and non-convexity. For optimization, the discontinuity at zero is a bit tricky but in general the model is very well-suited for regularizing image processing problems. For more details on the total variation we refer to Section 3.4. Replacing the Tikhonov regularizer with the total variation enables the denoising approach to better preserve image edges which results into sharper and therefore more natural results. A comparison of the denoising result with the Tikhonov model ($\lambda = 0.05$) and the ROF model ($\lambda = 5$) is given in Figure 3.3. Clearly, the result achieved by the ROF model features more image details and is more desirable.

The property of preserving image edges enables the total variation to be a commonly used regularization in variational approaches. In the following section we will give some more information and details on this regularization and furthermore we will utilize total



(a) Distribution of linear operator.

(b) $\pi(x) = |x|^2$.

(c) $\psi(x) = |x|^{0.55}$.

(d) $\psi(x) = |x|$.

Figure 3.2.: Distribution of a linear operator applied on natural images and different prior terms.

variation-like smoothness priors throughout this thesis.

(a) Noisy input image. **(b)** Tikhonov **(c)** ROF

Figure 3.3.: Comparison of denoising results obtain by (b) the Tikhonov model (3.23) with $\lambda = 0.05$ and (c) the ROF model (3.24) with $\lambda = 5$.

3.4. The Total Variation

As depicted for the image denoising task in Section 3.3, the total variation (TV) for regularizing image processing tasks features the benefit of preserving image edges. This is a useful property not only to image denoising but for any kind of image processing task. Indeed a TV-like regularization has been used by Shulman and Hervé [1989] for optical flow estimation. An introduction to TV in imaging and its mathematical properties is given in [Chambolle et al., 2010] and references therein. We refer the reader for further details on the topic. To outline the most important facts of the TV we define $v \in BV(\Omega)$ with $\Omega \subset \mathbb{R}^2$. Then the TV is defined by its dual formulation (more details on duality see Section 3.5.3) as

$$\mathcal{R}(v(x)) = \sup_q \left\{ -\int_\Omega v(x) \operatorname{div} q(x) \, \mathrm{d}x : \quad q \in \mathcal{C}^\infty, \quad ||q(x)||_\infty \leq 1 \right\}. \tag{3.25}$$

In the given case of v being a function of bounded variation the Divergence theorem is defined as

$$-\int_\Omega v(x) \operatorname{div} q(x) \, \mathrm{d}x = \int_\Omega q(x) \, Dv(x) \, . \tag{3.26}$$

Here, q is the dual variable and D denotes the BV gradient which is defined as a combination of the gradient in the continuous parts and the jump parts. Moreover, for smooth functions

$u \in W^{1,1}(\Omega)$ (3.27) gets

$$-\int_{\Omega} v(x) \operatorname{div} q(x)\, \mathrm{d}x = \int_{\Omega} q(x) \nabla v(x)\, \mathrm{d}x\,. \tag{3.27}$$

Hence, for a sufficiently smooth v, the TV regularization is defined as

$$\mathcal{R}(v(x)) = \int_{\Omega} |\nabla v(x)|\, \mathrm{d}x\,. \tag{3.28}$$

For convenience we stick to the latter notation in the rest of the thesis and assume that the function is sufficiently smooth.

To apply numerical schemes and optimize energy functionals that incorporate the TV, a commonly used version of the discrete TV is defined as

$$||\nabla v||_{2,1} = \sum_{j=1}^{N}\sum_{i=1}^{M} \sqrt{\left(\delta^{+}_{x^1} v\right)^2_{(i,j)} + \left(\delta^{+}_{x^2} v\right)^2_{(i,j)}} \tag{3.29}$$

The major benefits of using the TV regularizer in variational imaging problems has already been touched in previous section. By using the TV norm as smoothness prior the functional remains convex and still is able to preserve discontinuities. For the previous example of denoising, the discontinuities depict image edges but it is also important for other tasks like motion estimation to enhance motion discontinuities.

3.5. Convex Analysis

After devising an objective function, the task is to compute the desired result by minimizing the given energy functional. We restrict ourselves to convex problems. There, it is guaranteed that if a minimum exists, the minimum is a global one and not local. For this reason we give an overview on the properties of convex analysis before diving into the topic of convex optimization. For a more complete disquisition on convex analysis we refer to the textbooks from Rockafellar and Wets [1997], Rockafellar [1997], Bertsekas et al. [2003], and Boyd and Vandenberghe [2004] and references therein. In the following we give some details on convexity, convex sets and convex functions.

3.5.1. Convex Sets

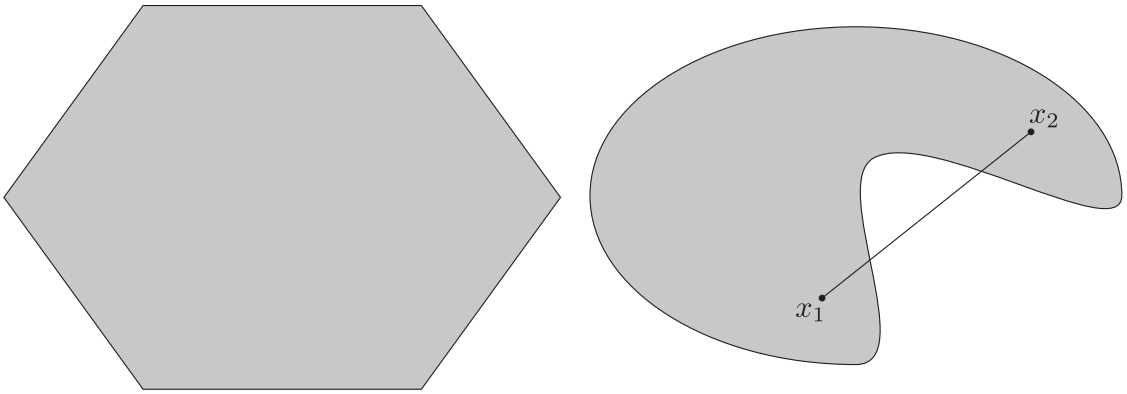
A set C is a convex set if it entirely contains all line segments joining any two points $x_1, x_2 \in C$ meaning that for all θ in the interval $[0, 1]$ all points on the line

$$\theta x_1 + (1 - \theta) x_2 \tag{3.30}$$

are again included in the convex set C. Figure 3.4 sketches an illustrative example for a convex and a non-convex set. The set of all convex combinations of the k points

$x_1, \ldots, x_k \in C$ span the convex hull of a set C which is convex by definition and formulated as

$$\left\{ \sum_{i=1}^{k} \theta_k x_k \right\}, \qquad \text{with} \quad \theta_i \geq 0 \quad \text{and} \quad \sum_{i=1}^{k} \theta_i = 1. \tag{3.31}$$



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

Figure 3.4.: Illustrative example of a convex and b non-convex set.

Furthermore, a set defined as

$$\theta_1 x_1 + \theta_2 x_2 \in C \tag{3.32}$$

with $x_1, x_2 \in C$ and $\theta_1, \theta_2 \geq 0$ is denoted as a convex cone. Similarly to the convex hull the definition of a conic convex hull is given as the set of all conic combinations of points in the set C by

$$\left\{ \sum_{i=1}^{k} \theta_k x_k \right\}, \quad \text{with} \qquad \theta_i \geq 0. \tag{3.33}$$

Operations on convex sets that preserve convexity are:

- The intersection of convex sets $\bigcap_i C_i$ is again a convex set.
- The sum $C_1 + C_2 = \{x_1 + x_2 | x_1 \in C_1, x_2 \in C_2\}$ of two convex sets C_1, C_2 is convex.
- Multiplying a convex set C by a scalar λ yields a convex set λC. Furthermore, for $\lambda_1 \geq 0, \lambda_2 \geq 0$ one obtains $(\lambda_1 + \lambda_2) C = \lambda_1 C + \lambda_2 C$.

3.5.2. Convex Functions

Let $F(x) : \mathbb{R}^n \to \mathbb{R}$ be a convex function if the assumption

$$F(\theta x_1 + (1 - \theta) x_2) \leq \theta F(x_1) + (1 - \theta) F(x_2) \tag{3.34}$$

holds for any pair of points $x_1, x_2 \in C$ for any θ with $0 \leq \theta \leq 1$. Furthermore, the domain of the function $F(x)$ is a convex set C. Geometrically, this defines a line connecting the

two points x_1, x_2 which lies entirely above the functions graph (see Figure 3.5). When considering the whole set of points lying on or above the graph, the function is convex if this set is a convex set. This so-called epigraph of $F(x)$ ($\operatorname{epi} F(x)$) is formally defined as the set

$$\operatorname{epi} F(x) = \{(x,t) \,|x \in \mathbb{R}^n, F(x) \leq t\} \ . \tag{3.35}$$

In other words, the epigraph of a function defines the area "above" the functional. A geometric interpretation of the epigraph is depicted in Figure 3.6. Hence, $F(x)$ is convex on $C \subset \mathbb{R}^n$, if $\operatorname{epi} F(x)$ is convex on a subset of $\mathbb{R}^{n+1}$. The function $F(x)$ is called strictly convex if (3.34) is strict in the sense that $x_1 \neq x_2$ implying $0 < \theta < 1$. If the function $-F$ is convex, F is called a concave function.

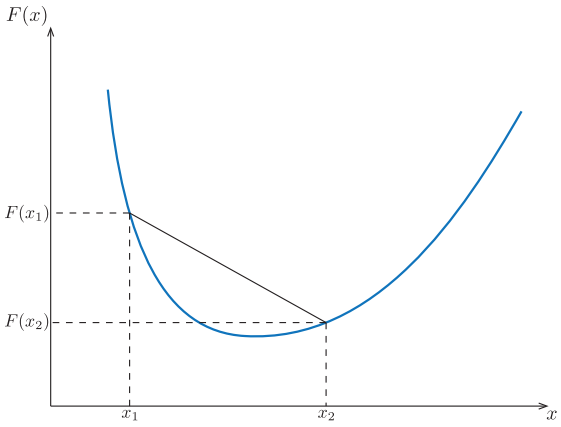


Figure 3.5.: Convex function.

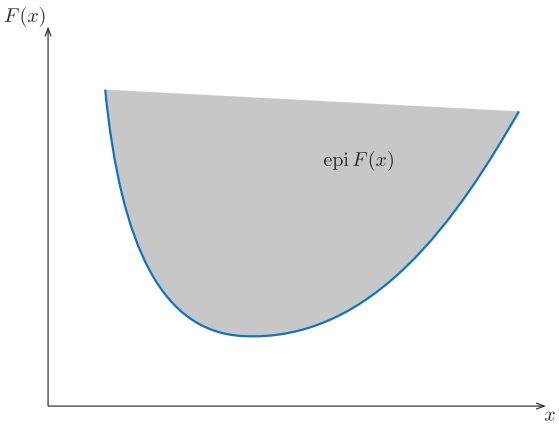


Figure 3.6.: Epigraph of a convex function.

In the following we show some properties of convexity and operations on convex (or concave) functions that maintain convexity (concavity) on the newly generated functions:

- If a minimum exists, convexity ensures that the minimum is not local but global.
- For strictly convex functions the minimum, if existent, is unique.
- By forming the convex envelope* of a non-convex function, the global minima are maintained.
- The set of global minima again form a convex set.
- The nonnegative scaling λF of a convex function F with a scalar $\lambda \geq 0$ is again convex.
- The sum of two convex functions $F = F_1 + F_2$ remains convex.

* The convex envelope $G(x)$ of a function $F(x)$ is its largest possible convex underestimator s.t. $G(x) \leq F(x) \quad \forall x \in \Omega$.

- Combining the two latter operations on a set of convex functions $F_1, \ldots, F_k$ with nonnegative weights $\lambda_1, \ldots, \lambda_k$ yields the convex cone

$$F = \sum_{i=1}^{k} \lambda_i F_i \,. \tag{3.36}$$

- Considering an affine mapping of $F : \mathbb{R}^n \to \mathbb{R}$ with a matrix $A \in \mathbb{R}^{n \times m}$ and a vector $b \in \mathbb{R}^n$ of the form

$$G(x) = F(Ax + b)\,, \tag{3.37}$$

  then the new function $G : \mathbb{R}^m \to \mathbb{R}$ is convex if F is a convex function.

- Computing the pointwise maximum of a set of functions $F_1, \ldots, F_k$ with

$$F(x) = \max\{F_1(x), \ldots, F_k(x)\}\,, \tag{3.38}$$

  results in a convex function F if $F_1, \ldots, F_k$ are convex.

3.5.3. The Conjugate Function

In this section we introduce an operation that is called conjugate transform, Fenchel transform or Legendre-Fenchel transform. For our further investigations we stick to the terminus of conjugate transform or simply the dual. More details on duality is given in the relevant literature [Rockafellar, 1997, Rockafellar and Wets, 1997, Boyd and Vandenberghe, 2004, Bertsekas et al., 2003].

Consider a function $F : \mathbb{R}^n \to \mathbb{R}$ and $F^* : \mathbb{R}^n \to \mathbb{R}$ with the relation

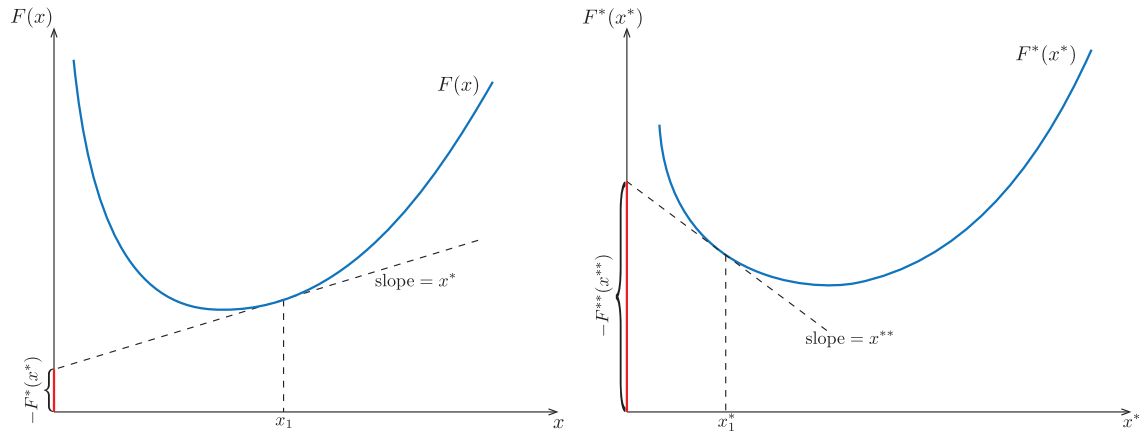
$$F^*(x^*) = \sup_{x \in \operatorname{dom} F} \{\langle x^*, x\rangle - F(x)\}\,, \tag{3.39}$$

then F^* is called the conjugate function of F. Moreover, the dual variable x^* is related to the function's subgradient ∂F representing the slope of the original function F. In succession the conjugate function F^* is dependent on the slope of F and is always a convex function independent of the shape of F (*cf.* convexity preserving pointwise max operation in Section 3.5.2). The geometric relations of the conjugate transform are shown in Figure 3.7.

Applying the conjugate transform to the conjugate function $F^*(x^*)$ results into the so-called biconjugate F^{**} of F defined as

$$F^{**}(x^{**}) = \sup_{x^* \in \operatorname{dom} F} \{\langle x^{**}, x^*\rangle - F^*(x^*)\}\,, \tag{3.40}$$

Geometrically speaking, the biconjugate forms the convex envelope of its original function and is convex by definition, independent of the shape of F. Specifically, if F is a closed convex function then the biconjugate F^{**} is equal to the original function F. In such



(a) $F(x) \rightarrow F^*(x)$ **(b)** $F^*(x) \rightarrow F^{**}(x)$

Figure 3.7.: Illustration of the conjugate transform's geometric principles. Computing the dual transfer points defined in $F(x)$ into slopes of $F^*(x^*)$ (a) $\rightarrow$ (b) while slopes of $F^*(x)$ become points in $F(x)$ (b) $\rightarrow$ (a). Note, for the shown example $x^{**} = x$ in (b) because of the convexity of $F(x)$.

a setting the conjugacy transform is a symmetric operation. According to the Fenchel inequality

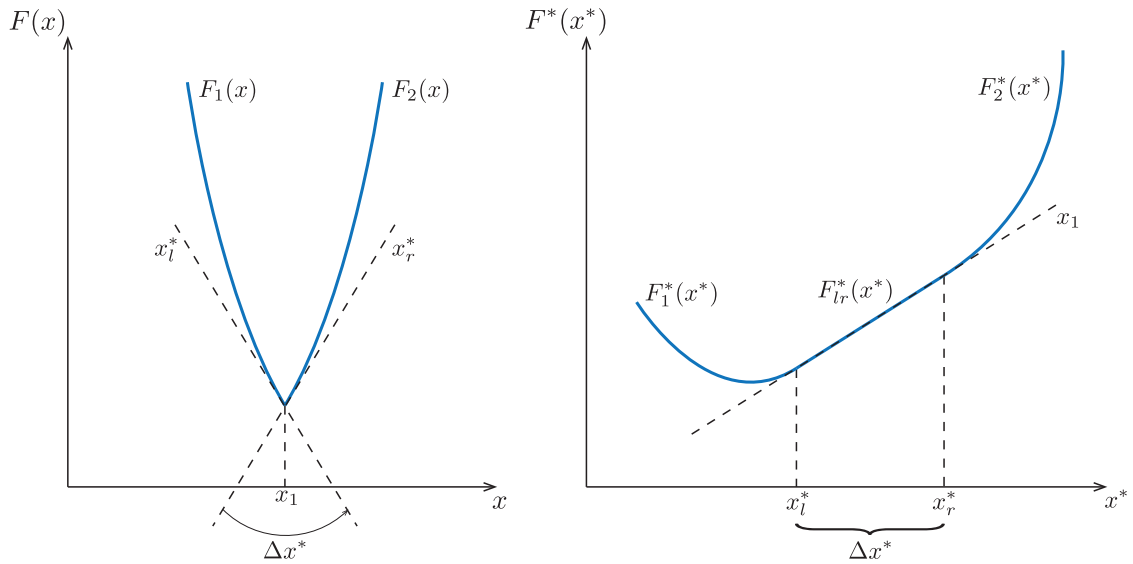
$$F(x) + F^*(x^*) \geq \langle x, x^* \rangle \tag{3.41}$$

$F^{**}(x^{**}) = F(x)$ if $F(x)$ is a convex function. If $F(x)$ is non-convex then the biconjugate forms the largest convex envelope of $F(x)$ satisfying $F^{**}(x^{**}) \leq F(x)$.

In the following we give some additional illustrative examples of conjugate functions for some relevant cases:

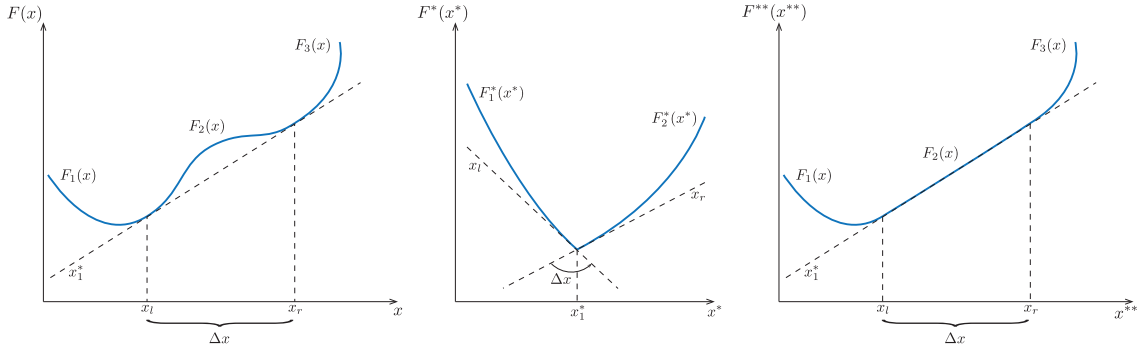
- Figure 3.8 shows the conjugate transform for an affine function with a non-differentiable point.
- Figure 3.9 shows the result of the conjugate transform applied to a non-convex function. It is nicely shown that the biconjugate $F^{**}(x)$ in Figure 3.9c forms the convex envelope of $F(x)$, Figure 3.9a.

For further details we once more refer to the textbooks [Rockafellar, 1997, Rockafellar and Wets, 1997, Boyd and Vandenberghe, 2004, Bertsekas et al., 2003].



(a) $F(x)$ **(b)** $F^*(x)$

Figure 3.8.: Conjugate transform of an affine function with a non-differentiable point x_1.



(a) $F(x)$ **(b)** $F^*(x)$ **(c)** $F^{**}(x)$

Figure 3.9.: Conjugate transform of a non-convex function.

3.6. Convex Optimization

Convex optimization is a special field in mathematical optimization and deals with the problem of finding the minimum of convex functions. Variational methods are based on continuous energy functionals and it is the intention to model this as a convex functional. In order to obtain its minimizer, an appropriate optimization method is applied. As convex optimization has been studied for more than a century, a vast amount of approaches and literature exist. For a more complete overview on convex optimization we refer to the textbooks from *e.g.* Rockafellar and Wets [1997], Rockafellar [1997], Bertsekas et al. [2003], Nesterov [2004a], and Boyd and Vandenberghe [2004]. In the following we will mainly

concentrate on some relevant approaches in imaging science that we are using to solve the arising optimization problems.

Consider an optimization problem of the form

$$\min F(v), \qquad F : \mathbb{R}^k \to \mathbb{R}\,, \tag{3.42}$$

with F being a convex functional and $F \in \mathcal{C}^1$, assuring that F is differentiable and its gradient ∇F is Lipschitz continuous. Let $v = (v_1, v_2, \ldots, v_k)$ be the optimization variables and assume that an optimum $\check{v}$ with $F(\check{v}) \leq F(v), \forall v$ exists. Furthermore, the function's gradient is defined as

$$\nabla F = \left(\frac{\partial F}{\partial v_1}, \ldots, \frac{\partial F}{\partial v_k} \right). \tag{3.43}$$

Due to F's convexity and $F \in \mathcal{C}^1$, a vanishing gradient $\nabla F\,(\check{v}) = 0$ is a necessary and sufficient condition for the optimum. This so-called optimality condition may be used to directly compute the solution of the minimization problem (3.42) but unfortunately for most imaging problems this is not possible. Hence, often an iterative process is utilized to compute the desired solution.

3.6.1. Gradient Descent Methods

As we restrict our models to convex functionals F, the necessary condition $\nabla F(v) = 0$ holds for any $v \in \mathbb{R}^k$ and is referred to as optimality condition. Unfortunately, it is rarely the case that a direct solution of can be computed for imaging problems. Therefore, the update direction for the gradient descent method is defined through the gradient's direction yielding

$$v^{n+1} = v^n - \tau \nabla F(v^n) \tag{3.44}$$

In (3.44) τ is the step size of the gradient update and has to be chosen properly so that the update scheme remains stable and does not overshoot. τ can be interpreted as a temporal discretization. In order to obtain a reasonable solution one has to choose a proper v^0 in such a way that the iterates (3.44) converge towards the desired stationary point.

When imposing additional constraints the projected gradient descent algorithm can be used to solve the respective optimization problem. Therefore, an additional projection onto the given set of constraints is added. Note that the requirement for applying this algorithm to constrained optimization problems is the ability to compute the projection in a reasonable fashion.

3.6.2. Interior Point Methods

The interior point methods, as the name suggests, converges to the solution through the interior of the convex polytope. Therefore, barrier functions are imposed to model the inequality constraints. In order to iterate towards the optimum, Newton iterations are utilized. Hence, the objective function $F\,(x)$ must be twice differentiable. Interior point

methods are state of the art in convex optimization and are able to achieve highly accurate results with fast convergence. Unfortunately, most of the problems in image processing have lots of unknowns and interior point methods are not suited to solve large scale optimization problems.

3.6.3. Proximal Point Methods

The key concept of proximal point methods as proposed by Rockafellar [1976] is to add equality constraints as quadratic terms to a convex function $F(x)$ in order to restrict the update to a certain bound. The proximal point method belongs to the family of simplex methods and is well suited to solve linear programming problems. Those methods test adjacent edges of the convex polytope to iteratively approach the optimum. The general task is to find a v such that $T(v) \ni 0$, with T being a maximal monotone operator. In order to seek the optimal v, T generates the sequence $\{v^n\}$ with

$$T\left(v^{n+1}\right) + \frac{1}{\lambda_n}\left(v^{n+1} - v^n\right) . \tag{3.45}$$

The proximal point algorithm can also be written with the help of the resolvent of T as

$$v^{n+1} = \left(I + \lambda_n T\right)^{-1}\left(v^n\right) . \tag{3.46}$$

For the case when T is defined as the object function's subgradient $T = \partial F(v)$ and F is again a convex function, the algorithm can be written as

$$v^{n+1} = \arg\max_v F(v) + \frac{1}{2\lambda_n}\left|\left|v - v^n\right|\right|_2^2 \tag{3.47}$$

The major drawback of this algorithm is the difficulty to compute the resolvent operator. For complex problems, computing the inverse of $I + \lambda_n T$ is often of the same complexity. Therefore, different approaches have been introduced to simplify this step. The operator T is split into $T = A + B$ with A and B being two maximally monotone operators. This enables the simpler computation of the resolvent operators $(I + \lambda_n A)$ and $(I + \lambda_n B)$ instead of the more complex $(I + \lambda_n A + B)$. Famous variants of this algorithm are the Douglas-Rachford splitting [Lions and Mercier, 1979] or alternating direction method of multipliers (ADMM), a variaint of the Augmented Lagrangian method [Bertsekas, 1982]. The latter one can be interpreted as an application of the Douglas-Rachford splitting algorithm which is in turn an application of the proximal point method [Eckstein and Bertsekas, 1992]. The disadvantage of proximal point methods compared to interior point methods is the worse accuracy. Still for the optimization problems that are discussed throughout this thesis, solving interior point methods is not an option due to the large amounts of unknowns. The trade-off in accuracy is not a problem either as most of the algorithms are an approximation by itself and therefore the accuracy of the proximal point approaches is sufficient. In general, proximal point methods are well suited to optimize

problems of the following form

$$\min_{v \in X} F(Kv) + G(v)\,, \tag{3.48}$$

with K being a linear operator with the mapping $K : X \to Y$ and F, G are convex functions. Most of the discussed problems will fit this scheme. In image processing, primal-dual splitting approaches have shown increased popularity and exhibit good performance for optimizing such problems. In the following we will review a variant of a primal-dual method proposed by Chambolle and Pock [2010].

3.6.4. Primal-Dual Approaches

We have already shown in Section 3.6.1 that setting the optimality condition to zero is sufficient in case of convexity. Now consider that the premise of a continuously differentiable function is not given. A simple, yet often used, function is the absolute function $|v|$ (as *e.g.* in the total variation) where the derivative degenerates at $v \to 0$. Concerning the total variation, Chambolle [2004] proposed a projection algorithm for minimizing the exact ROF energy functional (3.24) utilizing the dual formulation of the total variation. Again in the context of ROF denoising, Chan et al. [1996] were among the first that applied a primal-dual algorithm to solve the optimization problem. Subsequently, Pock et al. [2009] utilized a similar idea to optimize the Mumford-Shah functional. Generalizations and connections to other algorithms in the field have then been proposed by Esser et al. [2010] and Chambolle and Pock [2010]. In the latter, the algorithms convergence rate was shown to be optimal in the sense of Nemirovski [2004] and Nesterov [2004b]. We will refer to the work of Chambolle and Pock [2010] on this topic. In the following we reflect parts of their algorithm that are relevant for our upcoming considerations.

In general, first order primal-dual approaches have been popularized for solving problems of the form

$$\min_{v \in X} F(Kv) + G(v) \tag{3.49}$$

by transferring it to a primal-dual saddle-point problem yielding

$$\min_{v \in X} \max_{q \in Y} \langle Kv, q \rangle + G(v) - F^*(q)\,. \tag{3.50}$$

The respective dual problem is given as

$$\max_{q \in Y} -\left(F^*(q) + G^*\left(-K^* q\right)\right)\,. \tag{3.51}$$

We refer to v as the primal variable and q as its dual. The optimization problem depicts a minimization towards the primal function and a respective maximization on the dual formulation which yields a convex-concave saddle-point problem. In (3.50), K is a linear operator and the functions $G : X \to \mathbb{R} \cup \{\infty\}$ and $F^* : Y \to \mathbb{R} \cup \{\infty\}$ are convex functions.

Furthermore, F^* depicts the convex conjugate of the convex function F. For details on the Fenchel duality we refer to Section 3.5.3. When solving (3.50) with respect to minimizing the primal variable and maximizing the dual variable one ends up with an iterative algorithm that takes gradient ascent steps in the dual variable and gradient descent updates in the primal variable. Moreover, Chambolle [2004] added an additional extragradient step and showed convergence for $\theta = 1$. The algorithm then reads

$$\begin{aligned} q^{n+1} &= (I + \sigma\, \partial F^*)^{-1} \left(q^n + \sigma K \overline{v}^n\right) \\ v^{n+1} &= (I + \tau\, \partial G)^{-1} \left(v^n - \tau K^* q^{n+1}\right) \\ \overline{v}^{n+1} &= v^{n+1} + \theta \left(v^{n+1} - v^n\right) . \end{aligned} \tag{3.52}$$

Here, τ and σ are the step sizes of the primal and the dual update. They are chosen as $\tau > 0$ and $\sigma > 0$ satisfying $\tau\sigma L^2 \leq 1$, with $L^2 = ||K||^2$. One of the algorithm's restrictions is the structure of the operator K. It must have a simple structure so that its norm $L = ||K||$ is computable in a reasonable fashion.

In (3.52), $(I + \sigma\, \partial F^*)^{-1}$ and $(I + \tau\, \partial G)^{-1}$ are the so-called resolvent (or proximity) operators of the functions $G(v)$ and $F^*(q)$. The resolvent operator of a function $G(v)$ is defined as the minimizer of the following problem, given a $\hat{x} \in X$:

$$v^{n+1} = \arg\min_v \delta G(v) + \frac{1}{2} ||v - \hat{v}||_2^2 \tag{3.53}$$

With $\delta > 0$ the unique solution is given as

$$\delta\, \partial G(v) + (v - \hat{v}) \ni 0 \ . \tag{3.54}$$

Utilizing the operator notation, the optimal solution v^{n+1} can be computed via the resolvent operator of $G(v)$ which yields

$$v^{n+1} = (I + \delta\, \partial G(v))^{-1} (\hat{v}) \tag{3.55}$$

For simplicity reasons we set $\delta = 1$ for all further investigations. A requirement of the algorithm in (3.52) to the resolvent operator is the computability of its solution in a closed form. To sum up, there are two major restrictions of this primal-dual algorithm on the optimization problem that are

1. The operator K must have a simple structure such that the operator norm of K is computable in reasonable time.

2. The resolvent operator must be computable in a closed form manner (or with a fast algorithm) so that a unique minimizer to (3.53) can be computed.

Subsequent to (3.52), the authors proposed a preconditioned variant of the algorithm in [Pock and Chambolle, 2011]. There, they assured convergence properties for optimization problems, where the first restrictions of an easy to compute operator norm $||K||$ is not met.

3.6.5. Preconditioned Primal-Dual

[TODO: motivate the need for this in the thesis]

To overcome some of the models restriction of (3.52), Pock and Chambolle [2011] proposed a diagonal preconditioned primal-dual method. In case of a badly scaled linear operator K the convergence of the primal-dual algorithm (3.52) is bad. To assure convergence rates without the need to explicitly compute the step sizes τ and σ the positive and real-valued τ and σ of (3.52) are replaced with preconditioned matrices T and Σ. Therefore, the algorithm changes to

$$\begin{aligned} q^{n+1} &= (I + \Sigma\, \partial F^*)^{-1} \left(q^n + \Sigma K \overline{v}^n\right) \\ v^{n+1} &= (I + T\, \partial G)^{-1} \left(v^n - T K^* q^{n+1}\right) \\ \overline{v}^{n+1} &= v^{n+1} + \theta \left(v^{n+1} - v^n\right) . \end{aligned} \tag{3.56}$$

For the algorithmic proofs of converges we refer to the paper by Pock and Chambolle [2011]. If $\theta = 1$ the preconditioned matrices T and Σ must be symmetric, positive definite, and, similar to the step sizes, satisfy the condition

$$\left|\left|\Sigma^{\frac{1}{2}} K T^{\frac{1}{2}}\right|\right|_2^2 \leq 1 . \tag{3.57}$$

In order to improve convergence over (3.52) when $||K||$ is not simple to compute, the parametrization of the preconditioned algorithm is essential. In addition, the requirement of simple to compute resolvent operators remain. Here, Pock and Chambolle [2011] note that if G and F^* are separable in v_j and q_i and in addition if Σ and T are restricted to be diagonal matrices $\Sigma = \mathrm{diag}\,(\sigma_i)$ and $T = \mathrm{diag}\,(\tau_j)$, then the resolvent operators remain tractable to compute. Moreover, they showed convergence when choosing those diagonal elements of the preconditioner matrices as

$$\begin{aligned} \sigma_i &= \frac{1}{\sum_{j=1}^{N} |K_{i,j}|^{\alpha}} , \\ \tau_j &= \frac{1}{\sum_{i=1}^{M} |K_{i,j}|^{2-\alpha}} , \end{aligned} \qquad \text{with} \quad \alpha \in [0, 2] . \tag{3.58}$$

CHAPTER
FOUR

OPTICAL FLOW

When dealing with movies or other image sequences where some motion is involved, the knowledge of the pixels' movement is a rich source of information. In the following, we give an introduction on optical flow and its connection to motion estimation. In order to estimate optical flow vectors a variational formulation is modeled and we also introduce some common formulations. The introduced TV-ℓ_1 model selects the origin for all our further investigations in the field of optical flow estimation. Furthermore, we give details on the utilized optimization framework to solve this model.

Contents

4.1. Motion Perception

Motion perception describes the process of motion cognition in a scene. It infers the velocity and orientation of the object's motion. From a biological point of view it depicts an important field in neuroscience as well as psychology. Because of its complexity, it would be far beyond the scope of this thesis to discuss details of biological motion. The more important aspect for us is how to reconstruct the motion of elements in temporally succeeding images. For details on biological motion perception and especially how to connect those to computer science we refer to the work of Longuet-Higgins and Prazdny [1980], and Adelson and Bergen [1985] and references therein.

Concerning the estimated motion, we are mainly interested in the apparent motion of points in a sequence of consecutive images. The apparent motion is the illusion of motion when viewing sequenced images in a certain time frame. When *e.g.* watching a movie the human observer gets the impression of motion by looking at a rapid succession of still images where actually each element is pictured at a distinct location. The illusion of motion is not given until the temporal information is available. This shows that the combination of spatial and temporal information is important when considering motion estimation. A very basic concept of motion perception is to consider illumination changes of a point at the retina. In addition, this point is connected to the neighboring points such that illumination changes can be detected (*cf.* Hassenstein-Reichardt detectors). These changes of the brightness pattern over time are also denoted as optical flow.

The aim of the following chapter is to give an overview on optical flow and the most relevant historical findings in this field. Moreover, we reflect an approach of modeling an energy functional in a variational setting to compute the disparities between points in a pair of consecutive images. In addition, we introduce the optimization framework and give details on how to solve the given energy functional.

4.2. Optical Flow

Optical flow (sometimes also optic flow) describes the apparent motion of an image brightness pattern. It denotes a 2D disparity vector linking points of two consecutive frames together. Hence, it is only the 2D projection of some 3D motion. Let us point out that the optical flow and the 2D motion field can only be equated under certain circumstances. Although a lot of literature, among them Horn [1986], and Verri and Poggio [1989], already referred to this issue, large amounts of literature have used the term optical flow misleadingly when the true motion field is meant and vice versa. In order to emphasize this issue once more, assume a static scene with no object performing any type of motion, but with changing illumination conditions. Therefore, the image intensities will change at certain regions due to *e.g.* moving shadows, highlights and reflections. This causes a change between the brightness pattern of consecutive frames which is estimated as optical flow although no object motion is present (see Figure 4.1). Although the assumption of

constant intensities over time is incorrect for many situations it still does provide reasonable results and is heavily used. To simplify the terminology we stick to the equation of motion vectors and optical flow and account for the differences when we talk about approaches that are (or are not) robust to *e.g.* illumination changes. The versatile applicability of optical flow vectors is confirmed when looking at the variety of fields where optical flow is used like *e.g.* tracking, driver assistance systems, video compression, video processing, motion segmentation, structure from motion, superresolution, medical image registration, and many more.

(a) **(b)**

Figure 4.1.: Changing illumination causes an optical flow field although no object's motion is performed.

4.2.1. The Optical Flow Constraint

Consider an image sequence as space/time volume $I(x(t), t)$, where $x = (x^1, x^2)^T$ denotes the spatial position, t the time and $I(x, t) : \Omega \times T \to \mathbb{R}$ the intensity function. In the following we rely on the frequently made assumption (*e.g.* [Lucas and Kanade, 1981, Horn and Schunck, 1981]) of a constant intensity function over time (*brightness constancy* or *data conservation* assumption), that is

$$\frac{d}{dt} I(x(t), t) = 0\,. \tag{4.1}$$

Applying a Taylor expansion to the above equation, neglecting the second- and higher-order terms, yields

$$\frac{d}{dt} I(x(t), t) = \frac{\partial I(x(t), t)}{\partial x^1} \frac{dx^1}{dt} + \frac{\partial I(x(t), t)}{\partial x^2} \frac{dx^2}{dt} + \frac{\partial I(x(t), t)}{\partial t} = 0\,. \tag{4.2}$$

Furthermore, we define the motion vector as

$$u \in Y: \quad u\,(x) = \left(u^1, u^2\right)\,, \tag{4.3}$$

and the incremental motion vector

$$\mathrm{d}u(x) = u(x) - u_0(x) = (\frac{\mathrm{d}x^1}{\mathrm{d}t}, \frac{\mathrm{d}x^2}{\mathrm{d}t})^T\,, \tag{4.4}$$

the spatial image gradient

$$\nabla I(x(t),t) = \left(\frac{\partial I(x(t),t)}{\partial x^1}, \frac{\partial I(x(t),t)}{\partial x^2} \right)^T \tag{4.5}$$

and the temporal image derivative

$$I_t(x(t),t) = \frac{\partial I(x(t),t)}{\partial t}\,. \tag{4.6}$$

Substituting (4.4), (4.5) and (4.6) into (4.2) yields the classical optical flow constraint (OFC):

$$\rho(u(x)) = (\nabla I(x(t),t))^T(u(x) - u_0(x)) + I_t(x(t),t) = 0\,, \tag{4.7}$$

with u_0 being a given flow field. Note, that this equation is ill-posed for recovering both components of the flow field u which is related to the aperture problem:

The *aperture problem* denotes the ambiguity in motion perception when moving (mostly repetitive) structures are observed through an aperture. Assume a moving repetitive texture and further assume that the borders of the object are not visible as seen through an aperture. From the resulting pattern, visible in the aperture, the direction of the motion is ambiguous as shown in Figure 4.2. Furthermore, this ambiguity can result from untextured regions. When perceiving the motion of homogeneous areas the observer cannot tell in which direction the motion vectors are pointing until the object's boundaries (or at least a bigger neighborhood) is taken into account.

Different approaches have been proposed to make (4.7) well-posed which can essentially be divided into two classes, namely local and global approaches. In the following we give a brief overview of these concepts and their historically most seminal approaches.

4.2.2. Local Approach

Lucas and Kanade [1981] proposed a local approach that estimates the flow vectors within a small spatial neighborhood. The basic optical flow constraint is solved with the assumption that the entire proximity is displaced by a similar vector. This yields an over-determined system of equations, which can be solved with a least-squares estimator, minimizing the squared errors

$$E_{\mathrm{LK}}(u(x)) = \sum_{\overline{x}\in\mathcal{N}(x)} g(\overline{x},x)\left(I_t(\overline{x},t) + (\nabla I(\overline{x},t))^T u(x)\right)^2\,. \tag{4.8}$$

Here, the weighting function $g(\overline{x},x)$ steers the support between pixels within the neighborhood $\mathcal{N}(x)$ around x and usually decreases with increasing distance $|\overline{x}-x|$. Since (4.8) is

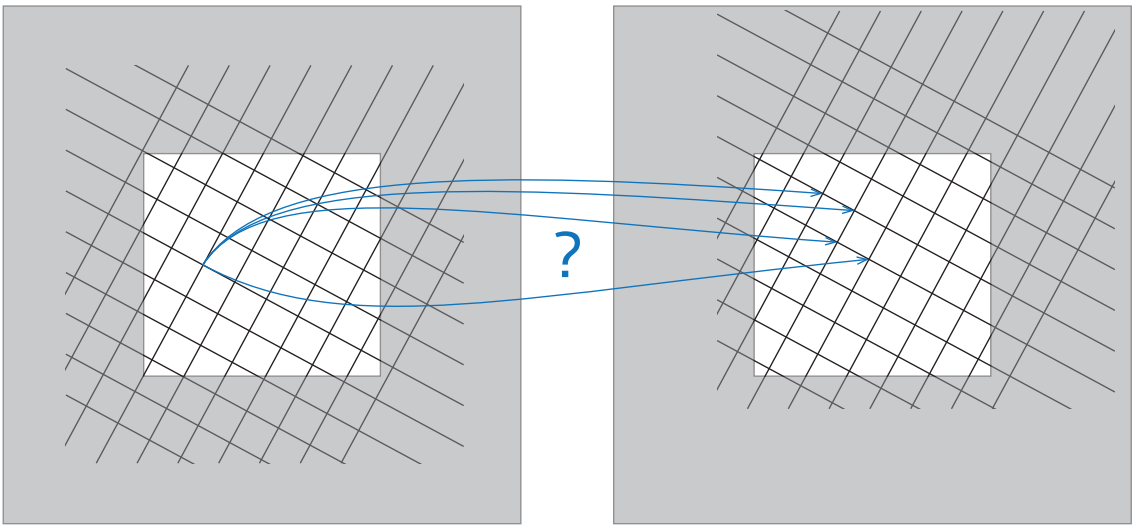


Figure 4.2.: Aperture problem: the ambiguity of motion.

convex, the global minimum of $E_{\mathrm{LK}}(u(x))$ is reached, when the derivatives with respect to the components of $u = (u^1, u^2)^T$ are zero:

$$\frac{\partial E_{\mathrm{LK}}(u)}{\partial u^1} = \sum g \left(I_x I_t + u^1 I_x^2 + u^2 I_x I_y \right) = 0 \,, \tag{4.9a}$$

$$\frac{\partial E_{\mathrm{LK}}(u)}{\partial u^2} = \sum g \left(I_y I_t + u^1 I_x I_y + u^2 I_y^2 \right) = 0 \,, \tag{4.9b}$$

which are often written in matrix form

$$Mu = b \,, \tag{4.10}$$

with

$$M = \begin{pmatrix} \sum g I_x^2 & \sum g I_x I_y \\ \sum g I_x I_y & \sum g I_y^2 \end{pmatrix} \,, \qquad b = \begin{pmatrix} \sum g I_x I_t \\ \sum g I_y I_t \end{pmatrix} \,. \tag{4.11}$$

When $\mathrm{rank}(M) = 2$ is satisfied, the image structure in the local neighborhood $\mathcal{N}(x)$ contains enough information to solve the aperture problem and the least-squares estimate can be calculated as $u = M^{-1}b$. To assure the ability to compute a unique solution, the neighborhood size is increased which in turn presumes the same displacement vector for this larger proximity. A trivial way to overcome this dilemma is to compute the optical flow only where a unique solution can be estimated. This yields a sparse solution of the resulting optical flow field. Although this approach is an entirely local approach, the method is still widely used in computer vision systems due to its simplicity. Clearly, the assumption of constant velocity within a patch is generally not valid for realistic data.

Therefore, various modifications of this approach were proposed in subsequent works. As the Lucas & Kanade algorithm is only suitable to estimate small displacements (as it is

based on the linearized OFC), it is common practice to use a coarse-to-fine approach to recover larger displacements [Moravec, 1979, Lucas and Kanade, 1981]. Robustness, on the other hand, can be increased by using more robust norms instead of the ℓ_2 penalty in (4.8) [Shulman and Hervé, 1989, Black and Anandan, 1991, Black, 1992], adapting the window size for defining the neighborhood $\mathcal{N}(x)$ [Black and Jepson, 1996] or explicitly model the neighborhood with *e.g.* the representation of moving layers [Wang and Adelson, 1994].

4.2.3. Global Approach

Regarding the desire of a dense optical flow field, the local approach shows some drawbacks when estimating optical flow vectors *e.g.* within a homogeneous region. Solving the resultant system of linear equations do not reveal an accurate solution in such areas or might even be rank deficient, hence cannot be solved uniquely. By contrast, the global approach uses a smoothness term that propagates the optical flow field from well-posed regions to regions with poorly conditioned data fidelity. As the name suggests, the global method utilizes the complete available data to estimate the optical flow field and yields a dense result.

With the seminal work of Horn and Schunck [1981], a new chapter in estimating optical flow was opened. The idea to circumvent the ill-posedness of the OFC is to formulate the optical flow problem as an optimization problem of the form

$$\min_{u(x)} \left\{ \underbrace{\int_\Omega |\nabla u(x)|^2 \, \mathrm{d}x}_{\text{regularizer } (\mathcal{R}(u))} + \lambda \underbrace{\int_\Omega |\rho(u(x))|^2 \, \mathrm{d}x}_{\text{data term } (\mathcal{D}(u))} \right\}, \tag{4.12}$$

where $\rho(u\,(x))$ denotes the OFC and the free parameter λ defines the trade-off between smoothness (regularization) and data fidelity (data term). As already stated, the major advantage of such global methods is the ability to handle regions where local approaches would end up with poorly conditioned systems. As for global methods, the regularization term propagates the optical flow into such regions by filling in from neighboring estimates. This enables approaches like (4.12) to gain a dense flow field. Mainly the computational effort to optimize energy equations like (4.12) has limited the use of global approaches in applications. With the introduction of more sophisticated optimization algorithms and possibilities to parallelize the algorithms on *e.g.* the GPU, the global approaches have seen a renewed interest. When it comes to accuracy, the results on the Middlebury benchmark [Baker et al., 2011] suggest that global variational methods are among the best performing algorithms available. Since the introduction of such variational approaches in [Horn and Schunck, 1981] a vast amount of literature and extensions have been published and we will give some insights on how to model such energy functionals in the following:

4.3. Modeling of Variational Optical Flow

The formulation of optical flow estimation as an energy minimization problem gives the flexibility to incorporate different regularization and data terms. Different approaches have already been reviewed in the surveys by Weickert et al. [2006] and Trobin [2009]. Driven by this flexibility various different approaches have been proposed since the original formulation (4.12) by Horn and Schunck [1981] (denoted as Horn & Schunck model in the following). Let us first recall the properties of the Horn & Schunck model: (4.12) combines a quadratic penalization on the classical optical flow constraint (OFC) (see Section 4.2.1) for modeling the data fidelity and again a quadratic penalization of the flow gradients enforcing a smooth flow field. The trend towards over-smoothed results caused by the quadratic penalty, not allowing for any outliers, shows room for improvement. In addition, a data term based on the OFC is prone to errors if outliers occur in one of the images or simply when the illumination changes over time. In the following we stick to the distinction on changing the regularization or data term and review some popular variants for modifying (4.12).

4.3.1. Regularizer

In order to overcome the aperture problem (see Section 4.2.1) it became common practice to add a smoothness constraint. Over the years, different penalty functions have been used to improve accuracy and robustness of the methods. A motivation for the regularization term from a stochastic point of view is given in [Trobin, 2009] and for the link to diffusion-reaction systems we refer the interested reader to [Weickert and Schnörr, 2001], and [Weickert et al., 2006]. In the following we will outline some popular concepts of regularization terms used for optical flow estimation.

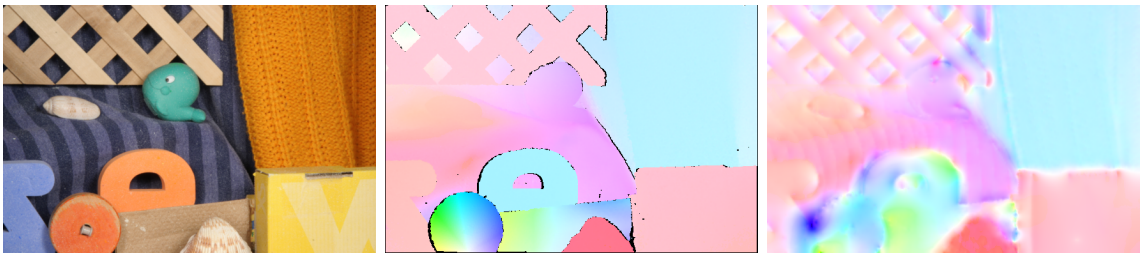
We divide the taxonomy of regularizers according to *e.g.* [Weickert et al., 2006], and [Trobin, 2009] into flow-driven and image-driven regularization. When the image function is used to modify the influence of the smoothing step, the approaches are called image-driven, whereas an adaption based on the flow field itself is called flow-driven. The fact that image features can easily be computed supports the image-driven regularization. However, when image features occur in areas where the flow field should be homogeneous the smoothness can be distracted. In such a case, the flow-driven type would be preferable.

Homogeneous regularization is a quadratic regularization as proposed by Tikhonov [1963] enforcing the optical flow field to be smooth in a homogeneous way by penalizing the gradients of the flow field using the regularizer

$$\mathcal{R}(u) = \int_{\Omega} |\nabla u|^2 \, \mathrm{d}x \,. \tag{4.13}$$

It represents the smoothness prior used in the original paper by Horn and Schunck [1981]. The drawback of this homogeneous prior assumption is reflected by blurry optical flow

fields caused by the well-known over-smoothing effects of (4.13) (see Figure 4.3). The reason is the non-robustness towards outliers and consequently the model does not allow for discontinuities in the flow field.



(a) Frame 10 **(b)** Ground truth **(c)** Horn & Schunck

Figure 4.3.: Optical flow field computed with the Horn & Schunck model [Horn and Schunck, 1981]. Input: Frame 10 and 11 from Middlebury's `RubberWhale` sequence.

Robust regularization techniques have become popular for mainly two reasons, the robustness towards outliers, and the ability to preserve motion discontinuities. In comparison to the quadratic regularization of the Horn & Schunck model, a robust penalization of flow gradients allows for outliers and to increases the robustness towards the presence of such. In terms of optical flow estimation, at first Shulman and Hervé [1989] and later Black and Anandan [1991], Black [1992], Black and Anandan [1993, 1996] used penalizers from robust statistic to improve the performance of (4.12) in the presence of outliers. Subsequently, Memin and Perez [1996] proposed an approach to compute a flow field in a variational setting as in (4.12) but with the additional constraints to prohibit smoothing across motion discontinuities and increase robustness towards outliers as well. Inspired by the work of Rudin et al. [1992] in the field of image restoration, a regularization like

$$\mathcal{R}(u) = \int_{\Omega} |Du| \tag{4.14}$$

was introduced to overcome the ambiguity of the OFC [Shulman and Hervé, 1989, Cohen, 1993, Brox et al., 2004]. The total variation regularization has become a widely used prior term to enforce coherence in the optical flow field. Often the approximation $\mathcal{R}(u) = \int_{\Omega} \sqrt{\varepsilon^2 + |\nabla u|^2}\, \mathrm{d}x$ with $\varepsilon > 0$ has been used to overcome the nondifferentiability at zero [Papenberg et al., 2006]. Zach et al. [2007] used a variant of Chambolle's algorithm [Chambolle, 2004] to solve for an optical flow model utilizing the exact total variation as smoothness prior.

4.3.1.1. Flow-driven Regularization

Isotropic flow-driven regularization reduces the smoothing of motion boundaries by decreasing the regularizer's influence nearby the edges of the flow field. This flow-driven regularization was introduced by Schnörr [1994] using the smoothness assumption

$$\mathcal{R}(u) = \int_\Omega \psi\left(|\nabla u\,(x)|^2\right)\,\mathrm{d}x\,, \tag{4.15}$$

where $\psi\,(\cdot)$ denotes a (differentiable) function that adapts the influence of the regularization towards the flow field.

Anisotropic flow-driven regularization was proposed by Weickert and Schnörr [2001] using

$$\mathcal{R}(u) = \int_\Omega \mathrm{tr}\left(\psi\left(\nabla u\,(x)\,(\nabla u\,(x))^T\right)\right)\,\mathrm{d}x\,. \tag{4.16}$$

The operator $\psi\,(\cdot)$ splits its argument into its Eigenvectros and uses such to encode the strength as well as the direction of the edges in the flow field. Then, $\psi\,(\cdot)$ adapts the regularization towards the edges and therefore guides the regularization process according to them. The resultant smoothing process is prohibited across flow edges and enforced along them.

4.3.1.2. Image-driven Regularization

Isotropic image-driven regularization is based on the assumption that motion boundaries coincide with image gradients [Nagel, 1983, Alvarez et al., 1999]. The modification reads

$$\mathcal{R}(u) = \int_\Omega g\left(|\nabla I\,(x)|^2\right)|\nabla u\,(x)|^2\,\mathrm{d}x\,, \tag{4.17}$$

where g depicts a strictly positive weighting function dependent on the image gradients $(\nabla I\,(x))$ so that the influence of the regularizer is reduced when image gradients are present.

Anisotropic image-driven regularization was proposed in an early version by Nagel [1983]. In addition to the isotropic variant (4.17), directional information of the image gradients are incorporated into the quadratic regularization (4.13) yielding

$$\mathcal{R}(u) = \int_\Omega (\nabla u\,(x))^T\,D\,(\nabla I\,(x))\,(\nabla u\,(x))\,\mathrm{d}x\,. \tag{4.18}$$

Here, $D(\cdot)$ is a symmetric, positive definite diffusion matrix that impedes smoothing across image edges and allows smoothing along them. In contrast to the scalar weighting function in (4.17), only taking the gradient's magnitude into account, the direction of the image edges is incorporated to weight the influence of the regularization.

4.3.2. Data Term

The data term, as the name suggests, searches for pixel correspondences in the input data. Usually, this is realized by penalizing the residual of some matching cost $\delta\left(I(x,t), u(x)\right)$. The penalty function ψ can be quadratic $\psi(\delta) = \delta^2$ as in (4.12) proposed by Horn and Schunck [1981]. In order to increase robustness towards outliers, Black and Anandan [1993] started to use robust penalizers as shown for the regularization term. A popular choice is the ℓ_1 norm, hence $\psi(\delta) = |\delta|$. In order to assign correspondences between pixels, an appropriate similarity measure δ must be found. A majority of optical flow models utilize the classical OFC (4.7) based on the brightness constancy constraint (4.1). However, for realistic image data, this assumption is often violated due to *e.g.* changing illumination, surface reflections or moving shadows. Therefore, methods that deal with such violations on the OFC have been introduced. Moreover, prior information can be incorporated in the matching energy. In a situation where *e.g.* the camera intrinsics are known, the geometric relations like camera movement or insights from the epipolar geometry can be added as an additional data term. Slesareva et al. [2005] incorporated the epipolar geometry to use optical flow methods for computing stereo depth maps. Then, Wedel et al. [2008] proposed to use a fundamental matrix prior in addition to the classical OFC. Valgaerts et al. [2008] also included an epipolar constraint as a second data term but unlike Wedel et al. [2008] it is not used as prior knowledge but to formulate a joint optimization problem in order to recover the optical flow and the fundamental matrix simultaneously.

In the following, we revisit some prominent examples on how to become invariant to gray-value changes in the intensity function. Despite all approaches to incorporate high level information, changing illumination conditions is among the most prominent situations. Especially when relying on the intensity function for computing correspondences in the data term, it is essential to make the approach robust towards changes in the brightness pattern.

Structure/Texture Decomposition: When solely relying on the image's intensity functions the approach demands for a preprocessing step to reduce the influence of brightness variations over time. Based on Aujol et al. [2006], Wedel et al. [2009] proposed to decompose the images into its "structure" and its "texture" part and reassemble the input images as a linear combination of both with emphasis on the "texture" part. This improves the robustness towards illumination changes as those are encoded into the structure part. Wedel et al. [2009] used the ROF denoising model (3.24) to obtain this splitting. A major drawback of this approach is the potential loss of useful information. Especially when no brightness variations occur, the model still neglects potentially useful information of the structure part. However, the decomposition can be computed quickly and is therefore well-suited in applications where a certain robustness towards illumination changes is desired but the computation time is limited.

Higher Order Constancy Assumption: Already in the 1980s, Tretiak and Pastor [1984], and Uras et al. [1988] suggested to use higher order terms to model the constancy assumption for local approaches. Later, Tistarelli [1996] used multiple data terms including gradient constancy in a local approach. For computing dense flow fields, Brox et al. [2004] proposed to impose gradient constancy as a data term for a global approach. Hence, the additional gradient constancy assumption changes (4.1) to

$$\frac{d}{dt}I(x(t),t) + \alpha\frac{d}{dt}\nabla I(x(t),t) = 0\,. \tag{4.19}$$

Here, $\nabla I(x(t),t)$ denotes the spatial gradient and α weights the two constancy assumptions. Using the gradient constancy assumption is invariant to additive illumination changes but unfortunately not to multiplicative ones. In addition, the sensitivity towards noise is increased. Recently, Zimmer et al. [2009] combined the use of gradient constancy with brightness constancy in the HSV color space. By combining a robust penalization of the data fidelity term and an anisotropic smoothness prior the accuracy of results were improved to previous models.

Moreover, higher order constancy assumptions, as *e.g.* constancy of the Hessian, can be incorporated. In Table 4.1 we review different variants of constancy assumption in the taxonomy of Papenberg et al. [2006]:

Table 4.1.: Variants of the constancy assumption (4.1) according to Papenberg et al. [2006].

constancy assumption	intensity function
brightness	$\frac{d}{dt}I(x(t),t) = 0$
gradient	$\frac{d}{dt}\nabla I(x(t),t) = 0$
Hessian	$\frac{d}{dt}\mathcal{H}(I(x(t),t)) = 0$
Laplacian	$\frac{d}{dt}\Delta I(x(t),t) = 0$
norm of the gradient	$\frac{d}{dt}\|\|\nabla I(x(t),t)\|\| = 0$
norm of the Hessian	$\frac{d}{dt}\|\|\mathcal{H}(I(x(t),t))\|\| = 0$
determinant of the Hessian	$\frac{d}{dt}\det\mathcal{H}(I(x(t),t)) = 0$

Modified Optical Flow Constraint: As the higher-order constancy assumptions of Table 4.1 are not invariant to multiplicative illumination changes but realistic illumination changes mostly imply such, extensive research has been performed in the field of modeling lightning changes as part of the OFC. Already Horn [1986], Nagel [1989], Verri and Poggio [1989], and Shulman and Hervé [1989] showed, that the assumption of brightness constancy generally does not hold due to variations in the gray value function over time. Therefore, Shulman and Hervé [1989], and in subsequent works Gennert and Negahdaripour [1987], Negahdaripour [1998], and Kim et al. [2005], impose additional functions into the OFC to model additive and multiplicative illumination changes. In order to guarantee the

smoothness of those, additional regularizers are imposed. Hence, the OFC reads

$$(\nabla I(x(t),t))^T(u(x)-u_0(x))+I_t(x(t),t)-mI(x(t),t)-c=0\,, \tag{4.20}$$

where m is the multiplicative and c the additive component.

Haussecker and Fleet [2001] used physical models of illumination conditions to allow for brightness variations in the OFC. For two input images only linear changes in the illumination can be modeled. If more images are available, also more complex models can be considered. Hence, a parametrized function $G(I(x(t),t,a)$ that is able to model (nonlinear) brightness changes over time is incorporated into the OFC yielding

$$(\nabla I(x,t))^T(u(x)-u_0(x))+I_t(x,t)=G(I(x(t)),t,a)\,, \tag{4.21}$$

with the n-dimensional parameter vector $a=[a_1,\ldots,a_n]$ modeling the brightness variations.

A different approach was proposed by Mileva et al. [2007] for global models and earlier by van de Weijer and Gevers [2004] for local approaches, who used a multichannel approach for photometric invariants. Therefore, the brightness constancy was redefined on multichannel data. The robustness on brightness variations is achieved by introducing a color channel transformation. The color input images (*e.g.* RGB) are transformed to a color space where the channels are invariant to varying lightning conditions. This normalized color space is used to model multiple constancy assumptions (channel-wise) to form the data term as the weighted sum of the residuals similar to (4.19).

Patch-based Similarity Measures: In the field of image matching, correlation-based measures are well established to become invariant to multiplicative illumination changes. Steinbrücker et al. [2009b] proposed a flexible data term that allows to use patch-based similarity measures as data term for optical flow computation. In order to be able to use *e.g.* a normalized cross-correlation measurement, the problem of linearizing the data term must be considered. The authors show that using traditional Taylor expansions to linearize the intensity function is difficult when using patch-based measures. Therefore, the optimization strategy uses alternative optimization steps for the components of the optical flow field which yields two convex subproblems each with a single unknown [Steinbrücker et al., 2009a].

4.4. TV-ℓ_1 Optical Flow

Let us now define the starting point for further investigation of computing optical flow fields. Considering the historic evolution of possible frameworks, the demand on dense flow fields emphasized the decision to use a global approach within a variational framework. The choice is confirmed by the model's flexibility and the possibilities to incorporate various extensions. In addition, most of the top performing approaches in the Middlebury

evaluation database are formulated within the variational framework which affirm this choice. Considering the discussed modeling approaches from Section 4.3 we decide to use an ℓ_1 penalization for both, the regularization and the data term. This allows for outliers; in addition it preserves motion discontinuities. Concerning the regularization term, we use a standard total variation regularization of the flow field's gradients and for the data term we assume constant intensities along motion trajectories and use the classical linearized OFC as in (4.7). Combining the individual parts yields the optimization problem

$$\begin{aligned} &\min_{u\in Y} \int_\Omega |\nabla u\,(x)|\,\mathrm{d}x + \lambda \int_\Omega |\rho(u\,(x))|\,\mathrm{d}x\,, \\ \text{with}\quad &\rho(u(x)) = (\nabla I(x(t),t))^T(u(x)-u_0(x)) + I_t(x(t),t) = 0\,, \\ \text{and}\quad &||u-u_0|| \le \Delta_{u_0}\,. \end{aligned} \tag{4.22}$$

Here, $\rho(u\,(x))$ is the linearized optical flow constraint (OFC) (4.7) and λ defines the tradeoff between the data fidelity and the regularization. In order to guarantee that the linearization remains valid we restrict the increment of updates of the flow vectors in an interval of radius Δ_{u_0} around u_0. For a graphical illustration of the linearization at u_0 and the restricted update interval see Figure 4.4.

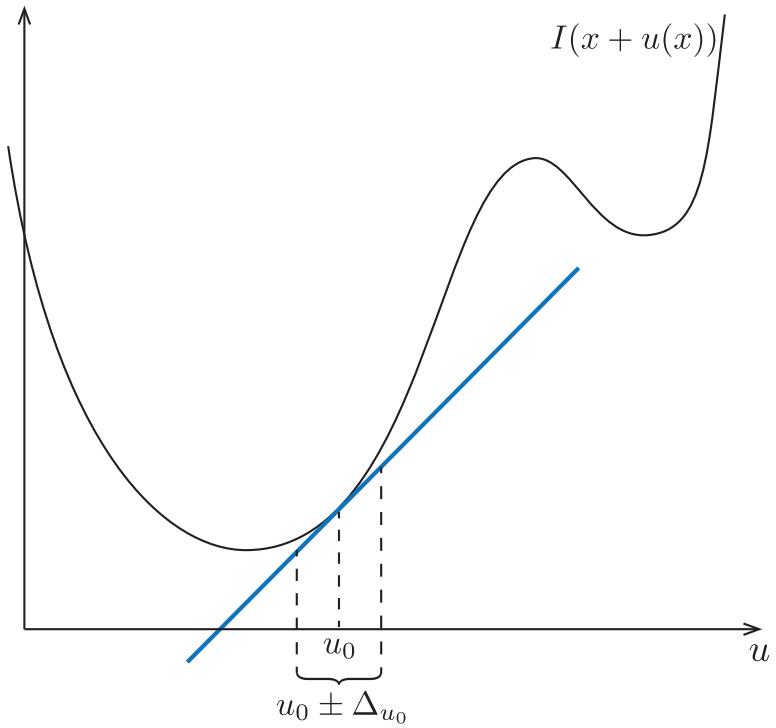


Figure 4.4.: Linearization of the image function. Restrict the update to an interval around u_0.

As pointed out in Chapter 3, the optimization of an energy functional like (4.22) using an ℓ_1-norm as penalizer function is not a trivial task. A clever splitting approach to optimize the TV-ℓ_1 model was proposed by Zach et al. [2007]. Recent advances in the field of convex approaches for imaging problems show especially for variational denoising models, that primal-dual approaches (see also Section 3.6.4) yield a good performance for optimizing such functionals. In the following, we will utilize the optimization framework proposed by Chambolle and Pock [2010] for solving the minimization problem in (4.22). But before that, let us first depict some details concerning the discretization of the energy functional (4.22).

4.4.1. Discretization

At first, we want to refer to Section 3.1 for the common overview of the used discretization approach. In the following, we will utilize parts of it to achieve a discretized variant of the TV-ℓ_1 optical flow model (4.22). Assume that we have two discrete images I_1 and I_2 both defined on the grid Ω of size $M \times N$. Let the images be a sequent pair out of an image sequence. In literature, I_2 is often denoted as the moving image and I_1 as the fixed or the reference image. In the following, we will denote discrete pixel locations in the images as $(I_i)_{i,j}$. Let us now define the discretized optical flow vector $u_{i,j} = \left((u^1)_{i,j}, (u^2)_{i,j}\right)^T \in Y$, also defined on a regular and equally spaced Cartesian grid. With that, we define the discretized variant of the TV-ℓ_1 optical flow model (4.22) as

$$\begin{aligned} &\min_{u \in Y} ||\nabla u||_{2,1} + \lambda \, ||\rho(u)||_1 \ , \\ &\quad \text{with} \quad \rho(u_{i,j}) = (I_t)_{i,j} + (\nabla I)_{i,j}^T \left(u_{i,j} - (u_0)_{i,j}\right) \ . \end{aligned} \tag{4.23}$$

4.4.2. Primal-Dual Optimization of the TV-ℓ_1 Optical Flow

For optimization we apply the primal-dual algorithm of Section 3.6.4 and therefore we have to convey (4.23) to a primal-dual saddle point problem like in (3.50). Hence, we apply the Legendre-Fenchel (LF) transform (3.39) to (4.23). Let us first introduce the dual variable of u denoted as $p = u^*, \ p \in Z$ and, in the discrete case, it is written as

$$p_{i,j} = \left(p^1_{i,j}, p^2_{i,j}, p^3_{i,j}, p^4_{i,j}\right) \ . \tag{4.24}$$

In the following we omit the indices (i, j) to simplify the notation. The convex conjugate of the total variation, defined as $F(u) = ||\nabla u||_{2,1}$, is

$$F^*(p) = \sup_{u \in Y} \left\{ \langle \nabla u, p \rangle - ||\nabla u||_{2,1} \right\} \ . \tag{4.25}$$

To find its supremum with respect to $u \in Y$ we distinguish the following cases:

- To maximize (4.25) for the case that $||p|| \leq 1$, $u = 0$ and $F^*(p) = 0$.
- If $|p| > 1$ the supremum is attained for $u \to \infty$.

Hence, two different cases are important to characterize the conjugate of the total variation, namely

$$F^*(p) = \begin{cases} 0 & |p| \leq 1 \\ \infty & \text{else} \end{cases} . \tag{4.26}$$

This can also be written with the indicator function

$$\delta_P(p) = \begin{cases} 0 & \text{if} \quad p \in P \\ \infty & \text{else} \end{cases} , \tag{4.27}$$

where $\delta_P(p)$ is the projection of p onto the convex set $P = \{p \in Z : ||p||_\infty \leq 1\}$. Returning to the generic saddle-point formulation (3.50) with

$$F(u) = ||\nabla u||_1 \tag{4.28}$$
$$G(u) = \lambda \, ||\rho(u)||_1 \tag{4.29}$$
$$F^*(p) = \delta_P(p) \tag{4.30}$$

the given problem (4.22) can be formulated as the primal-dual saddle-point problem

$$\min_{u \in Y} \max_{p \in Z} \langle \nabla u, p \rangle_Z + \lambda \, ||\rho(u)||_1 - \delta_P(p) \, . \tag{4.31}$$

This formulation is of the form (3.50), hence we can apply the primal-dual algorithm (3.52). Therefore, we have to compute the resolvent operators (or prox operators) $(I + \sigma \, \partial F^*)^{-1}$ and $(I + \tau \, \partial G)^{-1}$ with respect to $F^*(p)$ and $G(u)$. The resolvent operator in terms of $F^*(p)$, defined as $p = (I + \sigma \, \partial F^*)^{-1}(\hat{p})$, is given through the minimizer

$$p = \arg\min_p \left\{ \frac{1}{2\sigma} \, ||p - \hat{p}||_2^2 + F^*(p) \right\} = \arg\min_p \left\{ \frac{1}{2\sigma} \, ||p - \hat{p}||_2^2 + \delta_P(p) \right\} . \tag{4.32}$$

The minimizer of this optimization problem is

$$p_{i,j} = \text{prox}_P(\hat{p}_{i,j}) \, , \tag{4.33}$$

where prox_P denotes a pointwise projection of p onto the Euclidean unit ball. Similarly, the resolvent operator $u = (I + \tau \, \partial G)^{-1}(\hat{u})$ is defined as the minimizer of

$$u = \arg\min_u \left\{ \frac{1}{2\tau} \, ||u - \hat{u}||_2^2 + G(u) \right\} = \arg\min_u \left\{ \frac{1}{2\tau} \, ||u - \hat{u}||_2^2 + \lambda \, ||\rho(u)||_1 \right\} . \tag{4.34}$$

To simplify the following elaboration, we define the OFC as

$$\begin{aligned} \rho(u) &= a^T u + b \\ &\text{with} \quad a = \nabla I \quad \text{and} \quad b = I_t \end{aligned} \tag{4.35}$$

For computing the minimum of (4.34) the optimality condition yields

$$u - \hat{u} + \tau\lambda s = 0\,, \tag{4.36}$$

where s is the subgradient and is defined with the cases

$$s \in \partial \left| a^T u + b \right| = \begin{cases} a & \text{if} \quad a^T u + b > 0 \\ -a & \text{if} \quad a^T u + b < 0 \\ [-a, a] & \text{else} \end{cases} \tag{4.37}$$

In the following we consider these cases in more detail to derive the corresponding update rules:

- For the case when $a^T u + b > 0$, *i.e.* $\boldsymbol{\rho(u) > 0}$, we obtain the update rule

$$u = \hat{u} + \tau\lambda a \tag{4.38}$$

  Then we insert (4.38) into the condition $\rho(u) > 0$ which gives us the condition with respect to $\hat{u}$ as

$$\begin{aligned} a^T (\hat{u} + \tau\lambda a) + b &< 0 \\ a^T \hat{u} + b + \tau\lambda a^T a &< 0 \\ \rho(\hat{u}) &< -\tau\lambda a^T a\,. \end{aligned} \tag{4.39}$$

- Similarly, for $\boldsymbol{\rho(u) < 0}$, the update is

$$u = \hat{u} - \tau\lambda a \tag{4.40}$$

  and the corresponding thresholding check yields

$$\rho(\hat{u}) > \tau\lambda a^T a\,. \tag{4.41}$$

- Next, we consider the case when $\boldsymbol{\rho(u) = 0}$. Here, we add a Lagrange multiplier μ to (4.34). At first, the optimality condition can be computed as

$$u = \max_{\mu} \min_{u} \left\{ \frac{1}{2\tau} ||u - \hat{u}||_2^2 + \lambda \left\langle \mu, a^T u + b \right\rangle \right\}, \tag{4.42}$$

  yielding

$$\frac{\partial}{\partial u}: \quad u - \hat{u} + \tau\lambda\mu a = 0 \tag{4.43}$$

$$\frac{\partial}{\partial \mu}: \quad a^T u + b = 0\,. \tag{4.44}$$

Expressing u from (4.43) and inserting it in (4.44) yields

$$\begin{aligned} a^T\hat{u} + b - \tau\lambda\mu a^T a &= 0 \\ \mu &= \frac{\rho(\hat{u})}{\tau\lambda a^T a}\,. \end{aligned} \tag{4.45}$$

Then, μ is in turn inserted into (4.43) which finally leads to the update

$$u = \hat{u} - a\frac{\rho(\hat{u})}{a^T a} \tag{4.46}$$

The following table summarizes the different thresholding checks and the corresponding updates:

Table 4.2.: Thresholding checks and corresponding updates for solving the resolvent operator for the ℓ_1 data term.

condition	thresholding check	update
$\rho(u) > 0$	$\rho(\hat{u}) < -\tau\lambda(\nabla I_1)_{i,j}$	$u_{i,j} = \hat{u}_{i,j} + \tau\lambda(\nabla I_1)_{i,j}$
$\rho(u) < 0$	$\rho(\hat{u}) > \tau\lambda(\nabla I_1)_{i,j}$	$u_{i,j} = \hat{u}_{i,j} - \tau\lambda(\nabla I_1)_{i,j}$
else	$\lvert\rho(\hat{u})\rvert \leq \tau\lambda(\nabla I_1)_{i,j}$	$u_{i,j} = \hat{u}_{i,j} - (\nabla I_1)_{i,j}\frac{\rho(\hat{u}_{i,j})}{\lvert\nabla I_1\rvert^2_{i,j}}$

For a shorter notation, we define to use the notation $\text{shrink}(\hat{u})$ for the above soft-thresholding (shrinkage) scheme. With the computed resolvent operators we can then give the iterates for the primal-dual updates. For the initial setting $n = 0$ the primal as well as the dual variable is set to zero. For $n > 0$ we then perform the iterates

$$\begin{aligned} p^{n+1} &= \text{prox}_P\left(p^n + \sigma\nabla\overline{u}^n\right) \\ u^{n+1} &= \text{shrink}\left(u^n - \tau\text{div}\, p^{n+1}\right) \\ \overline{u}^{n+1} &= 2u^{n+1} - u^n\,. \end{aligned} \tag{4.47}$$

4.4.3. Iterative Coarse-To-Fine/Warping Scheme

Remember that the linearized version of the OFC (4.7) is defined via an incremental motion vector (4.4). In this case, the function $u(x)$ is linearized around a given $u_0(x)$. Therefore, the updates are only reasonable within a small vicinity around $u_0(x)$. This also means, that the resultant flow field $u(x)$ must not differ too much from the initial estimate $u_0(x)$. This is the reason why such methods are only valid to estimate small displacements. In order to

estimate optical flow for larger displacements, it is common practice to embed the approach in a coarse-to-fine framework [Anandan, 1989, Black and Anandan, 1996, Mémin and Pérez, 2002, Brox et al., 2004, Papenberg et al., 2006]. Alternatively, using a non-linearized OFC to allow for larger displacements was proposed by Nagel and Enkelmann [1986], and Alvarez et al. [2000]. Brox et al. [2004] has shown the equivalence of such approaches to the mathematical foundations of the coarse-to-fine/warping scheme. Recently, methods that use a feature stage to model large displacements was proposed by Xu et al. [2010, 2011], Brox and Malik [2010].

In our framework, we rely on the classical approach of embedding the optimization routine into a pyramidal coarse-to-fine framework. Otherwise, the optimization with a gradient approach might get stuck in local minima when the disparities of pixels are bigger than the discretization for the algorithm [Brox and Malik, 2010]. We denote the scale factor between neighboring levels as $\zeta \in (0, 1)$. For the algorithm, first multi-resolution pyramids with L levels are generated. Therefore, the input images are iteratively smoothed and subsampled. Subsequently, the results are then prolongated from the coarse to the fine level and used for its initialization. In addition, we add the possibility to filter the components of the optical flow field before prolongation with a 3×3 median filter. This reduces outliers in the optical flow field coming from wrong matches in coarse levels. Figure 4.5 shows a flow-chart to illustrate the coarse-to-fine approach. Note, that the "solve" step comprises the warping iterates.

Still, the creation of the multi-resolution pyramid is a crucial step. For moderate disparities the well studied Gaussian pyramid, generated by repeated Gaussian convolution and a scaling factor of $\zeta = 0.5$, is sufficient. This setting is well-studied and lots of literature exist [Burt, 1981, Crowley, 1981, Burt and Adelson, 1983, Lindeberg, 1994]. Though, for covering large motion of fine details a finer pyramid is mandatory and there the theoretical justifications are vague. Empirical experiments showed that three different settings of $\zeta = \{0.5, 0.8, 0.95\}$ cover most of the occurring conditions.

In order to avoid aliasing artifacts, the prefiltering of the input images must be adapted to the used scaling factor. For details on aliasing in image processing and, especially, texture mapping we refer to the work of Heckbert [1989]. There, the presented low-pass filters for avoiding aliasing when resampling the image is often time and memory consuming. Therefore, we decided to stick to classical Gaussian convolution with $\sigma = 0.3\sqrt{\zeta}$ and use standard linear interpolation for scaling operations.

In addition to the coarse-to-fine approach, we apply warping steps between the algorithm iterates on a level. After a defined amount of iterations the moving image is warped towards the fixed image. The iterates of the optimization are then continued with the recomputed spatial and temporal gradients. This reduces the displacements between the images within a pyramid level iteratively and help to overcome potential local minima. We denote the process of iteratively warping the input images towards each other as warping iterations. Moreover, the point of linearization u_0 is updated after each warping iteration. Again, we restrict the updates of flow vectors in an interval around u_0 to guarantee that

the linearization remains valid. In addition, we shrink this interval after each warping iteration with a factor κ_{u_0}. Experiments will show that this reduces the outliers on coarse levels and makes the median filtering at the prolongation step less crucial. Especially on finer levels the median filter can be omitted and then an unfiltered result is obtained there.

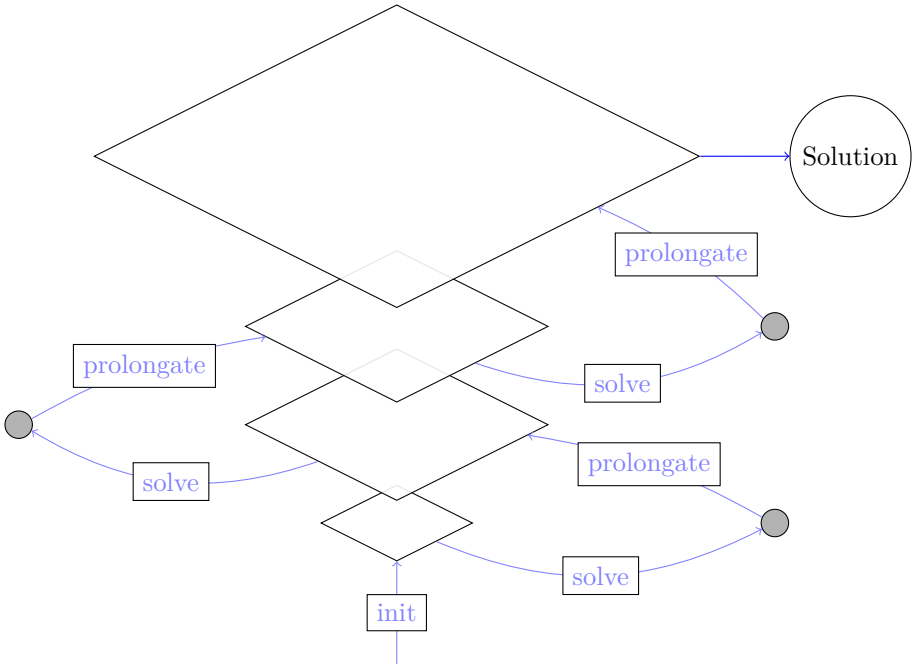
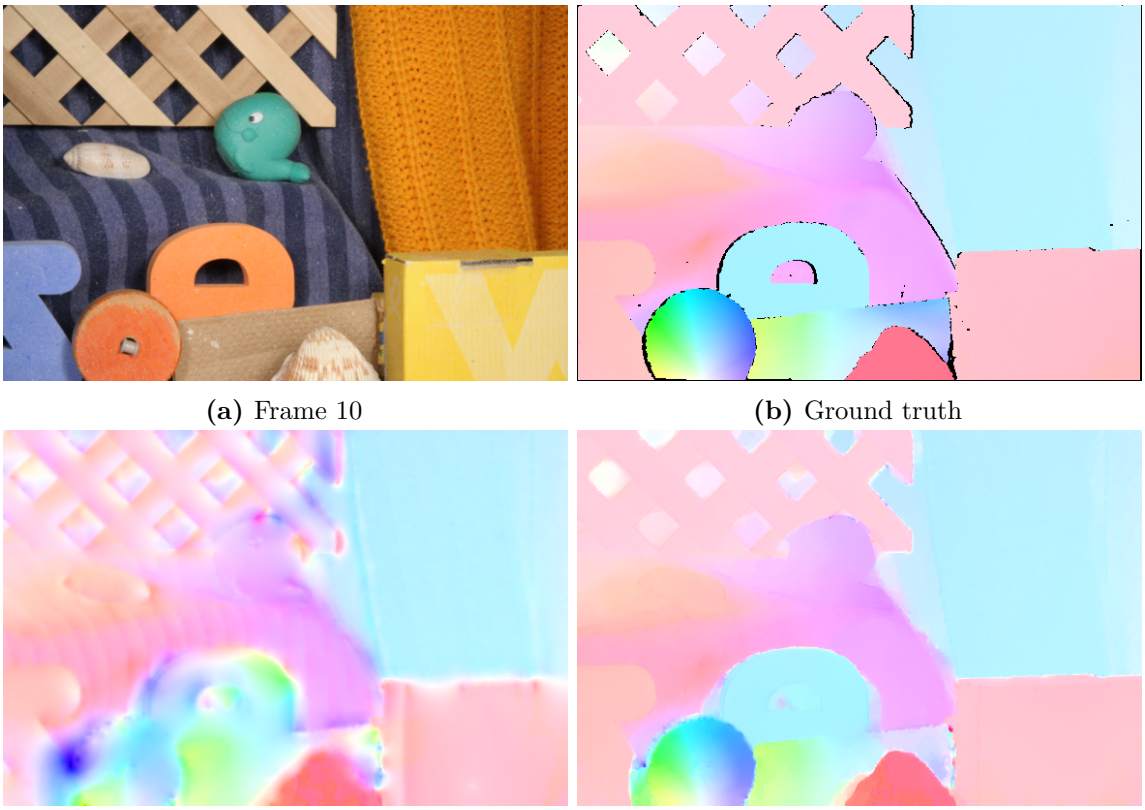


Figure 4.5.: Illustration of solving the coarse-to-fine scheme.

4.4.4. Discussion

Figure 4.6 shows a comparison of the previously presented Horn & Schunck model and the TV-ℓ_1 model. Both results were achieved using a coarse-to-fine warping scheme performing three warps per pyramid level with 100 iterations each. The scale factor between the pyramid levels was chosen with $\zeta = 0.5$. The results of the TV-ℓ_1 approach shows fewer outliers due to the ℓ_1 penalization in the data term. In addition, the oversmoothing effects caused by the ℓ_2 regularization of the Horn & Schunck Model are efficiently removed by using the total variation as smoothness prior.



(a) Frame 10 **(b)** Ground truth

(c) Horn & Schunck **(d)** TV-ℓ_1

Figure 4.6.: Comparison of the optical flow field computed with (c) the Horn & Schunck model [Horn and Schunck, 1981], and (d) the TV-ℓ_1 model.

CHAPTER
FIVE

OPTICAL FLOW ENHANCEMENTS

Motivated by the applications of optical flow there are two major points for improvement. On the one hand the robustness towards the input data, on the other hand the accuracy of the estimate. Ideally, the accuracy of the optical flow is increased and at the same time the method is more robust than the standard approach. In the following, we present some strategies on modifying a variational optical flow approach as presented in Section 4.3. We give insights on changes in the optimization approach.

Contents

5.1. Regularization Term

The subsequent chapters already demonstrated the effects of different regularizations. At first remember that some kind of regularization is needed in the context of variational optical flow estimation to overcome the data term's ambiguity. Recall the potential improvements when replacing a quadratic regularization by the total variation. Hence, the quality of the result is very dependent on the used smoothness prior. In the following we will present different modifications of the regularizer and we show how the model can be adapted towards different necessities. In general, we shall emphasize that there is no universal formulation that satisfies all different demands. Therefore it is a reasonable practice to have a framework where different regularizers (and later also data terms of course) can be used and comply with the current needs. In the following, the improvements of the regularizer are mainly about the type how to enforce the flow field to be smooth. On the one hand, typical artifacts caused by the total variation regularization are treated. On the other hand different weighting procedures are discussed to improve the optical flow estimate near motion boundaries.

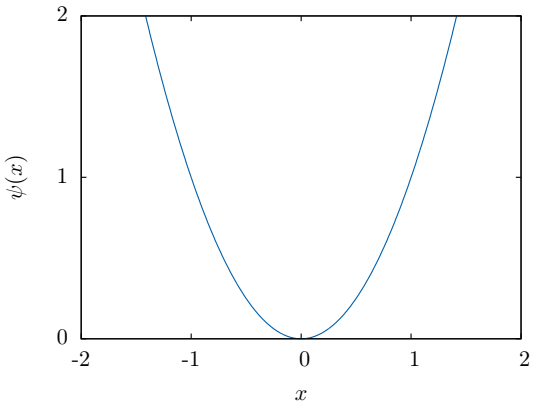
5.1.1. Huber-ℓ_1

While total variation (TV) regularization is a frequently used smoothness prior for variational methods, it exhibits some drawbacks especially in weakly textured areas. As discussed in Section 4.3.1, the principal reason for imposing a smoothness constraint is to overcome the ambiguity in certain areas due to the aperture problem. The total variation enforces a smooth flow field by penalizing its gradient magnitudes with an ℓ_1-norm. Due to the ℓ_1 norm's tendency to favor sparse solutions (*i.e.* lots of "zeros"), the fill-in effect caused by the regularizer leads to fronto-parallel solutions in weakly textured regions. Such piecewise constant areas are also prominent in larger regions where the data term is not strong enough to compute unique solutions. In the 1D setting this effect is known as "staircasing" artifact and we borrow this terminology to describe the piecewise constant results caused by the ℓ_1 regularization. In the field of robust statistics, Huber [1973] proposed an approach to combine a quadratic (ℓ_2) and linear (ℓ_1) penalization. Hence, the Huber norm, indicated as $|\cdot|_\varepsilon$, is defined by the two cases

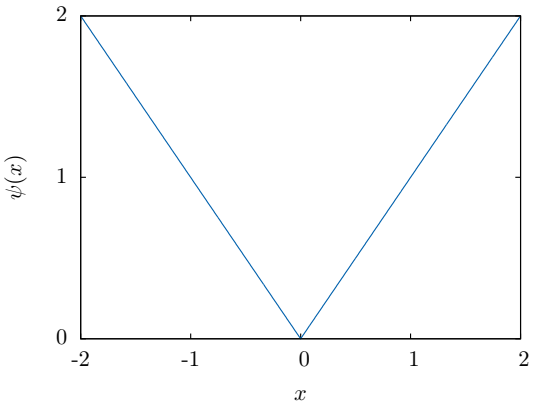
$$|s|_\varepsilon = \begin{cases} \frac{|s|^2}{2\varepsilon} & |s| \leq \varepsilon \\ |s| - \frac{\varepsilon}{2} & \text{else} \end{cases}, \tag{5.1}$$

where $\varepsilon > 0, \varepsilon \in \mathbb{R}$ defines the trade-off between ℓ_2 and ℓ_1 penalization. Figure 5.1 shows graphs of the different penalization functions using a quadratic norm, the total variation and the presented Huber norm.

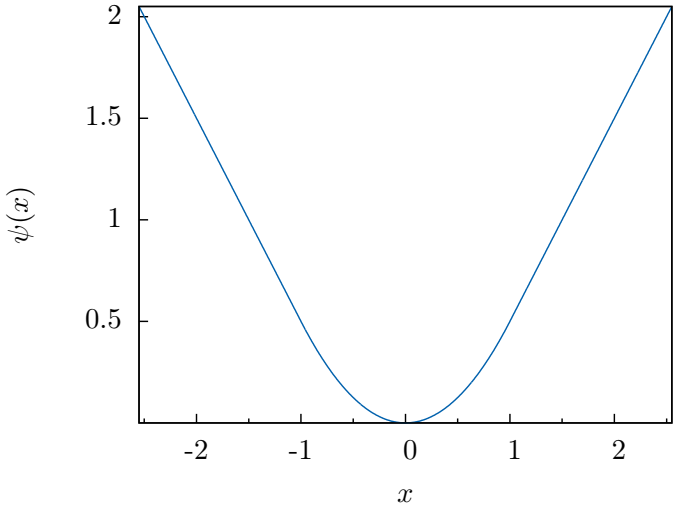
In the context of optical flow estimation, Shulman and Hervé [1989] were among the first to apply robust estimators as penalization functions. Using a Huber-norm to regularize the flow field reduces the staircasing effects because of penalizing small magnitudes of the



(a) $\psi(x) = |x|^2$



(b) $\psi(x) = |x|$



(c) $\psi(x) = |x|_\varepsilon$ with $\varepsilon = 1.0$

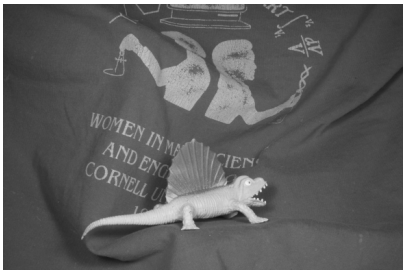
Figure 5.1.: Illustration of the (a) quadratic, (b) ℓ_1 and (c) Huber penalty function.

flow gradient by a quadratic norm. Let us now interchange the TV regularization in (4.22) with the Huber norm $|\nabla u|_\varepsilon$ and examine the changes in the optimization. Furthermore, let $||\nabla u||_\varepsilon$ denote the discretized Huber norm. Hence, the optimization problem yields

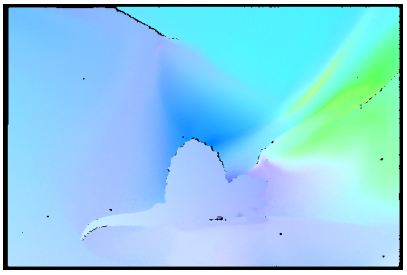
$$\min_{u \in Y} ||\nabla u||_\varepsilon + \lambda ||\rho(u)||_1 \ , \tag{5.2}$$

with the classical optical flow constraint (OFC) $\rho(u)$ (4.7). For applying the primal-dual algorithm we need the convex conjugate of the Huber function $F(u) = ||\nabla u||_\varepsilon$ yielding

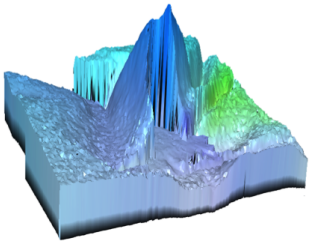
$$F^*(p) = \sup_{u \in Y} \{ \langle \nabla u, p \rangle_Z - ||\nabla u||_\varepsilon \} \ . \tag{5.3}$$



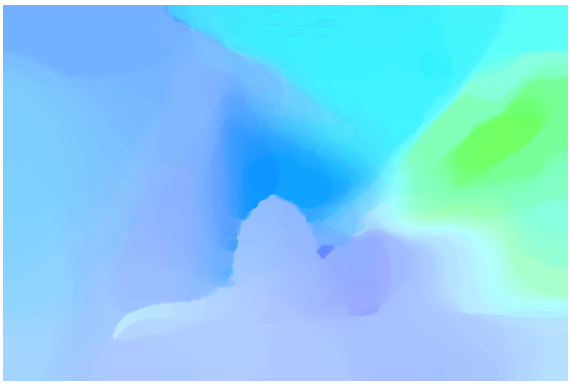
(a) Frame 10



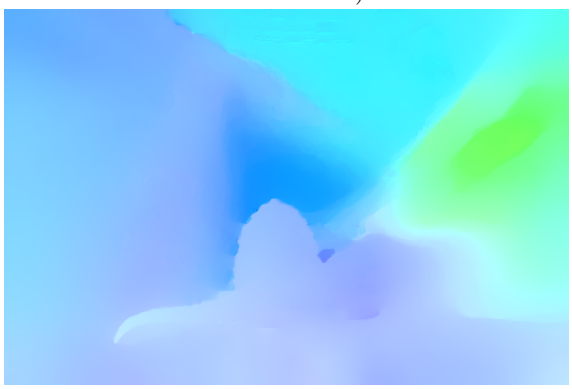
(b) Ground-truth



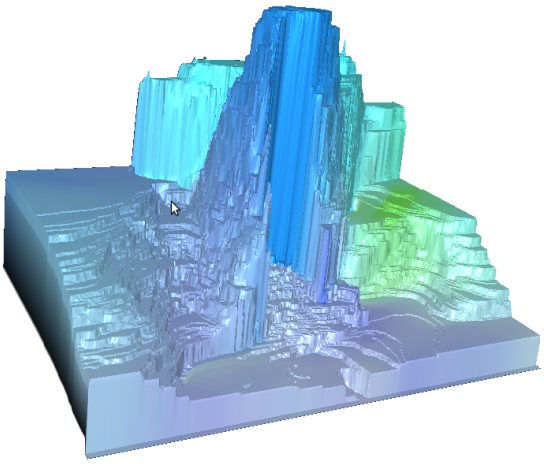
(c) Heightfield (ground-truth)



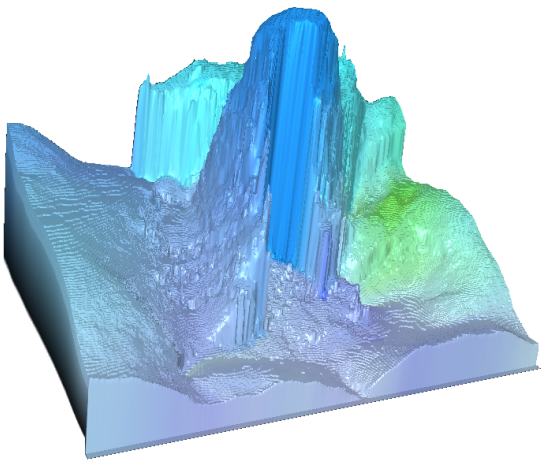
(d) TV



(e) Huber



(f) Heightfield (TV)



(g) Heightfield (Huber)

Figure 5.2.: Demonstration of the TV-prior's staircasing effect and the smoother result of the Huber norm. (a) one of the input images; (b) the groundtruth color coded optical flow and (c) the x^1-component's heightfield. (d) and (e) are the color-coded flow results, and (f) and (g) the corresponding heightfields. The heightfields are overlaid with the color-coded representation. The staircasing effect is clearly visible when using the TV prior (f) whereas the Huber-norm abates the zero-filling effects (g).

When computing this supremum as in (4.26), the cases of the Huber norm (5.1) must be considered and yield the optimization problem

$$F^*(p) = \sup_{u \in Y} \left\{ \langle \nabla u, p \rangle_Z - \begin{cases} \frac{|\nabla u|^2}{2\varepsilon} & |\nabla u| \leq \varepsilon \\ |\nabla u| - \frac{\varepsilon}{2} & \text{else} \end{cases} \right\} . \tag{5.4}$$

To compute the supremum of (5.4) we distinguish the respective cases:

- $|\nabla \boldsymbol{u}| \leq \boldsymbol{\varepsilon}$**:** In this cases the optimality condition is

$$\nabla u = \varepsilon p \tag{5.5}$$

  and the convex conjugate yields

$$F^*(p) = \frac{\varepsilon}{2} ||p||_2^2 . \tag{5.6}$$

  To get the condition with respect to the dual variable p we now re-insert (5.5) into the original condition $|\nabla u| \leq \varepsilon$ yielding $||p||_2 \leq 1$.

- **else:**

$$F^*(p) = \sup_{u \in Y} \left\{ \langle \nabla u, p \rangle_Z - |\nabla u| + \frac{\varepsilon}{2} \right\} \tag{5.7}$$

  comprises the conditions from the total variation which has been derived in (4.26). The additional term $\frac{\varepsilon}{2}$ is incorporated as follows:

$$F^*(p) = \begin{cases} \frac{\varepsilon}{2} ||p||_2^2 & ||p||_2 \leq 1 \\ \infty & \text{else} \end{cases} . \tag{5.8}$$

Combining the two cases we end up with the convex conjugate of the Huber regularization formulated as

$$F^*(p) = \frac{\varepsilon}{2} ||p||_2^2 + \delta_P(p) , \tag{5.9}$$

with the indicator function $\delta_P(p)$ (4.27). With that, incorporating the Huber norm changes the primal-dual saddle-point problem (4.31) to

$$\min_{u \in Y} \max_{p \in Z} \langle \nabla u, p \rangle_Z + \lambda ||\rho(u)||_1 - \frac{\varepsilon ||p||_2^2}{2} - \delta_P(p) . \tag{5.10}$$

Notice that the resolvent operator with respect to $G(u)$ is the same as for the TV-ℓ_1 optical flow model with the thresholding checks and updates presented in Table 4.2. With respect to $F^*(p)$ the resolvent operator $p = (I + \sigma \partial F^*)^{-1}(\hat{p})$ changes the pointwise minimizer (4.32) to

$$p_{i,j} = \text{prox}_P \left(\frac{\hat{p}_{i,j}}{1 + \sigma\varepsilon} \right) . \tag{5.11}$$

With that, the iterates of the primal-dual update yield

$$\begin{aligned} p^{n+1} &= \operatorname{prox}_P\left(\frac{p^n + \sigma\nabla\overline{u}^n}{1+\sigma\varepsilon}\right) \\ u^{n+1} &= \operatorname{shrink}\left(u^n - \tau\operatorname{div} p^{n+1}\right) \\ \overline{u}^{n+1} &= 2u^{n+1} - u^n\,. \end{aligned} \tag{5.12}$$

Here, prox_P again denotes a pointwise projection on the L^2 unit ball and shrink the soft-thresholding scheme (see Table 4.2).

The benefits when using a Huber-norm compared to the typically used TV-norm is presented in the illustrative example in Figure 5.2. There, heightfield plots are generated from the optical flow results of the `Dimetrodon` dataset from the Middlebury database. The experiment demonstrates the effects of the smoothness prior in regions with little texture such as the cloth in the background of Figure 5.2a. The x^1-component (u^1) of the optical flow vectors are plotted as heightfields to better visualize the leveling when the regularization propagates the vectors to homogeneous regions. The anticipated staircasing effects of the TV prior are clearly visible while the Huber norm induces smoother results in areas of vanishing gradients. Still, the Huber norm preserves edges due to the ℓ_1 regularization areas with larger gradient magnitudes like in textured areas (*cf.* Figure 5.2e). As long as the ε is reasonably chosen (for too large values the oversmoothing effects of the quadratic penalization will return) the Huber-norm does not exhibit any drawbacks compared to the TV regularization. We denote the combination of Huber regularization and classical OFC (5.2) as Huber-ℓ_1 model and use this model as the starting point for further investigations.

5.1.2. Weighted Huber-ℓ_1

Although the TV norm shows edge preserving capabilities of the optical flow field, it is often desired to improve the estimate near motion boundaries. For this reason, flow- and image driven regularizers have been proposed (*cf.* Section 4.3.1). In the following, we present three different types of image-driven priors and explain how to incorporate them into the previously introduced Huber-ℓ_1 model. At first we examine a rudimentary approach that incorporates a simple edge-weighting into the regularization. Then, we additionally use the direction of the edges to uncouple the weighting for the two components of the flow field. Finally, the directional information of the image gradients are used to incorporate a diffusion tensor-like weighting function. In Figure 5.3 we show the results of the different methods on the Middlebury's `Urban3` dataset. In addition, some cropped areas are presented in Figure 5.4 to better illustrate the approaches' characteristics.

The common idea of these approaches is to weight the regularization according to the

occurring image gradients. Assuming that image edges and motion boundaries coincide, the quality of the optical flow increases in those areas. Therefore, the smoothing influence is reduced across image edges while the smoothing along them is allowed. A simple approach is to use the gradient's magnitudes to directly weight the influence of the regularizer. This is strongly related to the weighted TV (also g-weighted TV or gTV) introduced by Bresson et al. [2005, 2007] in the context of image segmentation. They also showed that the weighted TV represents the Geodesic Active Contour model under certain conditions. Adding a pointwise weighting function $g(x)$ to the TV-norm (3.28) yields

$$\mathcal{R}(u) = \int_\Omega g(x) \left|\nabla u(x)\right| \mathrm{d}x \,. \tag{5.13}$$

Besides image segmentation, Leung and Osher [2005] used the weighted TV norm in combination with an ℓ_1 data fidelity term in the context of image denoising and inpainting.

5.1.2.1. Gradient Weighted Huber-ℓ_1

As mentioned above, the first approach uses the gradient magnitude to influence the regularization weight. Therefore, we incorporate a weighting function $g(x)$ into the Huber regularization which then reads

$$\mathcal{R}(u) = \int_\Omega \left|g(x) \nabla u(x)\right|_\varepsilon \mathrm{d}x \,, \tag{5.14}$$

$$\text{with} \quad g(x) = \exp\left(-\alpha \left|\left|\nabla I_0(x)\right|\right|_2^q\right) . \tag{5.15}$$

The function $g(x)$ weights the regularizer according to the image gradient's magnitude. Hence, the components u^1 and u^2 of the flow field are weighted equally. Note that unlike (5.13) we incorporate the weights into the Huber norm which simplifies the implementation. Before computing the gradients of the reference image I_0, the image is pre-filtered with a Gaussian filter ($\sigma = 1$) to reduce the influence of small gradients. When discretized, the gradient is computed using standard finite differences as in (3.12) and weighted according to (5.15) with $\alpha = 10$ and $q = 0.9$.

The numerical scheme of the Huber-ℓ_1 model changes only with respect to the gradient and divergence operator. For this reason we define a weighted gradient operator similar to (3.16) as

$$\left(\mathrm{diag}(g)\nabla\right) u = \left(g\,\partial_{x^1} u^1, g\,\partial_{x^2} u^1, g\,\partial_{x^1} u^2, g\,\partial_{x^2} u^2\right) . \tag{5.16}$$

Respectively, the divergence operator (3.17) changes to

$$\left(\nabla^T \mathrm{diag}(g)\right) p = \left(g\,\partial_{x^1} p^1 + g\,\partial_{x^2} p^2, g\,\partial_{x^1} p^3 + g\,\partial_{x^2} p^4\right) . \tag{5.17}$$

With that, the iterates for optimizing the edge-weighted Huber-ℓ_1 model are given as

$$\begin{aligned} p^{n+1} &= \operatorname{prox}_P\left(\frac{p^n + \sigma\left(\operatorname{diag}(g)\nabla\right)\overline{u}^n}{1+\sigma\varepsilon}\right) \\ u^{n+1} &= \operatorname{shrink}\left(u^n - \tau\left(\nabla^T \operatorname{diag}(g)\right)p^{n+1}\right) \\ \overline{u}^{n+1} &= 2u^{n+1} - u^n\,. \end{aligned} \tag{5.18}$$

5.1.2.2. Gradient Directed Weighted Huber-ℓ_1

An obvious extension to the edge weighting approach of the previous section is to take the direction of the gradient field into account. We change the weighting function (5.15) to separate the influence of the image gradients and define it as

$$g_d(x) = \left(g^1(x), g^2(x)\right)\,, \tag{5.19}$$

with

$$g_d^1(x) = \exp\left(-\alpha\,|\,\partial_{x^1} I_0(x)|^q\right)\,, \quad \text{and} \tag{5.20}$$

$$g_d^2(x) = \exp\left(-\alpha\,|\,\partial_{x^2} I_0(x)|^q\right)\,. \tag{5.21}$$

The straightforward changes of the gradient and the divergence operator yield

$$\left(\operatorname{diag}(g^1, g^2)\nabla\right)u = \left(g_d^1\,\partial_{x^1}u^1, g_d^2\,\partial_{x^2}u^1, g_d^1\,\partial_{x^1}u^2, g_d^2\,\partial_{x^2}u^2\right)\,, \quad \text{and} \tag{5.22}$$

$$\left(\nabla^T\operatorname{diag}(g^1, g^2)\right)p = \left(g_d^1\,\partial_{x^1}p^1 + g_d^2\,\partial_{x^2}p^2, g_d^1\,\partial_{x^1}p^3 + g_d^2\,\partial_{x^2}p^4\right)\,. \tag{5.23}$$

The algorithmic scheme for the iterates of the primal-dual update remain the same as in (5.18) replacing the gradient operator ∇^g with ∇^{g_d} defined in (5.22). Respectively, the divergence operator div^{g_d}, as in (5.23), is replacing the divergence operator div^g.

5.1.2.3. Tensor Directed Huber-ℓ_1

In Section 5.1.2.2 we already incorporate the direction of the gradient to weight the components of the regularizer. Still the directions are purely the two components of the gradients that are used separately to weight the smoothing process. To continue this approach we now integrate the full direction of the gradients into the Huber regularization. Therefore, we use a symmetric, positive definite diffusion tensor D to enforce regularization along image edges and to impede the smoothing across them. Replacing the edge-weighting

function $g(x)$ from (5.14) with a tensor-weight the regularizer becomes

$$\mathcal{R}(u) = \int_\Omega \left| D^{1/2} \nabla u(x) \right|_\varepsilon \, \mathrm{d}x \,. \tag{5.24}$$

We define the diffusion tensor

$$D^{1/2} = g(x)\, n(x)\, (n(x))^T + n^\perp(x)\, (n^\perp(x))^T \tag{5.25}$$

with the weighting function $g(x)$ as defined in (5.15), the vector

$$n(x) = \frac{\nabla I_0(x)}{||\nabla I_0(x)||_2} \,,$$

and its normal vector $n^\perp(x)$. Next, we define the elements of the 2×2 symmetric tensor matrix as

$$D^{1/2} = \begin{pmatrix} a & c \\ c & b \end{pmatrix} . \tag{5.26}$$

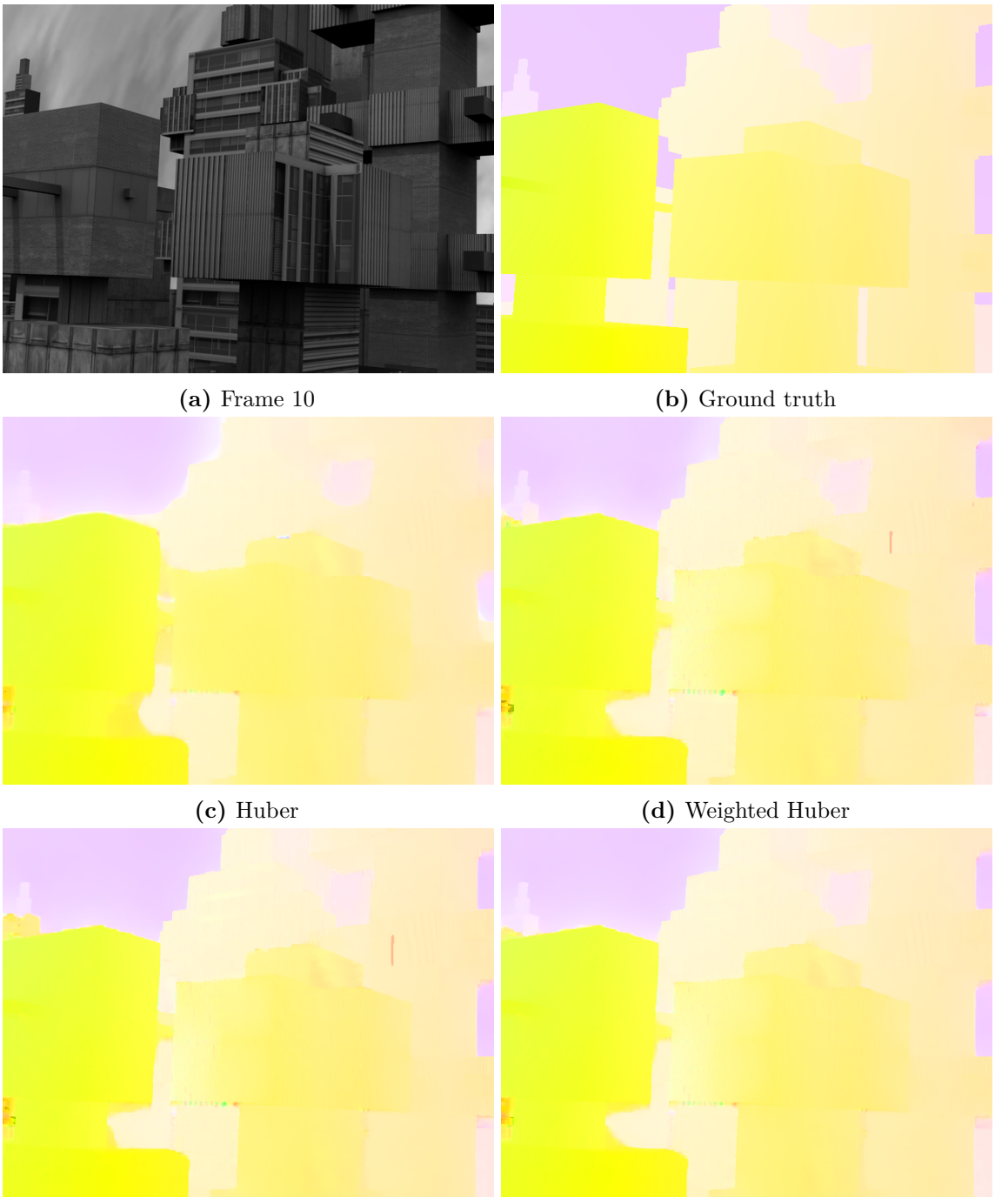
When optimizing a tensor directed Huber-ℓ_1 model the numerical scheme changes again with respect of the gradient and the divergence operator. The weighted operators are defined as

$$\left(D^{1/2}\nabla\right) u = \begin{pmatrix} a\,\partial_{x^1} u^1 + c\,\partial_{x^2} u^1 \\ c\,\partial_{x^1} u^1 + b\,\partial_{x^2} u^1 \\ a\,\partial_{x^1} u^2 + c\,\partial_{x^2} u^2 \\ c\,\partial_{x^1} u^2 + b\,\partial_{x^2} u^2 \end{pmatrix}, \quad \text{and} \tag{5.27}$$

$$\left(\nabla^T D^{1/2}\right) p = \begin{pmatrix} a\,\partial_{x^1} p^1 + c\,\partial_{x^1} p^2 + c\,\partial_{x^2} p^1 + b\,\partial_{x^2} p^2 \\ a\,\partial_{x^1} p^3 + c\,\partial_{x^1} p^4 + c\,\partial_{x^2} p^3 + b\,\partial_{x^2} p^4 \end{pmatrix} . \tag{5.28}$$

Consequently, the iterates of the primal-dual updates become

$$\begin{aligned} p^{n+1} &= \operatorname{prox}_P \left(\frac{p^n + \sigma \left(D^{1/2}\nabla\right) \overline{u}^n}{1 + \sigma\varepsilon} \right) \\ u^{n+1} &= \operatorname{shrink}\left(u^n - \tau \left(\nabla^T D^{1/2}\right) p^{n+1}\right) \\ \overline{u}^{n+1} &= 2u^{n+1} - u^n \,. \end{aligned} \tag{5.29}$$

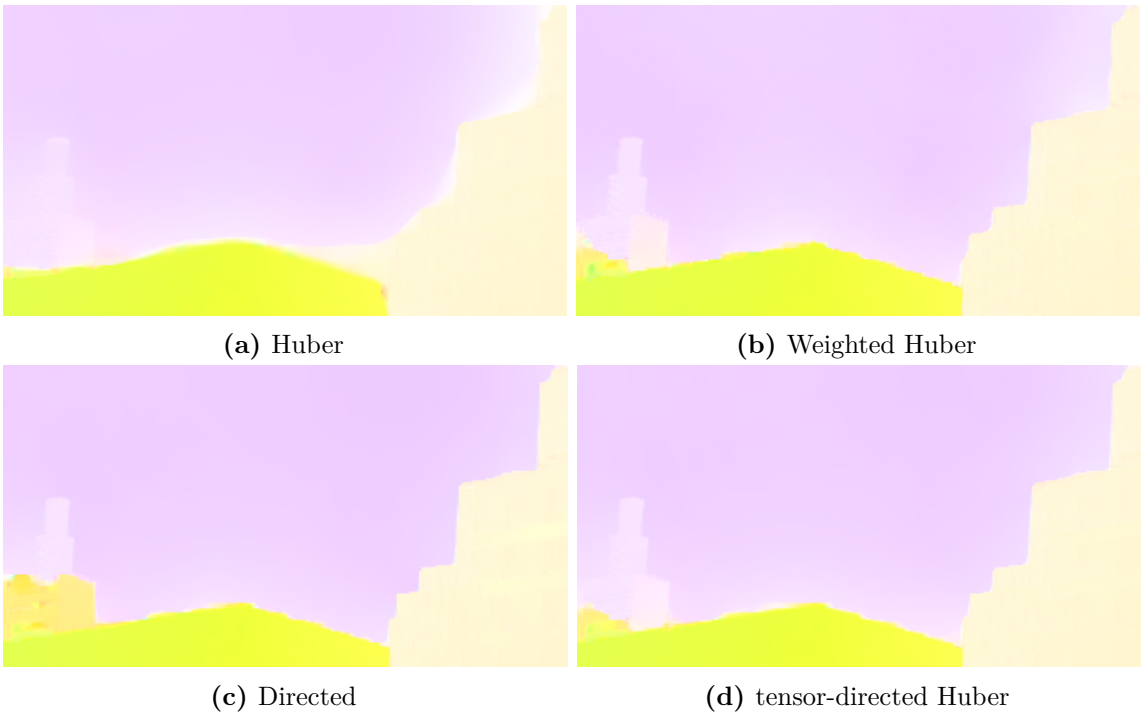


(a) Frame 10 **(b)** Ground truth

(c) Huber **(d)** Weighted Huber

(e) Directed Huber **(f)** Tensor-directed Huber

Figure 5.3.: Effects of the different methods to weight the regularization towards image structures shown on the optical flow estimate of the Middlebury's `Urban3` dataset. Frame 10 and 11 are used as input.



(a) Huber **(b)** Weighted Huber

(c) Directed **(d)** tensor-directed Huber

Figure 5.4.: Cropped regions of Figure 5.3 shows that the regularization weights enhance the details in the flow field.

(a) Weighted Huber **(b)** Directed Huber **(c)** Tensor-directed Huber

Figure 5.5.: Detail area of the results from Figure 5.3 demonstrating the possible artifacts due to strong texture. The tensor-directed Huber regularization is less prone to such artifacts than the approaches only relying on the edge strength.

5.1.2.4. Discussion

The presented variants of weighting the Huber regularization mainly differ on how to incorporate directional information. While the simple weighting case only uses gradient magnitudes the other two approaches use the directional information of the gradient to adapt the smoothing process accordingly. The gradient directed case uses the image's gradient directions to weight the components of the optical flow field differently. The tensor weighted Huber norm imposes the gradient's full orientation into the smoothing process.

Figure 5.3 shows the effect of the different approaches on the `Urban3/Middlebury` dataset. The results clearly improve near motion boundaries (*cf.* Figure 5.4). Still, the results of the three approaches are very similar. For this reason let us highlight the benefits of the presented methods and compare computation time for the models. The fact that image gradients are used to weight or direct the smoothing process may also lead to unwanted modifications of the regularizer. While the guidance of the regularization is desired when image edges coincide with motion boundaries, a weighting function caused by textured regions may lead to artifacts. Solely relying on the gradient's magnitude results in more outliers compared to the methods that also incorporate the directional information. Figure 5.5 demonstrates this effect of strongly textured areas.

The computation time to compute the results of the `Urban3/Middlebury` sequence are given in Table 5.1. We use a scale pyramid of the input images with a scale factor $\zeta = 0.8$, 10 warps per level and with 10 iterates of the primal-dual updates per warping iteration.

Table 5.1.: Computation time (algorithm only) of the weighted Huber-ℓ_1 models on an image pair of the `Urban3/Middlebury` dataset (image size: 640×480). The error is computed as the average end-point error (see Section 5.3).

Method	computation time [ms]	error [epe]
Huber-ℓ_1	140	0.485
gradient-weighted Huber-ℓ_1	155	0.398
gradient-directed Huber-ℓ_1	165	0.383
tensor-directed Huber-ℓ_1	180	0.385

Considering the results and computation time we suggest to use the tensor weighted Huber-ℓ_1. For the given dataset the accuracy is almost the same for all three approaches. Considering general data, the tensor directed approach is the superior approach when it comes to robustness towards different kind of input data. Especially for strongly textured regions, the tensor-directed approach is more robust then the other weighing variants.

5.1.3. Nonlocal Huber-ℓ_1

The Huber regularization as well as its weighted variants only consider "local"* derivative operators and hence the prior is not aware of higher order structures. The terminology local refers to the type of how the gradient operator is computed. Hence, local regularizers only use local neighbors, *i.e.* direct neighbors, for computing the gradient operator as in (3.16). As a consequence also the divergence operator is defined (3.17). What if we integrate bigger spatial neighborhoods in order to group similar moving regions together? Assume that an object (or at least parts of this object) exhibit a similar motion pattern. If this spatial coherence can be integrated into the regularization term, several restrictions of local regularizers can be overcome:

- In poorly textured regions, a stronger regularization can improve the filling-in effect and transport valid optical flow vectors from areas where a matching can be performed into ambiguous areas.
- The optical flow cannot be computed in occluded regions and the regularization must take care of those regions if no explicit occlusion handling is performed.
- Small scale structures are canceled due to the smoothness prior as the isotropic regularization does not adapt well to local image structures, especially small structures. Anisotropic smoothness priors (*cf.* Section 5.1.2) improve the quality in such regions but still fine details with relatively large motion might be canceled from the optical flow field.

The demand of a stronger regularization with an expanded proximal influence brings us to the well-known neighborhood filters. To fit our framework we use a nonlocal extension of the Huber norm which is then defined via the derivative operators computed in a nonlocal sense. Bigger neighborhoods are taken into account and an additional weighting function models the pixel's connections within this area. The support weights between the single pixels further strengthen their cohesion. Before defining the energy functional, we go through the most important contributions that incorporate such neighborhood filters.

5.1.3.1. History

The idea of neighborhood filters is based on the works of Lee [1983] (sigma filter), and Yaroslavsky [1985] (Yaroslavsky filter). The intention of those filters has been to impede the smoothing of image edges when denoising an image. This has been achieved by restricting the filter operation towards similar pixels (similar in the sense of *e.g.* similar gray values). Aurich and Weule [1995] picked up this thought and proposed to add an additional weight to a Gaussian filter to integrate data information and adapt the Gaussian kernel to underlying image structures. The resultant nonlinear Gaussian filter enhanced the edge preserving

* Also TV regularization and most of its image and flow-driven variants belong to the class of local regularization.

capabilities of the smoothing process. Similarly, Smith and Brady [1997] proposed the "SUSAN" filter where a mask restricts the Gaussian filtering towards similar gray values. Then, Tomasi and Manduchi [1998] shaped the term of the bilateral filter. Again, the use of a nonlinear combination of image values defined the filtering kernel. Additionally, Smith and Brady [1997], and Tomasi and Manduchi [1998] included a proximal weighting to strengthen the coherence of contiguous pixels. Further investigations in this direction were made by Buades et al. [2005b,a] with the nonlocal means algorithm. In addition, Buades et al. [2005b] showed the relations between bilateral filters and PDEs in the sense of Perona and Malik [1990]. Connections between the mentioned neighborhood filters were established by Buades et al. [2006] and Paris et al. [2009]. Moreover, Barash [2004, 2002] discussed the relations to nonlinear diffusion processes.

In a graph-based learning framework, Zhou and Schölkopf [2004, 2005] proposed a nonlocal gradient and divergence operator for the regularization of such discrete structures. Such weighted graphs were then also used by Bougleux et al. [2007] and Elmoataz et al. [2008]. For the continuous setting, at first Kindermann et al. [2004] and later Gilboa and Osher [2007] incorporated neighborhood filters into an energy functional. Those works shaped the terminus of nonlocal TV regularization. Moreover, the weights that were used in the nonlocal regularizer from Gilboa and Osher [2007] are strongly related to the ones used by Zhou and Schölkopf [2004, 2005].

To define the connectedness of pixels in specified areas principles of Gestalt psychology have often been considered. The so-called Gestalt principles of grouping suggest that elements of certain properties belong together. The most often used properties are proximity and similarity. They imply that it is likely that nearby elements belong together (*proximity*). In addition, elements can be grouped together according to their appearance (color, texture, *etc.*) to strengthen the relation between pixels (*similarity*). Of course also other properties could be used to increase the connectedness within regions but indeed proximity and similarity are the most important and most often used ones.

Especially for stereo algorithms, more precisely for cost volume filtering, the benefits of such nonlocal regularization have already been emphasized. Yoon and Kweon [2006] incorporated support weights based on the Gestalt principles. These weights are computed as a combination of the pixels' color differences and their spatial distances. A subsequent work by Yang et al. [2009] used similar support weights to build the cost volume for computing the disparities. For optical flow estimation, Xiao et al. [2006] used a bilateral filter to regularize the optical flow field. Therefore, an initial flow estimate is made using an anisotropic regularizer in combination with a gray-value constancy data term. Then, occlusions are computed and a bilateral filter is used to propagate flow vectors from regions that are similar in appearance and motion into the occluded areas. Subsequently, Sand and Teller [2008] simplified this idea and incorporated the bilateral filter as regularization in areas near motion boundaries in the framework of Brox et al. [2004].

5.1.3.2. Nonlocal Huber Regularization

According to Yoon and Kweon [2006] we define support weights that combine the influence of similarity and proximity of adjacent pixels. Theoretically, the neighborhood for computing the support weights could be expanded to the whole image but the computation time would be too high and the memory consumption for solving such a system would be not manageable at the moment. Hence, a neighborhood system $\mathcal{N}_x \subseteq \Omega$ defines a set of pixels y that are within a defined photometric and geometric vicinity to x. The similarity measure for the spatial distance between two points x and y is defined using a simple Euclidean distance *i.e.* the proximity

$$\Delta_s = d(x, y) \tag{5.30}$$

and the photometric similarity as an Euclidean distance in a color space (*e.g.* grayscale, RGB or LAB) that is defined as

$$\Delta_c = d(I(x), I(y)). \tag{5.31}$$

In combination, the support weights within $\mathcal{N}_x$ yield

$$g(x, y) = \exp\left(-\left(\frac{\Delta_c(x, y)}{w_c} + \frac{\Delta_s(x, y)}{w_s}\right)\right). \tag{5.32}$$

The parameters $w_c \in \mathbb{R}$ and $w_s \in \mathbb{R}$ weight the influence of color similarity and proximity. In addition, we assume that the weights are symmetric such that the weighting $g(x, y)$ between pixel x and y with $y \in \mathcal{N}_x$ is equal to the weighting $g(y, x)$ when $x \in \mathcal{N}_y$, hence, $g(x, y) = g(y, x)$. Figure 5.8 shows exemplary support weights for some selected regions of the `Army/Middlebury` sequence. These examples reveal that the support weights essentially incorporate a low level segmentation, based on the underlying image data, into the regularization. Incorporating the support weights into the smoothness term yields the nonlocal Huber regularizer

$$\mathcal{R}(u) = \int_\Omega \int_{\mathcal{N}_x} g(x, y) \left|(u(y) - u(x))\right|_\varepsilon \, \mathrm{d}y \, \mathrm{d}x. \tag{5.33}$$

5.1.3.3. Minimization

In order to discretize (5.33) we first define the discretized support weights within a restricted neighborhood. As already mentioned, theoretically this neighborhood can be extended to the whole image domain. Nonetheless, we define a window of the size $2w + 1$, the number of neighbors $L = 2w(w + 1)$, and a set of offset vectors

$$\begin{aligned} \xi &= (\underbrace{-w, \ldots, 0, \ldots, w}_{w\text{-times}}, \ldots, -w, \ldots, -1) \\ \eta &= (\underbrace{-w, \ldots, -w}_{2w+1\text{-times}}, \underbrace{-w+1, \ldots, -w+1}_{2w+1\text{-times}}, \ldots \underbrace{0, \ldots, 0}_{w\text{-times}}). \end{aligned} \tag{5.34}$$

Afterwards, we define the neighboring pixels of (i, j) as

$$\left(i_n^+, j_n^+\right) = (i + \xi_n, j + \eta_n)$$

and

$$(i_n^-, j_n^-) = (i - \xi_n, j - \eta_n) \ ,$$

with $n = 1 \dots L$. Next, we define the discrete version of the support weights as

$$g_{i,j,i^+,j^+} = \exp\left(-\left(\frac{(\Delta_c)_{i,j,i^+,j^+}}{w_c} + \frac{(\Delta_s)_{i,j,i^+,j^+}}{w_s}\right)\right) . \tag{5.35}$$

Note that the symmetric definition of the support weights defines $g_{i^-,j^-,i,j} = g_{i,j,i^+,j^+}$. Figure 5.6 shows an illustration of a neighborhood with window radius $w = 2$. The corresponding weights are depicted in Figure 5.7. The symmetry of the weights is hinted by only defining the weights for half of the neighborhood patch. Finally, we show the offset vectors for the specific case $w = 2$:

$$\begin{array}{rcccccccccccc} \xi = & (-2, & -1, & 0, & 1, & 2, & -2, & -1, & 0, & 1, & 2, & -2, & -1) \\ \eta = & (-2, & -2, & -2, & -2, & -2, & -1, & -1, & -1, & -1, & -1, & 0, & 0) \end{array} \tag{5.36}$$

Moreover, we define the nonlocal gradient operator $K^{\mathrm{NL}} : Y \to Z^+$, with $Z^+ = \mathbb{R}^{LMN}$ as

$$K^{\mathrm{NL}} = \begin{bmatrix} \delta_1^+ \\ \delta_2^+ \\ \vdots \\ \delta_L^+ \end{bmatrix} , \tag{5.37}$$

with the nonlocal forward differences (δ_n^+) with Neumann boundary conditions defined as

$$(\delta_n^+ u)_{i,j} = \begin{cases} u_{i_n^+,j_n^+} - u_{i,j} & \text{if} \quad (i_n^+, j_n^+) \in \Omega \\ 0 & \text{else} . \end{cases} \tag{5.38}$$

Respectively, we define the discretized adjoint operator $(K^{\mathrm{NL}})^* : Z^+ \to Y$:

$$(K^{\mathrm{NL}})^* = \sum_{n=1}^{L} \delta_n^- , \tag{5.39}$$

and the nonlocal backward differences

$$(\delta_n^- p)_{i,j} = \begin{cases} \left(p_{i,j} - p_{i_n^-,j_n^-}\right) & \text{if} \quad (i_n^+, j_n^+) \in \Omega \quad \text{and} \quad (i_n^-, j_n^-) \in \Omega \\ -p_{i_n^-,j_n^-} & \text{if} \quad (i_n^+, j_n^+) \notin \Omega \quad \text{and} \quad (i_n^-, j_n^-) \in \Omega \\ p_{i,j} & \text{if} \quad (i_n^+, j_n^+) \in \Omega \quad \text{and} \quad (i_n^-, j_n^-) \notin \Omega \\ 0 & \text{else} . \end{cases} \tag{5.40}$$

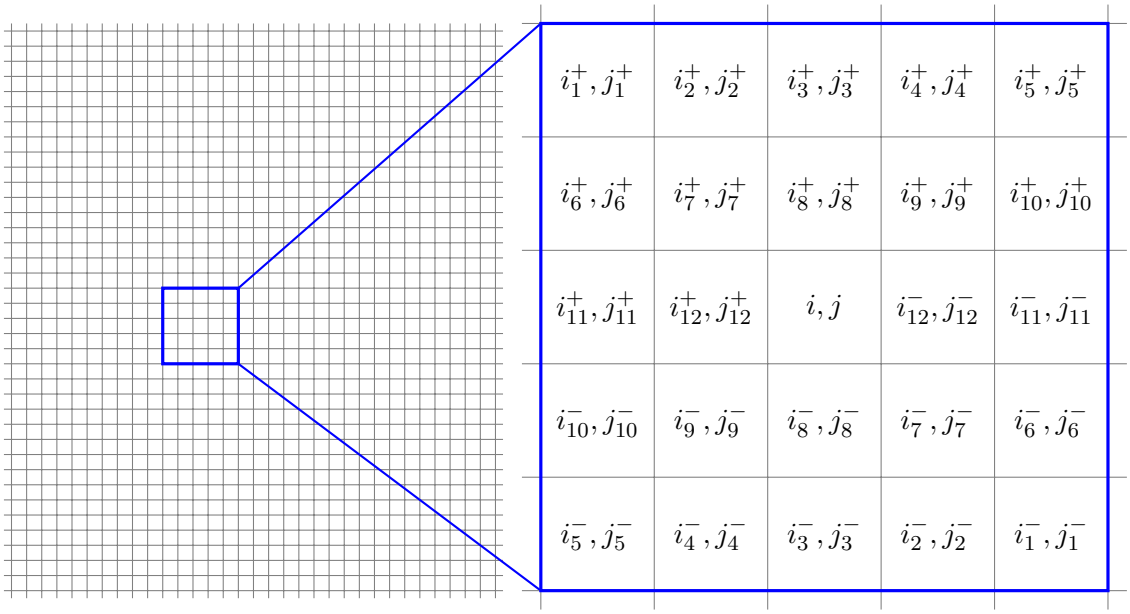


Figure 5.6.: Illustration of an exemplar neighborhood patch of the pixel (i, j).

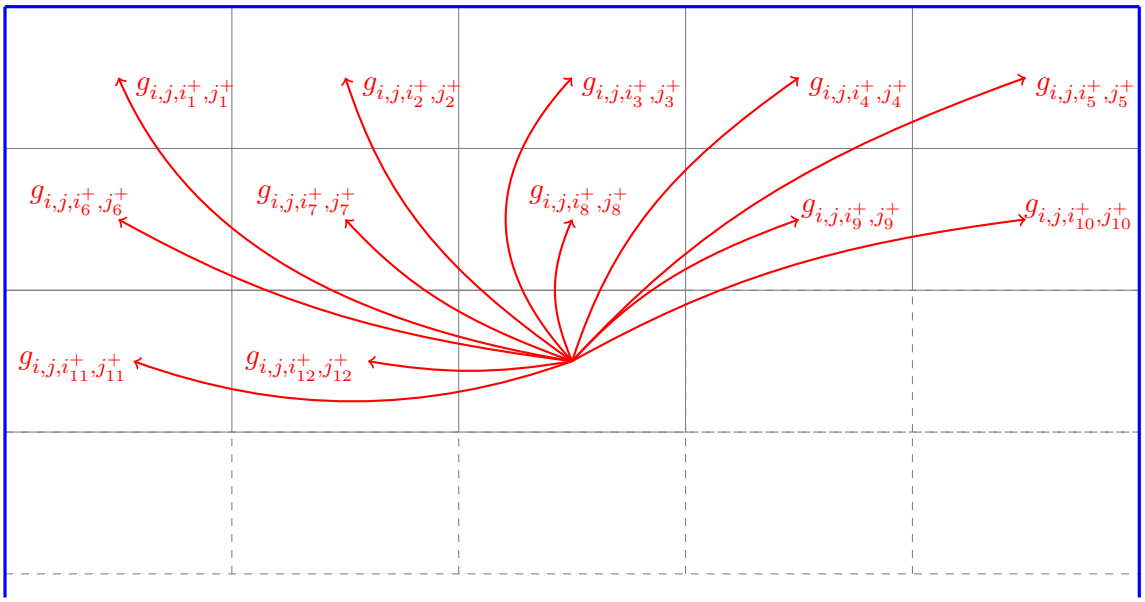
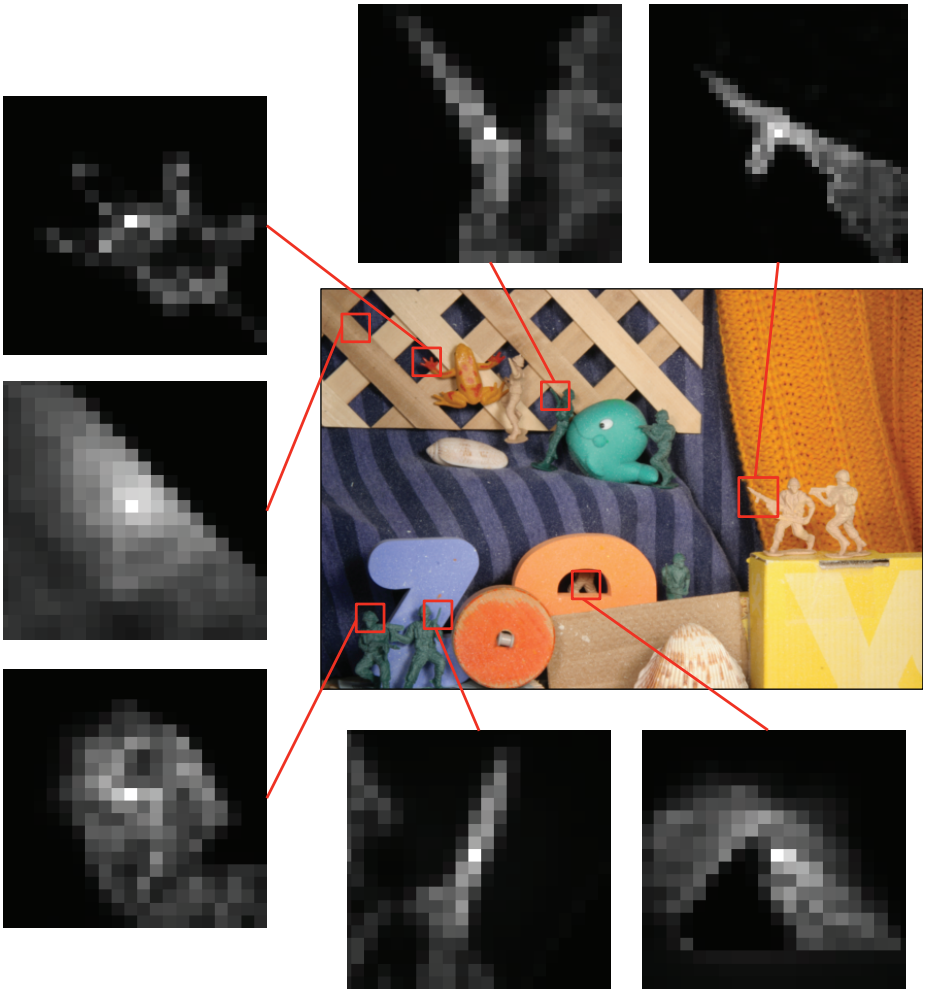
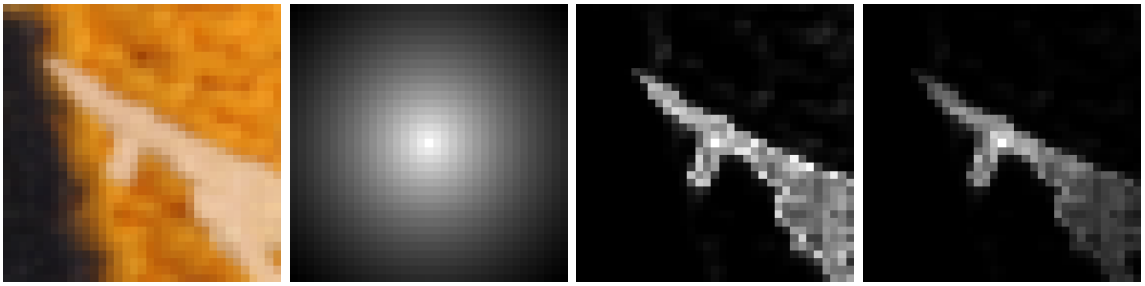


Figure 5.7.: Support weights for an exemplar neighborhood patch of pixel (i, j). As the weights are symmetric, they are only defined for half of the patch.



(a)



(b) patch **(c)** proximity **(d)** similarity **(e)** support weight

Figure 5.8.: Support weights of some exemplar patches of the Middlebury's `Army` sequence. In (b-e) the individual elements for computing the support weights are shown for the patch in (b). The support weights (e) are a combination of the distances (*i.e.* proximal weight) (c) and the pixel similarities (d)

With that we can define the discrete version of the nonlocal Huberℓ_1 model as

$$\min_{u\in Y} \left|\left|K^{\mathrm{NL}}u^1\right|\right|_\varepsilon + \left|\left|K^{\mathrm{NL}}u^2\right|\right|_\varepsilon + \lambda \left|\left|\rho(u)\right|\right|_1 . \tag{5.41}$$

Until now we did not explain how to incorporate the support weights into the framework. When we introduced the usage of the weighted Huber-norm in Section 5.1.2, we directly incorporated the weights into the gradient and divergence operator. Now, we want to demonstrate an alternative variant and use the support weights to modify the reprojection step of the dual variables. Therefore, we recall the resolvent operator $p = (I + \sigma\,\partial F^*)^{-1}(\hat{p})$ which is defined as the minimization problem

$$\arg\min_p \left\{ \frac{1}{2\sigma} \left|\left|p - \hat{p}\right|\right|_2^2 + \frac{\varepsilon}{2g} \left|\left|p\right|\right|_2^2 + \delta_{||p||\le g}(p) \right\} . \tag{5.42}$$

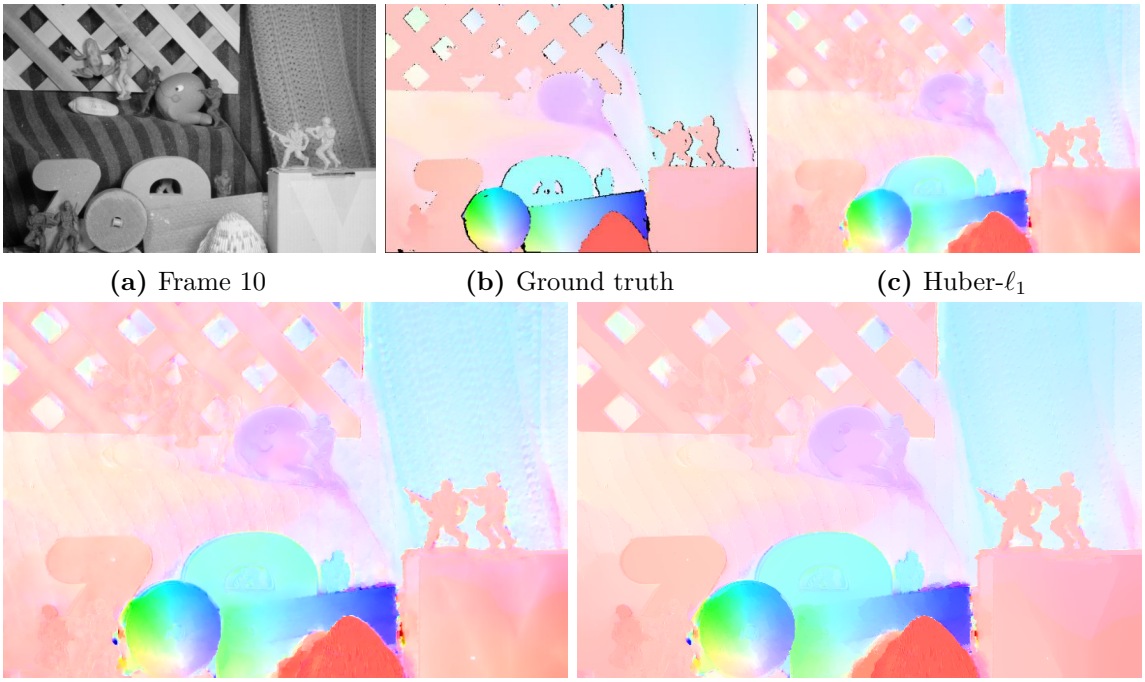
Hence, we project the dual variable not onto the unit ball but adapt the radius according to the weighting factor g such that the projection is made onto a ball $\mathcal{B}_g$ with radius g. With that we get the primal-dual algorithm (3.52) with the iterates

$$\begin{aligned} p^{n+1} &= \mathrm{prox}_{\mathcal{B}_g}\left(\frac{p^n + \sigma K^{\mathrm{NL}}\overline{u}^n}{1+\sigma\frac{\varepsilon}{g}}\right) \\ u^{n+1} &= \mathrm{shrink}\left(u^n - \tau\left(K^{\mathrm{NL}}\right)^* p^{n+1}\right) \\ \overline{u}^{n+1} &= 2u^{n+1} - u^n . \end{aligned} \tag{5.43}$$

5.1.3.4. Discussion

Figure 5.9 shows an exemplar result of the presented nonlocal Huber regularization in combination with the standard OFC as data term. There we especially want to emphasize the comparison to the tensor-directed regularization. In terms of accuracy of motion boundaries the rather simple `Army` sequence does not reveal any big differences. Still, fine and small details in the motion field are more accurate using the nonlocal regularizer. In terms of smoothness within the moving areas the nonlocal approach is clearly in favor. Especially for strong textured areas (*e.g.* courtain on the right) the nonlocal approach does not reveal the typical structures of the underlying texture as it is the case for the tensor-directed regularization. Still, keep in mind that the nonlocal approach is more complex to compute in terms of computation and also memory due to the larger neighborhood relations. In general, the nonlocal regularization is the "strongest" regularization presented in this thesis. It is possible to handle larger areas where no data information can be computed and also more prior knowledge can be incorporated in form of the neighborhood

relations. The defined weights are able to guide the regularization as desired which means that the quality of the optical flow field can be improved near motion boundaries, within untextured areas and also for unknown areas the weights can be modified to have positive influence on the flow field's accuracy. To anticipate, the nonlocal Huber regularization will be perfectly suited with strong and robust data terms where the other smoothness priors have too little influence to additionally guide the regularization.



(a) Frame 10 **(b)** Ground truth **(c)** Huber-ℓ_1

(d) Tensor-directed H.-ℓ_1 **(e)** nonlocal H.-ℓ_1

Figure 5.9.: Exemplar result of the nonlocal Huber regularization. We compare results on the `Army` sequence from (a) the Huber-ℓ_1 model, (d) its tensor-directed variant, and (e) the nonlocal Huber regularization with ℓ_1 data term. Already the tensor-directed approach enhances the result near motion boundaries but also introduces more noise in textured areas as the nonlocal model (*e.g.* curtain on the right). We set $\lambda = 30$ for all three models.

5.2. Data Term

In the upcoming Section we have a closer look on the data fidelity term and how we can improve its robustness. The first clear disadvantage is the general assumption about the brightness constancy which is violated at any change of illumination conditions. Although optical flow estimation mostly considers consecutive frames of video data, recorded in a tight temporal interval, the brightness constancy generally cannot be guaranteed. In fact this assumption is often violated for real-world data. In addition, not only changes in

the global illumination setup are an issue but also moving shadows, specular highlights or simple reflections violate the assumption of the points' constant intensities over time. Therefore, models that solely rely on this classical optical flow constraint (OFC) are bound to fail or at least exhibit some failures in the flow field. Moreover, it is not only the task to improve the robustness of the data term but also to improve the strength of the data fidelity.

To outline the improvements concerning the data term we define the Huber-ℓ_1 model (*cf.* Section 5.1.1) as the starting point for further improvements. Let us point out, that all other regularization terms can be combined with the considered data terms and we will show results from different combinations in Section 5.3.

5.2.1. Structure/Texture Decomposition of the Input Images

At first we will reflect a common practice of decomposing the input images into its structure and texture components. As already mentioned in the introduction, the robustness concerning illumination variations comes from reassembling the structure and texture components with different weightings. The illumination changes are mostly encoded in the structure part and a linear combination of the components (with more weight on the texture part) reassembles the input images and increases the robustness towards illumination changes. But shifting the weight of structure and texture part when reassembling the image also neglects potential information for optical flow computation. This is one of the major drawbacks when using structure/texture decomposed input images. It might occur that valid information is discarded of the image data and the quality might degrade in such areas. Nevertheless, this preprocessing step is widely used and one can obtain reasonable and stable results with this simple preprocessing step. The benefit of the approach is clearly the low complexity of computation and as it is simply a preprocessing step we can stick to the presented baseline model for computing the optical flow fields.

We utilize the ROF denoising model (3.24) [Rudin et al., 1992] to decompose the input image $I(x)$ into its structure $I_s(x)$ and texture $I_t(x)$ component. Hence, we solve the denoising problem

$$\min_{I_s} ||\nabla I_s||_{2,1} + \frac{\lambda}{2} ||I_s - I||_2^2 \tag{5.44}$$

with the primal-dual algorithm (3.52) as proposed in Chambolle and Pock [2010]. We chose they parameter $\lambda = 1.0$ for computing the structure part of the input image. The texture part is then simply computed by

$$I_t(x) = I(x) - I_s(x) \tag{5.45}$$

and finally the reassembled input image yields

$$\tilde{I}(x) = (1 - \vartheta) I_s(x) + \vartheta I_t(x) \ , \tag{5.46}$$

where ϑ defines the trade-off between structure and texture part. For all our experiments where we use structure-texture decomposed input images we set $\vartheta = 0.8$ unless otherwise stated.

Figure 5.10 shows the structure and texture parts of an image and the effects of using a recombination of those as the input for the Huber-ℓ_1 model. The influence of moving shadows, reflections (*e.g.* rear window of the car) and highlights are reduced. Generally, it can be observed that the results using decomposed input images are smoother which results from the sparser information for computing the data term. Hence, the decrease of used data information might effect the available details in the flow field.

5.2.2. Compensating Brightness Constancy Violations

Motivated by a number of works that model an explicit illumination model to describe OFC violations, we use a very simple variant to estimate the deviations of the classical OFC which is also used in Chambolle and Pock [2010]. Most of the related approaches discussed in Section 4.3.2 try to fit some kind of illumination model in the data but, in fact, not only illumination but also occlusions, noise, specular highlights, impairments, *etc.* can be the cause for not complying the classical OFC. The task for our model is to keep it simple but still be strong enough to handle such outliers. Therefore, we do not try to estimate an illumination model per se but, according to Chambolle and Pock [2010], add an additional variable that directly models the abnormalities of the OFC (4.7). Adding such an absorption variable $c\,(x) \in X$ we obtain the following variant of (4.1) by

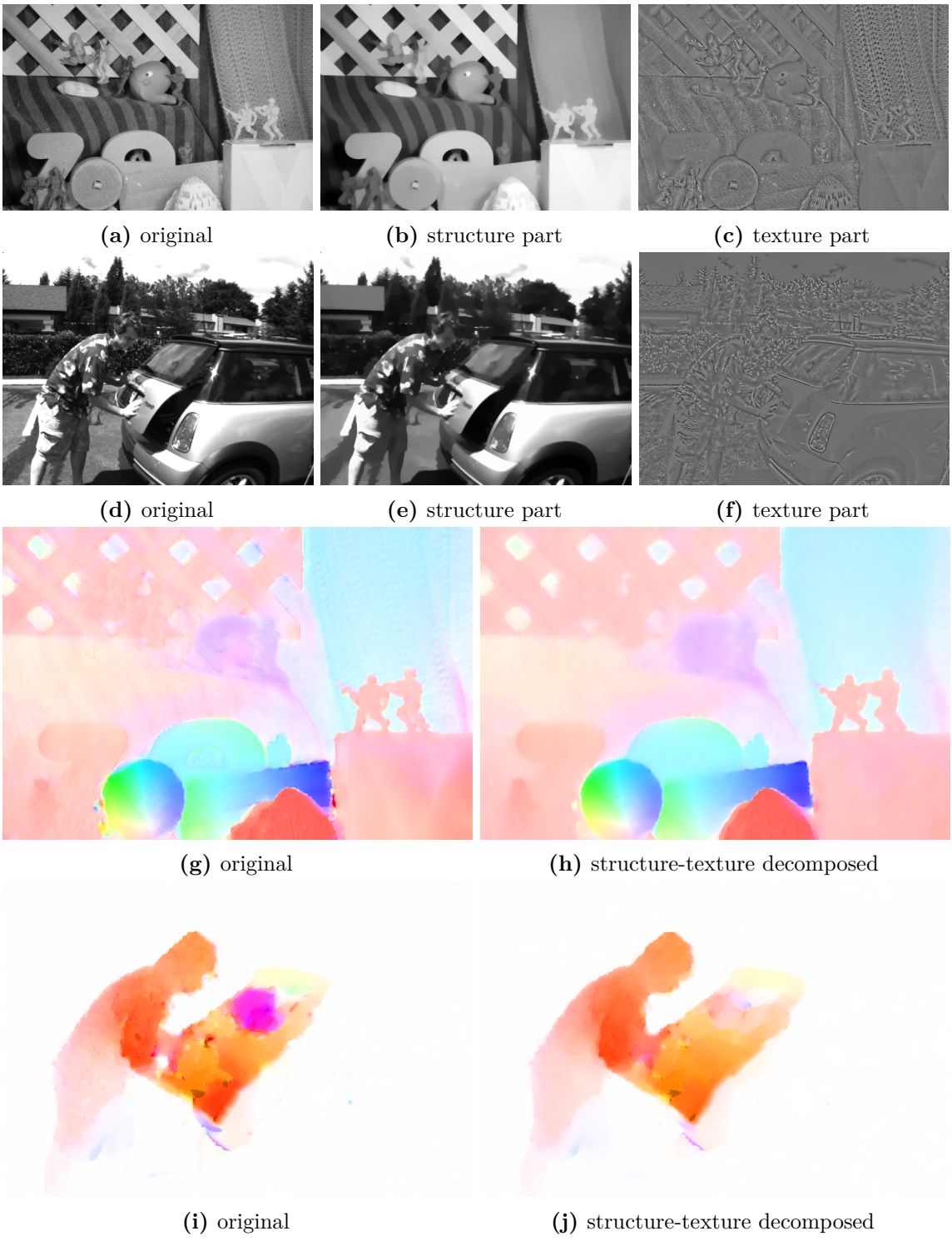
$$\frac{d}{dt} I(x(t), t) = -\gamma c\,(x)\ . \tag{5.47}$$

With that, we can formulate the modified OFC as

$$\rho(u\,(x)\,, c\,(x)) = I_t(x(t), t) + (\nabla I(x(t), t))^T (u\,(x) - u_0\,(x)) + \gamma c\,(x) = 0\,. \tag{5.48}$$

The real-valued parameter γ steers the influence of the compensation variable $c\,(x)$. This means that for low values of γ the optical flow is estimated in a rather classical way and only slight intensity differences are absorbed. Likewise, larger values of γ incorporate more severe outliers of the flow field into $c\,(x)$. This enables a stronger compensation for situations when the flow field would be severely degraded due to strong violation of (4.7). Attention should be paid to the fact that this behavior can also be exaggerated. A too large influence of the compensation may incode valuable information in the compensation variable while it should actually be encoded in the optical flow fiel.

Note that this modified OFC (5.48) is still linear and hence the resultant optimization problem remains convex. In addition to the absorption of outliers in the OFC, we want $c\,(x)$ to change smoothly and therefore an additional spatial regularization on $c\,(x)$ penalizes



(a) original **(b)** structure part **(c)** texture part

(d) original **(e)** structure part **(f)** texture part

(g) original **(h)** structure-texture decomposed

(i) original **(j)** structure-texture decomposed

Figure 5.10.: An input image of the Middlebury sequences `Army` (a-c) and `MiniCooper` (d-f) is decomposed into its structure and texture components. For the ROF denoising we use $\lambda = 10$. (g-j) compares results of using the original images as input versus using structure-texture decomposed images. The intensities of structure and texture images are rescaled for visualization.

the gradient variations by the already introduced Huber-norm:

$$\mathcal{R}(c) = \int_{\Omega} |Dc|_{\varepsilon} \tag{5.49}$$

Integrating the modified OFC (5.48) and the additional smoothness prior term (5.49) into the discretized energy formulation for computing the optical flow, we end up with the minimization problem

$$\begin{aligned} &\min_{u\in Y, c\in X} ||\nabla u||_{\varepsilon} + ||\nabla c||_{\varepsilon} + \lambda\, ||\rho(u,c)||_1 \;, \\ &\qquad \text{with} \quad \rho(u,c)_{i,j} = (I_t)_{i,j} + (\nabla I)_{i,j}^T \left(u_{i,j} - (u_0)_{i,j}\right) + \gamma c_{i,j} \;. \end{aligned} \tag{5.50}$$

For optimization we transfer this minimization problem into the saddle-point formulation

$$\begin{aligned} \min_{u\in Y, c\in X} \max_{p\in Z, q\in Y} &\langle \nabla u, p\rangle_Z + \langle \nabla c, q\rangle_Y \\ &+ \lambda\, ||\rho(u,c)||_1 - \frac{\varepsilon\, ||p||_2^2}{2} - \delta_P(p) - \frac{\varepsilon\, ||q||_2^2}{2} - \delta_Q(q) \end{aligned} \tag{5.51}$$

with $q \in Y$ being the dual variable of c. With that we obtain the updates

$$\begin{aligned} p^{n+1} &= \operatorname{prox}_P\left(\frac{p^n + \sigma\nabla\overline{u}^n}{1+\sigma\varepsilon}\right) \\ q^{n+1} &= \operatorname{prox}_P\left(\frac{q^n + \sigma\nabla\overline{c}^n}{1+\sigma\varepsilon}\right) \\ u^{n+1} &= \operatorname{shrink}\left(u^n - \tau \operatorname{div} p^{n+1}\right) \\ c^{n+1} &= \operatorname{shrink}\left(c^n - \tau \operatorname{div} q^{n+1}\right) \\ \overline{u}^{n+1} &= 2u^{n+1} - u^n \\ \overline{c}^{n+1} &= 2c^{n+1} - c^n \;. \end{aligned} \tag{5.52}$$

As in (5.11), the resolvent operator $(p,q) = (I + \sigma\,\partial F^*)^{-1}(\hat{p},\hat{q})$ is a pointwise projection onto ℓ_2 unit balls. To simplify the notation in the thresholding checks, we define

$$r_{i,j} = \frac{\rho(\hat{u}_{i,j}, \hat{c}_{i,j})}{|\nabla I|_{i,j}^2 + \gamma^2} \;.$$

The modified thresholding checks and updates from Table 4.2 are presented in Table 5.2.

In Figure 5.11 and 5.12 we show two exemplar results (`Army` and `Schefflera` dataset from the Middlebury benchmark). In the cropped regions we show the benefits when adding a compensation variable in areas where the constancy assumption of the OFC is not satisfied. Most likely the illumination situation has changed and for the shown examples,

moving shadows are one of the major causes for not complying the brightness constancy over time. For the `Schefflera` dataset difficulties when matching small textured structure are absorbed by the added compensation. Still, for this dataset the emerging results contain outliers and in the cropped area the moving shadow is not entirely handled.

Table 5.2.: Thresholding checks and corresponding updates for solving the resolvent operator for the ℓ_1 data term with an additional compensation variable.

condition	thresholding check	update
$\rho(u,c) > 0$	$r_{i,j} < -\tau\lambda \left(\nabla I_1\right)_{i,j}$	$u_{i,j} = \hat{u}_{i,j} + \tau\lambda \left(\nabla I_1\right)_{i,j}$ $c_{i,j} = \hat{c}_{i,j} + \tau\lambda\gamma$
$\rho(u,c) < 0$	$r_{i,j} > \tau\lambda \left(\nabla I_1\right)_{i,j}$	$u_{i,j} = \hat{u}_{i,j} - \tau\lambda \left(\nabla I_1\right)_{i,j}$ $c_{i,j} = \hat{c}_{i,j} - \tau\lambda\gamma$
else	$\lvert r_{i,j}\rvert \leq \tau\lambda \left(\nabla I_1\right)_{i,j}$	$u_{i,j} = \hat{u}_{i,j} - r_{i,j} \left(\nabla I_1\right)_{i,j}$ $c_{i,j} = \hat{c}_{i,j} - r_{i,j}\gamma$

5.2.3. Optical Flow Constraint Based on Gradient Constancy

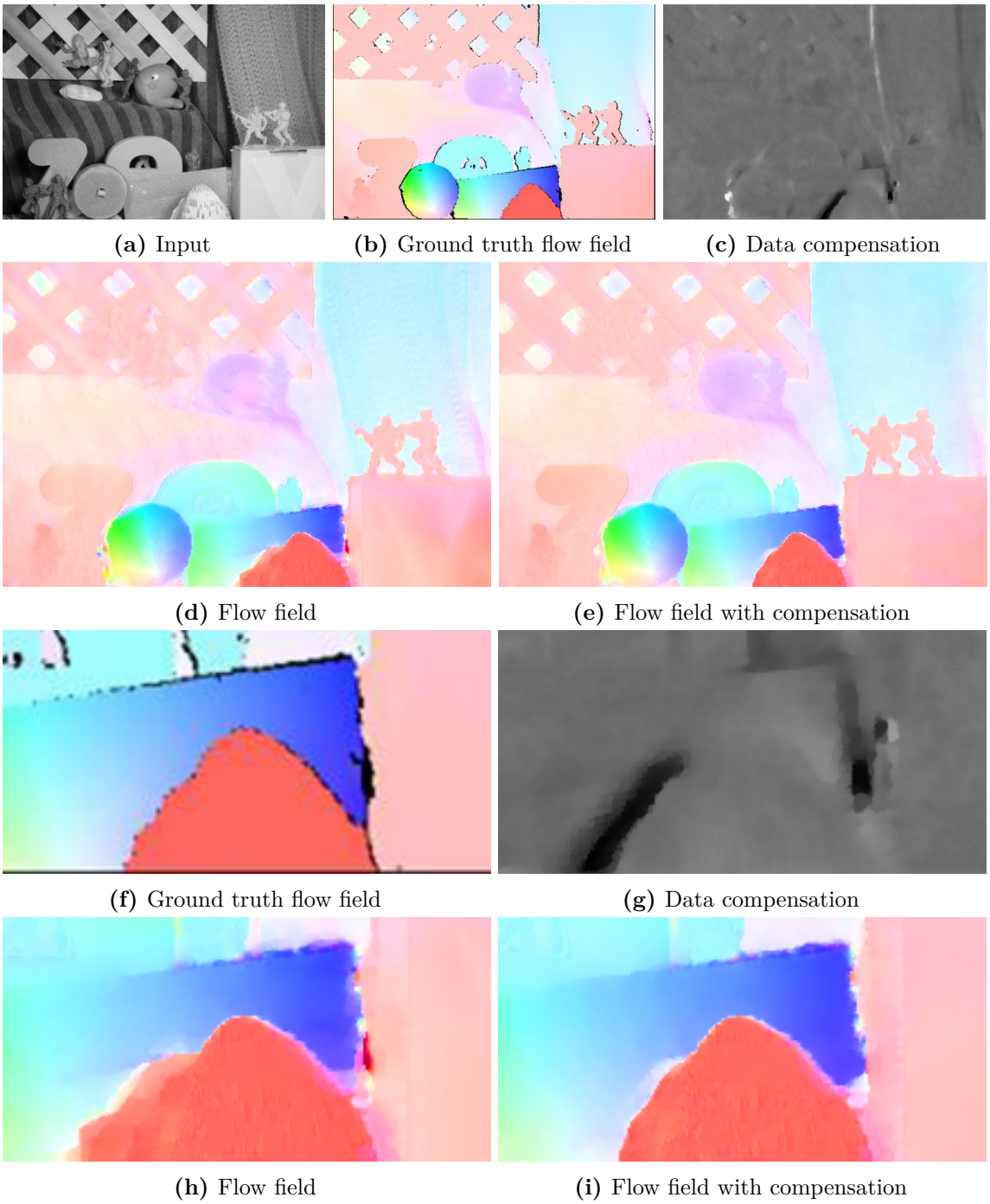
The remaining difficulties in handling brightness variations brings us to the next approach of modifying the OFC. In literature there are different approaches on how to exchange the brightness constancy for higher-order constancy assumptions. The related approaches have been discussed in Section 4.3.2. In the following, we present some insights on how to integrate a higher order constancy assumption and in particular on how to incorporate the constancy of moving gradients into the presented variational framework. At first, we will show how to integrate simple gradient constancy and then we give some details on how to optimize even more complex models with the preconditioned primal-dual algorithm (3.56) of Pock and Chambolle [2011] (*cf.* Section 3.6.5).

For a simple usage of gradient constancy we interchange the brightness constancy by assuming moving gradients to be constant. This results in a data term similar to (4.19) proposed by Brox et al. [2004] but neglecting the quota of the brightness constancy. With the gradient operator $\nabla = \left(\nabla^{x^1}, \nabla^{x^2}\right)^T$ we define the gradient field of the input intensity function as

$$\nabla I(x(t),t) = \begin{bmatrix} \nabla^{x^1} I(x(t),t) \\ \nabla^{x^2} I(x(t),t) \end{bmatrix} = \begin{bmatrix} I^{x^1} \\ I^{x^2} \end{bmatrix}.$$

Assuming gradient constancy over time yields the data term

$$\mathcal{D}(u) = \int_\Omega \underbrace{\left| I_t^{x^1} + \left(\nabla I^{x^1}\right)^T \left(u\left(x\right) - u_0\left(x\right)\right)\right|}_{\rho_1} + \underbrace{\left| I_t^{x^2} + \left(\nabla I^{x^2}\right)^T \left(u\left(x\right) - u_0\left(x\right)\right)\right|}_{\rho_2} \tag{5.53}$$



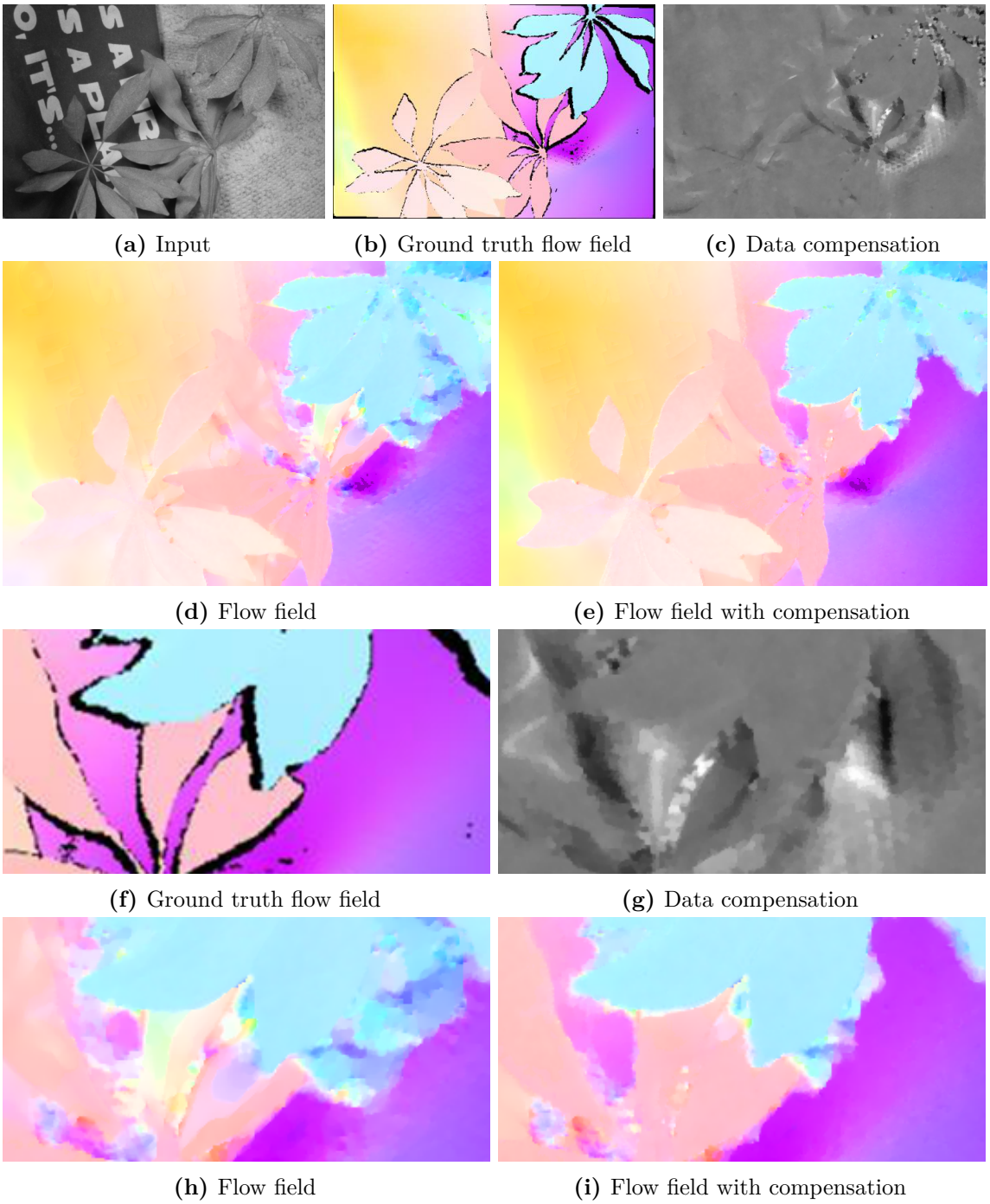
(a) Input **(b)** Ground truth flow field **(c)** Data compensation

(d) Flow field **(e)** Flow field with compensation

(f) Ground truth flow field **(g)** Data compensation

(h) Flow field **(i)** Flow field with compensation

Figure 5.11.: Effects of adding a compensation variable to the OFC on the `Army/Middlebury` dataset. The cropped area in (f-i) demonstrates the enhancement for moving shadows.



(a) Input **(b)** Ground truth flow field **(c)** Data compensation

(d) Flow field **(e)** Flow field with compensation

(f) Ground truth flow field **(g)** Data compensation

(h) Flow field **(i)** Flow field with compensation

Figure 5.12.: Effects of adding a compensation variable to the OFC on the `Schefflera/Middlebury` dataset. The cropped area in (f-i) demonstrates the enhancement for moving shadows and for wrong data matches due to the movement of small textures. This example shows that the compensation enhances the results but still the result is by far not perfect.

and the discretized minimization problem

$$\begin{aligned} &\min_{u\in Y} ||\nabla u||_{\varepsilon} + \lambda \left(||\rho_1(u)||_1 + ||\rho_2(u)||_1 \right) , \\ &\quad \text{with} \quad \rho_1(u_{i,j}) = (I_t^{x^1})_{i,j} + (\nabla I^{x^1})_{i,j}^T \left(u_{i,j} - (u_0)_{i,j} \right) , \\ &\quad \text{and} \quad \rho_2(u_{i,j}) = (I_t^{x^2})_{i,j} + (\nabla I^{x^2})_{i,j}^T \left(u_{i,j} - (u_0)_{i,j} \right) . \end{aligned} \tag{5.54}$$

When comparing the minimization problem to the baseline model (5.2) the only change is concerning the function $G(u)$ (4.29) which changes to

$$G(u) = \lambda \, ||\rho_1(u)||_1 + \lambda \, ||\rho_2(u)||_1 \, . \tag{5.55}$$

Hence, the difference for the algorithms iterates (5.12) is that the soft-thresholding scheme changes due to the resolvent operator with respect to $G(u)$. A similar approach on using two data terms is discussed in Wedel et al. [2008]. In the following we present a different approach how to derive the cases for the thresholding scheme. Therefore, we come back to the resolvent operator that is defined as the minimizer of

$$u = \arg\min_u \left\{ \frac{1}{2\tau} \, ||u - \hat{u}||_2^2 + \lambda \, ||\rho_1(u)||_1 + \lambda \, ||\rho_2(u)||_1 \right\} . \tag{5.56}$$

For convenience we define the data residuals as

$$\rho_1(u) = a^T u + b \qquad \text{and} \qquad \rho_2(u) = c^T u + d \, ,$$

and accordingly, the corresponding optimality condition yields

$$u - \hat{u} + \tau\lambda \, (s + t) = 0 \, . \tag{5.57}$$

Here, s and t are defined as the subgradient of $\rho_1(u)$ and $\rho_2(u)$ as

$$s \in \partial \left| a^T u + b \right| = \begin{cases} a & \text{if} \quad a^T u + b > 0 \\ -a & \text{if} \quad a^T u + b < 0 \\ [-a, a] & \text{else} \end{cases} \qquad \text{and} \tag{5.58}$$

$$t \in \partial \left| c^T u + d \right| = \begin{cases} c & \text{if} \quad c^T u + d > 0 \\ -c & \text{if} \quad c^T u + d < 0 \\ [-c, c] & \text{else} \end{cases} \, . \tag{5.59}$$

Similar to the derivation of the thresholding scheme in Section 4.4.2 different cases of the OFC's values must be made:

$\boldsymbol{\rho_1(u) > 0}$, $\boldsymbol{\rho_2(u) > 0}$: We use this case to represent the derivation of update rules for

the cases when

$$\begin{aligned} &\rho_1(u) > 0,\ \rho_2(u) > 0\,, \\ &\rho_1(u) > 0,\ \rho_2(u) < 0\,, \\ &\rho_1(u) < 0,\ \rho_2(u) > 0\,, \\ &\rho_1(u) < 0,\ \rho_2(u) < 0\,. \end{aligned} \tag{5.60}$$

All these cases deduce in a similar way and mainly only the signs of s and t changes as listed in the cases in (5.58) and (5.59). In the case of positive residuals $\rho_1(u)$ and $\rho_2(u)$, both signs in (5.57) are positive and this leads directly to the update rule

$$u = \hat{u} - \tau\lambda\,(a + c)\ . \tag{5.61}$$

To obtain the thresholding check with respect to $\hat{u}$ this update is substituted into the initial conditions $\rho_1(u) > 0$ and $\rho_2(u) > 0$ that in turn leads to

$$\begin{aligned} &\rho_1\left(\hat{u}\right) > \tau\lambda\left(a^T a - a^T c\right)\ , \text{ and} \\ &\rho_2\left(\hat{u}\right) > \tau\lambda\left(c^T a - c^T c\right) \end{aligned} \tag{5.62}$$

The related update rules and thresholding checks for the remaining cases in (5.60) are listed in Table 5.3.

$\boldsymbol{\rho_1(u) = 0,\ \rho_2(u) \gtrless 0}$: Here, the resolvent operator is defined as the minimizer of

$$u = \arg\min_u \left\{ \frac{1}{2\tau}\,||u - \hat{u}||_2^2 + \lambda\,\langle \mu_1, \rho_1(u)\rangle + \lambda\,||\rho_2(u)||_1 \right\}\ , \tag{5.63}$$

with μ_1 being the Lagrange multiplier modeling $\rho_1(u) = 0$. The optimality condition with respect to u and μ_1 yield

$$\frac{\partial}{\partial u}: \qquad u - \hat{u} + \tau\lambda\,(a\mu_1 \pm c) = 0 \tag{5.64}$$

$$\frac{\partial}{\partial \mu_1}: \qquad a^T u + b = 0 \tag{5.65}$$

By inserting u from (5.64) into (5.65) μ_1 becomes

$$\mu_1 = \frac{1}{a^T a}\left(\frac{1}{\tau\lambda}\rho_1 u \mp a^T c\right)\ . \tag{5.66}$$

By inserting μ_1 into (5.64) we get to the update rule

$$u = \hat{u} - \rho_1 u \frac{a}{a^T a} \pm \tau\lambda\left(a\frac{a^T c}{a^T a} - c\right)\ . \tag{5.67}$$

Transferring the conditions $\rho_2(u) \gtrless 0$ with respect to $\hat{u}$ the thresholding cases yield

$$\rho_2(\hat{u}) \gtrless \rho_1(\hat{u}) \pm \tau\lambda \left(-\frac{\left(c^T a\right)^2}{a^T a} + c^T c \right) \tag{5.68}$$

Note that for the case when $\rho_1(u) \gtrless 0,\ \rho_2(u) = 0$ the updates and thresholding cases are achieved in a similar way and are listed in Table 5.3.

$\boldsymbol{\rho_1(u) = u,\ \rho_2(u) = 0}$: Consiquently, we add two Lagrange multipliers μ_1 and μ_2 to model the equality constraint of the residuals in the minimization problem to obtain the resolvent operator for this case:

$$u = \arg\min_u \left\{ \frac{1}{2\tau} ||u - \hat{u}||_2^2 + \lambda \langle \mu_1, \rho_1(u) \rangle + \lambda \langle \mu_2, \rho_2(u) \rangle \right\} \tag{5.69}$$

The respective optimality condition yields

$$\frac{\partial}{\partial u}: \qquad u - \hat{u} + \tau\lambda (a\mu_1 + c\mu_2) = 0 \tag{5.70}$$

$$\frac{\partial}{\partial \mu_1}: \qquad a^T u + b = 0 \tag{5.71}$$

$$\frac{\partial}{\partial \mu_2}: \qquad c^T u + d = 0 \tag{5.72}$$

are used to obtain the update equation. By inserting u, expressed from (5.70), into (5.71) and (5.72) we obtain two linear equations with the two unknowns μ_1 and μ_2. Solving this system of equation for μ_1 and μ_2 and re-inserting them into (5.70) yields the update equation

$$u = \hat{u} + \frac{1}{a^T a c^T c - \left(a^T c\right)^2} \left(-a \left(\rho_1(\hat{u})\, c^T c - \rho_2(\hat{u})\, a^T c \right) + c \left(\rho_1(\hat{u})\, c^T a - \rho_2(\hat{u})\, a^T a \right) \right). \tag{5.73}$$

Hence, for optimizing (5.54) we apply the iterates (5.12) with the soft-thresholding scheme presented in Table 5.3. Still, examining the complex result for the soft-thresholding scheme when only considering two data terms, it becomes clear that for more than two data terms the solution to the resolvent operator for $G(u)$ will get even more complicated. An alternative approach on directly solving the resolvent operator is to dualize the data term. A point that must be considered then is the strength of the updates by adapting the stepsizes τ and σ accordingly. Recently, a variant of the used primal-dual algorithm has been proposed by Pock and Chambolle [2011] where a preconditioner can be utilized to balance the strength of the corresponding updates. In the following we utilize this approach to show how to optimize for multiple data terms.

Table 5.3.: Updates and corresponding thresholding checks for the resolvent operator when dealing with two data terms.

condition	thresholding check	update
$\rho_1(u) > 0$ $\rho_2(u) > 0$	$\rho_1(\hat{u}) > \tau\lambda\left(a^Ta + a^Tc\right)$ $\rho_2(\hat{u}) > \tau\lambda\left(c^Ta + c^Tc\right)$	$u = \hat{u} - \tau\lambda\left(a + c\right)$
$\rho_1(u) > 0$ $\rho_2(u) < 0$	$\rho_1(\hat{u}) > \tau\lambda\left(a^Ta - a^Tc\right)$ $\rho_2(\hat{u}) < \tau\lambda\left(c^Ta - c^Tc\right)$	$u = \hat{u} - \tau\lambda\left(a - c\right)$
$\rho_1(u) < 0$ $\rho_2(u) > 0$	$\rho_1(\hat{u}) < \tau\lambda\left(-a^Ta + a^Tc\right)$ $\rho_2(\hat{u}) > \tau\lambda\left(-c^Ta + c^Tc\right)$	$u = \hat{u} - \tau\lambda\left(-a + c\right)$
$\rho_1(u) < 0$ $\rho_2(u) < 0$	$\rho_1(\hat{u}) < \tau\lambda\left(-a^Ta - a^Tc\right)$ $\rho_2(\hat{u}) < \tau\lambda\left(-c^Ta - c^Tc\right)$	$u = \hat{u} - \tau\lambda\left(-a - c\right)$
$\rho_1(u) = 0$ $\rho_2(u) > 0$	$\tau\lambda\left(a^Ta + a^Tc\right) \geq \rho_1(\hat{u}) \geq \tau\lambda\left(-a^Ta + a^Tc\right)$ $\rho_2\left(\hat{u}\right) > \rho_1\left(\hat{u}\right)\frac{c^Ta}{a^Ta} + \tau\lambda\left(-\frac{\left(c^Ta\right)^2}{a^Ta} + c^Tc\right)$	$u = \hat{u} - \rho_1\left(\hat{u}\right)\frac{a}{a^Ta} + \tau\lambda\left(a\frac{a^Tc}{a^Ta} - c\right)$
$\rho_1(u) = 0$ $\rho_2(u) < 0$	$\tau\lambda\left(a^Ta - a^Tc\right) \geq \rho_1(\hat{u}) \geq \tau\lambda\left(-a^Ta - a^Tc\right)$ $\rho_2\left(\hat{u}\right) < \rho_1\left(\hat{u}\right)\frac{c^Ta}{a^Ta} - \tau\lambda\left(-\frac{\left(c^Ta\right)^2}{a^Ta} + c^Tc\right)$	$u = \hat{u} - \rho_1\left(\hat{u}\right)\frac{a}{a^Ta} - \tau\lambda\left(a\frac{a^Tc}{a^Ta} - c\right)$
$\rho_1(u) > 0$ $\rho_2(u) = 0$	$\rho_1\left(\hat{u}\right) > \rho_2\left(\hat{u}\right)\frac{a^Tc}{c^Tc} + \tau\lambda\left(a^Ta - \frac{\left(c^Ta\right)^2}{c^Tc}\right)$ $\tau\lambda\left(c^Ta + c^Tc\right) \geq \rho_2(\hat{u}) \geq \tau\lambda\left(c^Ta - c^Tc\right)$	$u = \hat{u} - \rho_2\left(\hat{u}\right)\frac{c}{c^Tc} + \tau\lambda\left(-a + c\frac{c^Ta}{c^Tc}\right)$
$\rho_1(u) < 0$ $\rho_2(u) = 0$	$\rho_1\left(\hat{u}\right) < \rho_2\left(\hat{u}\right)\frac{a^Tc}{c^Tc} - \tau\lambda\left(a^Ta - \frac{\left(c^Ta\right)^2}{c^Tc}\right)$ $\tau\lambda\left(-c^Ta + c^Tc\right) \geq \rho_2(\hat{u}) \geq \tau\lambda\left(-c^Ta - c^Tc\right)$	$u = \hat{u} - \rho_2\left(\hat{u}\right)\frac{c}{c^Tc} - \tau\lambda\left(-a + c\frac{c^Ta}{c^Tc}\right)$
else		$u = \hat{u} + \frac{1}{a^Tac^Tc - \left(a^Tc\right)^2}\left(-a\left(\rho_1\left(\hat{u}\right)c^Tc - \rho_2\left(\hat{u}\right)a^Tc\right) + c\left(\rho_1\left(\hat{u}\right)c^Ta - \rho_2\left(\hat{u}\right)a^Ta\right)\right)$

5.2.4. Increasing the Data Information – Multiple Dataterms

We have seen in the previous section (*cf*. Table 5.3) that already for two data terms the direct solution of the resolvent operator with respect to the function $G(u)$ is cumbersome to compute and this gets worse for an increasing number of data terms. In the following we want to consider the general case for K data terms. Let us therefore consider the case where we dualize not only the ℓ_1 norm of the smoothness term but also for the data terms. Introducing the dual variables $q = (q_1, \dots, q_K)$ with $q_k \in X$ the primal-dual saddle-point formulation yields

$$\min_{u\in Y} \max_{p\in Z, q\in X} \langle \nabla u, p\rangle_Z - \frac{\varepsilon \, ||p||_2^2}{2} - \delta_P(p) + \lambda \sum_{k=1}^{K} \left(\langle \rho_k(u), q_k\rangle_X - \delta_{\{|q_1|_\infty \le 1\}} \right) . \tag{5.74}$$

With the definition of the diagonal preconditioner matrices elements (3.58) we end up with the iterates

$$\begin{aligned} p^{n+1} &= \operatorname{prox}_P \left(\frac{p^n + \Sigma \nabla \overline{u}^n}{1 + \Sigma\varepsilon} \right) \\ q_k^{n+1} &= \operatorname{prox}_{[-1,1]} \left(q_k^n + \Sigma\lambda \rho_k(\overline{u}^n) \right) \\ u^{n+1} &= \operatorname{shrink} \left(u^n - T \left(\operatorname{div} p^{n+1} + \lambda \sum_{k=1}^{K} \left(q_k^{n+1} \frac{\partial}{\partial u} q_k^{n+1} \right) \right) \right) \\ \overline{u}^{n+1} &= 2u^{n+1} - u^n , \end{aligned} \tag{5.75}$$

with the pointwise truncation $\operatorname{prox}_{[-1,1]}$. **[TODO: i think the stuff with the sum and the partial is not correct. recheck!]**

For better understanding of the preconditioner matrices elements we apply the algorithm to optimize a model using a data term with gradient-based constancy assumption discussed in Section 5.2.3. By dualizing the Huber-norm of the smoothness term and the data terms ℓ_1-norms, (5.54) transfers to the saddle-point formulation

$$\begin{aligned} \min_{u\in Y} \max_{p\in Z, q\in X} \langle \nabla u, p\rangle_Z - \frac{\varepsilon \, ||p||_2^2}{2} - \delta_P(p) \\ + \lambda \left(\langle \rho_1(u), q_1\rangle_X - \delta_{\{|q_1|_\infty \le 1\}} + \langle \rho_2(u), q_2\rangle_X - \delta_{\{|q_2|_\infty \le 1\}} \right) . \end{aligned} \tag{5.76}$$

In order to compute the elements of the preconditioner matrix, we bring (5.76) in the form

$$\left\langle Ku, (p, q_1, q_2)^T \right\rangle + \left\langle b, (q_1, q_2)^T \right\rangle . \tag{5.77}$$

In the following we deduce the corresponding operator matrix K for which we will use the operators $\nabla^{x^1}, \nabla^{x^2}$ for the x- and y-gradient operators which are of size $MN \times MN$.

With that we define the matrices

$$K = \begin{bmatrix} \nabla^{x^1} & \\ \nabla^{x^2} & \\ & \nabla^{x^1} \\ & \nabla^{x^2} \\ \operatorname{diag}(\lambda \nabla^{x^1} I^{x^1}) & \operatorname{diag}(\lambda \nabla^{x^2} I^{x^1}) \\ \operatorname{diag}(\lambda \nabla^{x^1} I^{x^2}) & \operatorname{diag}(\lambda \nabla^{x^2} I^{x^2}) \end{bmatrix}, \tag{5.78}$$

and

$$b = \begin{bmatrix} \lambda \nabla^{x^1} I^{x^1} + \lambda \nabla^{x^2} I^{x^1}) \\ \lambda \nabla^{x^1} I^{x^2} + \lambda \nabla^{x^2} I^{x^2}) \end{bmatrix}. \tag{5.79}$$

Computing the elements of the preconditioner matrices according to (3.58) is trivial by considering the row-wise sum of the elements in (5.78) for computing Σ. Respectively, for T use the column-wise sum of K. For convenience we furthermore define

$$c = \begin{bmatrix} \lambda q_1 \nabla^{x^1} I^{x^1} + q_2 \lambda \nabla^{x^1} I^{x^2}) \\ \lambda q_1 \nabla^{x^2} I^{x^1} + q_2 \lambda \nabla^{x^2} I^{x^2}) \end{bmatrix}.$$

With that, the iterates for optimizing the Huber-ℓ_1 model with gradient constancy data term yield

$$\begin{aligned} p^{n+1} &= \operatorname{prox}_P \left(\frac{p^n + \Sigma \nabla \overline{u}^n}{1 + \Sigma \varepsilon} \right) \\ q_1^{n+1} &= \operatorname{prox}_{[-1,1]} \left(q_1^n + \Sigma \lambda \rho_1(\overline{u}^n) \right) \\ q_2^{n+1} &= \operatorname{prox}_{[-1,1]} \left(q_2^n + \Sigma \lambda \rho_2(\overline{u}^n) \right) \\ u^{n+1} &= \operatorname{shrink} \left(u^n - T \left(\operatorname{div} p^{n+1} + c \right) \right) \\ \overline{u}^{n+1} &= 2u^{n+1} - u^n, \end{aligned} \tag{5.80}$$

5.2.5. Quadratic Fitting on the Dataterms Energy Functional

Standard optical flow algorithms often use some kind of constancy assumption for computing similarity measures. In order to optimize for the disparities with a formulation like (4.22), one has to linearize the image function around a given point $u_0(x)$ to obtain an objective function that is linear in $u(x)$ and therefore convex (*cf.* Section 4.2.1). This restricts the data fidelity modeling to rather simple constraints as *e.g.* the brightness or gradient constancy assumption. We introduced an alternative approach in [Werlberger et al., 2010], where not the image function is linearized but a Taylor expansion is directly applied to

the data terms energy functional to linearize it. The major benefit of this approach is its flexibility to add arbitrarily complex data terms. This enables us to use window-based approaches like *e.g.* sum of squared/absolute differences, correlation-based similarities or the Census transform. In the following we discuss the second-order Taylor approximation of the data term and its optimization in the primal-dual framework. In addition we present some relevant similarity measures that work well for different applications of optical flow.

Therefore, let us first define a general data term function

$$\phi\left(x, u\left(x\right)\right) = \delta\left(x, I_1\left(x\right), I_2\left(x\right), u\left(x\right)\right) , \tag{5.81}$$

where $\delta\left(x, I_1\left(x\right), I_2\left(x\right), u\left(x\right)\right)$ defines an arbitrary similarity measure of the input image $I_1\left(x\right)$ (fixed image) and the transformed image $I_2\left(x\right)$ (moving image). $u\left(x\right)$ again denotes the flow vectors from $I_1\left(x\right)$ to $I_2\left(x\right)$. Let us ignore the details of the similarity measure for the moment. Applying a Taylor expansion to (5.81), neglecting terms higher than second order, yields the approximation

$$\begin{aligned}\breve{\phi}(x, u\left(x\right)) \approx \phi(x, u_0\left(x\right)) + \left(\nabla\phi(x, u_0\left(x\right))\right)^T \left(u\left(x\right) - u_0\left(x\right)\right) + \\ \frac{1}{2}\left(u\left(x\right) - u_0\left(x\right)\right)^T \nabla^2\phi(x, u_0\left(x\right)) \left(u\left(x\right) - u_0\left(x\right)\right) .\end{aligned} \tag{5.82}$$

For brevity we will omit the argument x for the following considerations and use the notation $\breve{\phi}$, $\phi(u)$ and $\phi(u_0)$ respectively. In (5.82), $\nabla\phi(u_0) = (\phi_x(u_0), \phi_y(u_0))$ denotes the function's gradient, whereas $\nabla^2\phi(u_0)$ is its Hessian matrix defined as

$$\nabla^2\phi(u_0) = \mathcal{H}\left(\phi\left(u_0\right)\right) = \begin{bmatrix} \phi_{xx}(u_0) & \phi_{xy}(u_0) \\ \phi_{yx}(u_0) & \phi_{yy}(u_0) \end{bmatrix} . \tag{5.83}$$

To guarantee convexity of the data term's approximation the Hessian matrix must be positive semidefinite. This is the case when both Eigenvalues of (5.83) are non-negative. In the following we demonstrate two different approaches to handle violations of this requirement. At first we describe how to check the positive semidefiniteness via the computation of the Eigenvalues. Further, we transform the Hessian into a positive semidefinite matrix if an Eigenvalue is negative. Secondly, we show a simple approximation of the Hessian to guarantee convexity.

5.2.5.1. Positive Definiteness of the Hessian Matrix

In the following we show two approaches that guarantee for convexity of the used second-order Taylor approximation of the data term. Therefore, a sufficient condition is the positive definiteness of the Hessian matrix which is fulfilled when the two Eigenvalues are positive. At first we show the computation of the Eigenvalues and Eigenvectors and how to transform the Hessian matrix into a positive definite matrix if the condition of positive Eigenvalues does not hold. We will refer to this as the "full approximation of the

Hessian" in the following. Furthermore, we will discuss a second approach that uses a simple approximation of the Hessian that is positive definite by design. We will see that the approximation is simpler to compute and above all the results are similar to the full approximation of the Hessian.

Full Approximation of the Hessian Matrix

To compute the Eigenvalues of (5.83) we consider the characteristic equation

$$\det\left(\mathcal{H}(\phi(u_0)) - \lambda I\right) = 0\,. \tag{5.84}$$

Its solution are the Eigenvalues. For the given symmetric 2×2 matrix $\mathcal{H}(\phi(u_0))$ this yields

$$\det\begin{pmatrix} \phi_{xx}(u_0) - \lambda & \phi_{xy}(u_0) \\ \phi_{xy}(u_0) & \phi_{xx}(u_0) - \lambda \end{pmatrix} = 0\,. \tag{5.85}$$

Solving the resultant quadratic equation reveals the Eigenvalues

$$\lambda_{1/2} = \frac{1}{2}\left(\left(\phi_{xx}(u_0) + \phi_{yy}(u_0)\right) \pm \sqrt{4\left(\phi_{xy}(u_0)\right)^2 + \left(\phi_{xx}(u_0) - \phi_{yy}(u_0)\right)^2}\right)\,. \tag{5.86}$$

Furthermore, the corresponding Eigenvectors $v_{1/2}$ are computed through the identity

$$\mathcal{H}(\phi(u_0))v = \lambda v\,, \quad \text{yielding} \qquad \begin{pmatrix} \phi_{xx}(u_0) - \lambda & \phi_{xy}(u_0) \\ \phi_{xy}(u_0) & \phi_{yy}(u_0) - \lambda \end{pmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}\,. \tag{5.87}$$

With that we can compute the slope of the first Eigenvalue $v_1 = (v_1^1, v_1^2)$ through the component's relations

$$\begin{aligned} \left(\phi_{xx}(u_0) - \lambda_1\right) v_1^1 + \phi_{xy}(u_0) v_1^2 = 0 \quad &\rightarrow \quad \frac{v_1^1}{v_1^2} = \frac{\phi_{xy}(u_0)}{\phi_{xx}(u_0) - \lambda_1} \quad \text{and} \\ \phi_{xy}(u_0) v_1^1 + \left(\lambda_1 - \phi_{xx}(u_0)\right) v_1^2 = 0 \quad &\rightarrow \quad \frac{v_1^1}{v_1^2} = \frac{\lambda_1 - \phi_{yy}(u_0)}{\phi_{xy}(u_0)}\,. \end{aligned} \tag{5.88}$$

Hence, for $\phi_{xy}(u_0) \neq 0$ the Eigenvector normalized to the unit length yields

$$v_1 = \left(\frac{\phi_{xy}(u_0)}{\sqrt{\left(\phi_{xy}\right)^2 + \left(\lambda - \phi_{xx}\right)^2}}, \frac{\lambda_1 - \phi_{xx}(u_0)}{\sqrt{\left(\phi_{xy}\right)^2 + \left(\lambda - \phi_{xx}\right)^2}},\right)^T\,. \tag{5.89}$$

For the symmetric 2×2 matrix $\mathcal{H}(\phi(u_0))$, the Eigenvectors are orthogonal which yields

$$v_2 = \left(v_1^2, -v_1^1\right)^T\,. \tag{5.90}$$

In case when $\phi_{xy}(u_0) = 0$, the Eigenvectors reduce to $v_1 = (1, 0)$, and $v_2 = (0, 1)$.

To guarantee the Hessian to be positive semidefinite, we restrict the Eigenvalues to be positive with

$$\lambda_1^+ = \max\left\{0, \lambda_1\right\}, \qquad \text{and} \qquad \lambda_2^+ = \max\left\{0, \lambda_2\right\},$$

and use those to reconstruct the matrix $\mathcal{H}^+(\phi(u_0))$ with the elements

$$\begin{aligned} \phi_{xx}^+ &= \lambda_1^+ (v_1^1)^2 + \lambda_2^+ (v_2^1)^2 \\ \phi_{yy}^+ &= \lambda_1^+ (v_1^2)^2 + \lambda_2^+ (v_2^2)^2 \\ \phi_{xy}^+ &= \lambda_1^+ v_1^1 v_1^2 + \lambda_2^+ v_2^1 v_2^2 \,. \end{aligned} \tag{5.91}$$

It can be shown that $\mathcal{H}^+(\phi(u_0))$ is the orthogonal projection to the set of symmetric positive semidefinite 2×2 matrices.

Approximation of the Hessian Matrix

The previous approach guarantees for convex approximations of the data term by computing the Eigenvalues and in case of non-positive Eigenvalues, we approximate the Hessian by a reasonable positive definite matrix. As an alternative approach we propose to use only positive second order derivatives and set the mixed derivative $\phi_{xy}(u_0) = 0$. By design, this yields the positive definite approximation of the Hessian

$$\mathcal{H}^+(\phi(u_0)) = \begin{bmatrix} \phi_{xx}^+(u_0) & 0 \\ 0 & \phi_{yy}^+(u_0) \end{bmatrix}. \tag{5.92}$$

Discussion

We anticipate some of the results before discussing the optimization routine and evaluate both approximations on the Middlebury benchmark database. The aim is to show the differences of the two approximations in terms of accuracy. With that, we can check if the additional costs for projecting the Hessian to the set of symmetric positive semidefinite 2×2 matrices pay of or if the simple approximation is sufficient. For computing the similarity measure we use pointwise absolute differences which corresponds to the same as when using the standard optical flow constraint based on intensities. In Table 5.4 the two approaches are evaluated on the Middlebury's benchmark sequences where a ground truth is available. Figure 5.13 shows an exemplar result and affirms that the computed flow fields are equal, independent on the used approximation. There, even the simpler approximation is a little bit in favor. Hence, we can use the approach of only using positive second order derivatives to construct the Hessian matrix without any limitations.

5.2.5.2. The Similarity Measure

The major benefit of this model is the generality of using arbitrary data terms. In the following we will give a brief overview of some possibilities that turn out to work well in

Table 5.4.: Comparison of the Hessian's full approximation with the faster/simpler approach of only using the positive diagonal entries of the Hessian matrix ($\mathcal{H}^+$). As error measurement the average end-point error of the flow vectors is used. (For details on the evaluation, see Section 5.3)

Approx.	Hydrangea		Venus		Grove3		Urban3		
	Dimetrodon		RubberWhale		Grove2		Urban2		Average
Full	0.209	0.239	0.196	0.449	0.247	0.713	0.457	0.623	0.392
$\mathcal{H}^+$	0.208	0.239	0.195	0.444	0.247	0.706	0.433	0.596	0.384

(a) Frame 10 **(b)** Ground truth **(c)** Full **(d)** $\mathcal{H}^+$

Figure 5.13.: Visual comparison of the results on Middlebury's `Urban2` sequence. Almost no differences are visible when comparing the color-coded visualization when using the full (c) or simple (d) approximation of the Hessian matrix. Both results are close to the ground truth flow field (b).

the context of optical flow estimation. Beside the classical pointwise data matching term, the already introduced classical optical flow constraint, comparing intensities or gradients the quadratic approximation enables us to use patch based similarity measures. Related to the intensity constancy assumption we can use the sum of absolute differences in a defined window (mostly a rectangular neighborhood around the current pixel).

Furthermore, also more complex similarity measures can be incorporated. In the field of image matching correlation based measurements like the normalized cross correlation (NCC) is well-established and with the presented approximation of the data term functional the integration is trivial. A major benefit is its invariance to multiplicative illumination changes which can be very useful in the field of motion estimation. To define a truncated variant of the NCC we first define the mean of a certain patch within the input images $I_1(x)$ and $I_2(x)$ as

$$\begin{aligned} \mu_1(x) &= \int_\Omega I_1(y) B_L(x-y)\,\mathrm{d}y\,, \quad \text{and} \\ \mu_2(x) &= \int_\Omega \widetilde{I}_2(y) B_L(x-y)\,\mathrm{d}y\,, \end{aligned} \tag{5.93}$$

where B_L denotes a box filter of width L and $\int B(z)\,\mathrm{d}z = 1$.

$$\widetilde{I}_2(y) = I_2(y + u_0)$$

depicts the warped image. Likewise, the standard deviation $\sigma(x)$ is defined as

$$\begin{aligned} \sigma_1(x) &= \sqrt{\int_\Omega \left(I_1(y) - \mu_1(x)\right)^2 B_L(x-y)\,\mathrm{d}y}\,, \quad \text{and} \\ \sigma_2(x) &= \sqrt{\int_\Omega \left(I_2(y) - \mu_2(x)\right)^2 B_L(x-y)\,\mathrm{d}y}\,. \end{aligned} \tag{5.94}$$

Hence, the pointwise matching costs using the normalized cross correlation yields

$$\begin{aligned} &\mathrm{NCC}(x, I_1(x), I_2(x), u(x)) = \\ &\qquad \frac{1}{\sigma_1(x)\,\sigma_2(x)} \int_\Omega (I_1(y) - \mu_1(x))(\widetilde{I}_2(y) - \mu_2(x)) B_L(x-y)\,\mathrm{d}y\,. \end{aligned} \tag{5.95}$$

Note that the matching term is computed based on the warped image $\widetilde{I}_2(y) = I_2(y + u_0)$ to better account for the local distortion within the correlation window. The values of the NCC is in the range $[-1, 1]$ and for modeling the actual similarity measure we incorporate a pointwise truncation and only consider positive values of (5.95). Neglecting poor correlations increases the robustness of the data term towards outliers that are *e.g.* caused by occlusions. This yields the similarity measure

$$\delta_{\mathrm{TNCC}}(x, I_1(x), I_2(x), u(x)) = \min\left\{1, 1 - \mathrm{NCC}(x, I_1(x), I_2(x), u(x))\right\} \tag{5.96}$$

and due to the truncation we call the matching function the truncated normalized cross correlation (TNCC). This is then introduced into the data term as

$$\phi(x, u(x)) = \delta_{\mathrm{TNCC}}(x, I_1(x), I_2(x), u(x))\,. \tag{5.97}$$

5.2.5.3. Minimization

Combining the Huber norm (5.1) for regularizing the flow field and the presented second order approximation of the data term (5.82) yields the minimization task

$$\min_{u \in Y} \int_\Omega |Du|_\varepsilon + \lambda \int_\Omega \breve{\phi}(x)\,\mathrm{d}x\,. \tag{5.98}$$

For minimization we define $\breve{\phi}_{i,j}$ to be the the discretized version of the approximated data term (5.82). With $||\cdot||_\varepsilon$ being the discretized Huber norm, the optimization problem (5.98) yields

$$\min_{u \in Y} ||\nabla u||_\varepsilon + \lambda \sum_{i,j} \breve{\phi}_{i,j}\,. \tag{5.99}$$

Accordingly, we obtain the convex-concave saddle-point formulation

$$\min_{u\in Y}\max_{p\in Z}\langle\nabla u,p\rangle_Z-\frac{\varepsilon\,||p||_2^2}{2}-\delta_P\left(p\right)+\lambda\sum_{i,j}\breve{\phi}_{i,j}\,. \tag{5.100}$$

To define the updated algorithm for this model, we have to treat the resolvent operator with respect to $G(u)=\lambda\sum_{i,j}\breve{\phi}_{i,j}$ that is defined as the minimizer

$$u=\arg\min_u\left\{\frac{1}{2\tau}\,||u-\hat{u}||_2^2+\lambda\breve{\phi}\right\}\,. \tag{5.101}$$

Its optimality conditions with respect to the flow fields components u_1,u_2 yield

$$\begin{aligned}\frac{\partial}{\partial u_1}:&\quad u_1-\hat{u}_1+\tau\lambda\left(\phi_x+\phi_{xx}\left(u_1-u_{1,0}\right)\right)=0\\ \frac{\partial}{\partial u_2}:&\quad u_2-\hat{u}_2+\tau\lambda\left(\phi_y+\phi_{yy}\left(u_2-u_{2,0}\right)\right)=0\end{aligned} \tag{5.102}$$

Hence, the pointwise minimizers of (5.101) for the components u_1,u_2 yield

$$u=\mathrm{prox}_{\mathrm{QF}}\left\{\hat{u}\right\}=\begin{cases}\frac{1}{\frac{1}{\tau}+\lambda\phi_{xx}(u_{1,0})}\left(\frac{\hat{u}_1}{\tau}-\lambda\phi_x+\lambda\phi_{xx}u_{1,0}\right)\\ \frac{1}{\frac{1}{\tau}+\lambda\phi_{yy}(u_{2,0})}\left(\frac{\hat{u}_2}{\tau}-\lambda\phi_y+\lambda\phi_{yy}u_{2,0}\right)\end{cases}\,. \tag{5.103}$$

With that we once more solve the primal-dual saddle-point problem by applying a gradient descent update on the primal variable and a gradient ascent on the dual. The iterates are given as

$$\begin{aligned}p^{n+1}&=\mathrm{prox}_P\left\{\frac{p^n+\sigma\nabla\overline{u}^n}{1+\sigma\varepsilon}\right\}\\ u^{n+1}&=\mathrm{prox}_{\mathrm{QF}}\left\{u^n-\tau\mathrm{div}\,p^{n+1}\right\}\\ \overline{u}^{n+1}&=2u^{n+1}-u^n\,.\end{aligned} \tag{5.104}$$

5.2.5.4. A Note on Multiple Data Terms

Throughout the whole section about the quadratic approximation of the data term we argued about its flexibility on using any arbitrary data term. In the following we round up this reasoning by explaining how to incorporate multiple data terms. This is advantageous when using *e.g.* a higher order constancy assumption or simply when using more than two input images. This also completes the approaches for multiple data terms beside the explicit solution of the resolvent operator by the soft-thresholding scheme presented in

Section 5.2.3 and the preconditioned variant in Section 5.2.4. For both approaches (in particular the soft-thresholding scheme) the computation can get cumbersome when having more than two data terms. With the approach presented in the following the extension to arbitrary frames is trivial.

When having K data terms, the individual similarity measures are defined as the sum

$$\phi(x, u\,(x)) = \sum_{k=1}^{K} \phi_k(x, u\,(x))\,. \tag{5.105}$$

When applying its second-order approximation, the derivatives of $\phi(x, u_0\,(x))$ are computed as sum of the gradients of the individual data terms:

$$\begin{aligned} \nabla\phi(x, u_0\,(x)) &= \sum_{k=1}^{K} \nabla\phi_k(x, u_0\,(x)) \\ \nabla^2\phi(x, u_0) &= \sum_{k=1}^{K} \nabla^2\phi_k(x, u_0\,(x))\,. \end{aligned} \tag{5.106}$$

The rest of the optimization remains the same.

5.2.5.5. Discussion

To demonstrate the flexibility of the presented data term's quadratic approximation we show some exemplar results using different similarity measures. Therefore, we utilize the tensor-directed Huber regularization as smoothness prior and different similarity measures to model the data fidelity. For a more extensive evaluation we refer to Section 5.3 where also other regularizers are used in conjunction with different similarity measures. As representative results we show three different similarity measures, absolute differences (AD) which is equatable with the classical OFC, the sum of absolute differences over a 3×3 patch (SAD), and the truncated normalized cross correlation (TNCC). For this comparison we manually tuned the λ parameter to obtain reasonable results for the specific data set. The corresponding values are given in the images' captions. While the flow fields computed with the pointwise AD similarity measure and the patch-based SAD do not show much of a difference, the result computed with the TNCC is able to capture much more details, especially for the shown `Schefflera` sequence. Especially in areas with lightning changes and strong texture the TNCC outreaches the other approaches. Furthermore, in combination with an even stronger regularizer (*e.g.* nonlocal Huber regularization) the performance can be increased even more.

In general the notion of having a flexible way of modeling the data term is affirmed by the results and possibilities how similarity measures can be integrated. Nevertheless we mention some comments about the drawback of the model. As the approximation is based on the energy functional, and not the image's intensity function, the model is in general more sensitive to disturbances. Therefore, the restriction of the update according to the

linearization (*cf*. Figure 4.4 in Section 4.4.3) is even more important than for the models using the classical OFC. This sensitivity particularly emerges when using a pointwise data matching cost. For path-based approaches and there especially for strong similarity measures as the TNCC, the linearization restriction becomes less important again. For using *e.g*. absolute differences as matching cost, we set the interval around the linearized point to $\Delta_{u_0} = 1.0$ and the factor for reducing the size of the interval to $\kappa_{u_0} = 1.75$. As mentioned for stronger data terms the interval can be increased to $\Delta_{u_0} = 2.0$ and $\kappa_{u_0} = 1.25$ without any problems but actually the effect is barely visible in the accuracy of the estimated flow field. Therefore we simply stick to the restricted update interval for all the used similarity measures.

(a) Frame 10 **(b)** Ground truth **(c)** AD, $\lambda = 25$

(d) SAD, $\lambda = 30$ **(e)** TNCC, $\lambda = 3$

(f) Frame 10 **(g)** Ground truth **(h)** AD, $\lambda = 25$

(i) SAD, $\lambda = 30$ **(j)** TNCC, $\lambda = 1$

Figure 5.14.: Exemplar results on the Middlebury's `Schefflera` and `Urban3` sequences. The results are computed with the tensor-directed Huber-Quadfit model with the (h) similarity measures absolute differences (AD), (i) sum of absolute differences (3×3 window) (SAD), and (j) the truncated normalized cross correlation (3×3 window) (TNCC).

5.3. Experimental Comparison

In this section we compare the different modifications of the TV-ℓ_1 model that have been presented throughout this chapter. Because of the models' different characteristics, the parameterization must be chosen differently for each evaluation. Especially the parameter λ, defining the trade-off between regularization and data fidelity, can have a strong influence on the resultant optical flow field. To obtain a suitable choice for λ we perform an extensive evaluation on the Middlebury's benchmark datasets where ground-truth data is available [Baker et al., 2011]. These eight sequences consist of four natural and four synthetic ones. For comparing the flow vectors u with some ground-truth flow vectors $\breve{u}$ two different error measures have been enforced in the past:

- Fleet and Jepson [1990] introduced the angular error (ae) that is defined as

$$\text{ae} = \arccos \frac{u^T \breve{u}}{||u||_2 \, ||\breve{u}||_2} \,, \text{ and} \tag{5.107}$$

- Otte and Nagel [1995] proposed to use the end-point error (epe) that is defined as the Euclidean distance between the flow vector's end-points:

$$\text{epe} = ||u - \breve{u}||_2 \,. \tag{5.108}$$

For our evaluation we will use the end-point error because the angular error penalizes vectors differently depending on their magnitude. As we want to penalize vectors with large magnitude the same way than vectors with small magnitudes we utilize the end-point error. Still, let us point out that the use of the angular error would lead to similar results for the upcoming evaluations. Another fact to mention is that the natural scenes have annotated occlusions where the error measure is not evaluated. Interestingly, the synthetic datasets, where a virtual camera moves along a rendered scene, have no marked occlusions. This raises the error for these datasets which is also visible in all our evaluations where the synthetic datasets always exhibit a higher error than the natural scenes.

In order to find a suitable choice for λ, we evaluate all the combinations of presented regularization and data terms on the benchmark sequences with varying λ. We then compute the average end-point error (aepe) of the resultant flow fields and in addition we average the error over all datasets to obtain a statement of the best λ over multiple sequences. The plots of these evaluations are shown in Appendix A. Furthermore, Table 5.5 lists the errors for the optimal λ and Figures 5.16-5.18 show an assortment of resultant color-coded flow fields.

The evaluation shows that the combination of regularization and data term is not trivial and actually is also dependent on the given data. As the datasets do not exhibit any strong lightning changes, hardly any noise and no outliers, we should not expect any big differences in the error rates. Still, the stability of the algorithms can be judged according to changing

parametrization. A steady error rate over varying λ is a good sign for robustness towards changing this parameter. Let us now comment some of the findings of this evaluation:

- Already the baseline model, using a Huber regularization and ℓ_1 penalized intensity constancy assumption, yields good results on this database.

- Using structure-texture decomposed input image increases the robustness against lightning changes but in turn also useful information is neglected by omitting data. This is observed by increasing error rates in the evaluation but still for certain situations the quickly computable extension is a reasonable choice. The evaluation shows that for clean data more emphasis is placed on the data term (higher λ) to obtain similar results as the baseline model. If outliers are present a high λ would increase the influence of such. This fact should be carefully attended when using structure-texture decomposed input images.

- In terms of enforcing the regularization within objects the weighting according to image edges improves the estimate near motion boundaries. For the clean data the outcome of the different variants of regularization weight (edge-weighted, directed or tensor-directed) are similar with slight benefits for the tensor-directed approach as it is the most flexible one. In addition, strong textured regions are less influential compared to the edge-weighted/directed approach.

- With respect to adding robustness towards intensity changes, and in our implementation towards general outliers, the additive compensation variable performs well. The benefit of absorbing small outliers due to *e.g.* too strong regularization makes it an applicable data term for a weighted regularization approach. Outliers that are caused by *e.g.* enforcing the regularization within textured objects are then absorbed by the compensation in the data term.

- Exchanging the intensity constancy assumption by a higher-order constancy assumption (*e.g.* gradients), the robustness towards lightning changes and moving shadows are increased again, but the use of gradients in the data matching cost exhibits the disadvantage of being prone to errors caused by strong image texture. This is particularly the case for the synthetic datasets and the reason for the relatively poor performance on the given sequences. Nevertheless, it is a heavily used data matching in the field of optical flow and often the results are improved by combining the intensity and gradient constancy assumption dependent on the underlying data.

- Using the quadratic fitting of the data term's function yields a very flexible approach where also more robust matching terms can be easily integrated. The evaluation shows that especially patch based approaches (with *e.g.* small 3×3 patches) work well in this setting. Especially for strong matching terms like correlation measurements or normalized gradients the approach is very well suited. Especially in combination

with a strong regularization one can obtain state of the art results. In addition using bigger patches increases the robustness towards outliers and the approach is able to handle also corrupted data as we will demonstrate in Section 5.3.1.

- A strong regularization is formed by the nonlocal Huber regularization which imposes knowledge of the underlying data to enforce the smoothness prior within associated regions. The combination with the quadratic fitting of the data term using correlation based measurements results in the most accurate model in terms of aepe for this benchmark sequence. In addition, the combination with a TNCC data term shows the lowest variance over the Middlebury data set which is an indicator of being a robust approach.

Table 5.5.: Result of the evaluation (see also Appendix A) for the "best" λ for different combinations of regularization and data terms. The table depicts a listing of the computed average end-point errors [pixels] for a specific combination of regularization, data term, and λ.

Regularization	Dataterm	λ	Hydrangea		Venus		Grove3		Urban3		
			Dimetrodon		RubberWhale		Grove2		Urban2		Average
Huber	L1	50	0.186	0.235	0.149	0.279	0.217	0.625	0.370	0.485	0.318
Huber	L1 , str.-tex.	70	0.178	0.187	0.122	0.293	0.221	0.645	0.356	0.621	0.328
Weighted H.	L1	35	0.183	0.231	0.139	0.248	0.216	0.590	0.355	0.398	0.295
Weighted H.	L1 , str.-tex.	65	0.171	0.183	0.118	0.278	0.214	0.631	0.358	0.622	0.322
Directed H.	L1	35	0.180	0.224	0.136	0.278	0.207	0.583	0.351	0.383	0.293
Directed H.	L1 , str.-tex.	60	0.168	0.178	0.117	0.276	0.202	0.636	0.360	0.626	0.320
Tensor-d. H.	L1	35	0.180	0.219	0.135	0.287	0.203	0.579	0.348	0.385	0.292
Tensor-d. H.	L1 , str.-tex.	55	0.165	0.174	0.115	0.276	0.193	0.637	0.358	0.622	0.318
Nonlocal-H.	L1	35	0.168	0.208	0.130	0.333	0.189	0.570	0.401	0.382	0.298
Nonlocal-H.	L1 , str.-tex.	75	0.161	0.179	0.114	0.277	0.225	0.648	0.345	0.504	0.307
Huber	L1+comp.	50	0.180	0.175	0.114	0.293	0.220	0.640	0.366	0.714	0.338
Tensor-d. H.	L1+comp.	45	0.180	0.170	0.100	0.271	0.220	0.608	0.339	0.438	0.291
Nonlocal-H.	L1+comp.	45	0.159	0.170	0.099	0.357	0.215	0.598	0.354	0.492	0.306
Huber	L1-grad.	40	0.239	0.186	0.114	0.348	0.299	0.749	0.481	0.834	0.406
Tensor-d. H.	L1-grad.	25	0.225	0.166	0.099	0.398	0.268	0.703	0.445	0.898	0.400
Nonlocal-H.	L1-grad.	30	0.218	0.165	0.100	0.427	0.295	0.731	0.481	0.756	0.397

Regularization	Dataterm	λ	Hydrangea		Venus		Grove3		Urban3	Average
			Dimetrodon	RubberWhale		Grove2		Urban2		
Huber	AD	45	0.208	0.239	0.195	0.444	0.247	0.706	0.433	0.596	0.384
Huber	AD, str.-tex.	75	0.175	0.193	0.153	0.346	0.229	0.701	0.422	0.741	0.370
Huber	SAD3	60	0.187	0.231	0.185	0.363	0.192	0.648	0.450	0.564	0.353
Huber	SAD3, str.-tex.	95	0.168	0.190	0.148	0.321	0.183	0.658	0.433	0.674	0.347
Huber	grad.	25	0.212	0.198	0.158	0.449	0.240	0.786	0.533	1.156	0.467
Huber	AD+grad.	30	0.211	0.197	0.159	0.437	0.235	0.752	0.517	1.018	0.441
Huber	TNCC3	1.75	0.189	0.185	0.126	0.293	0.185	0.670	0.394	0.575	0.327
Huber	norm. grad.	2.75	0.236	0.190	0.124	0.322	0.190	0.691	0.481	0.559	0.349
Tensor-d. H.	AD	35	0.219	0.241	0.196	0.452	0.258	0.694	0.451	0.552	0.383
Tensor-d. H.	AD, str.-tex.	60	0.175	0.190	0.153	0.400	0.220	0.694	0.420	0.709	0.370
Tensor-d. H.	SAD3	40	0.191	0.220	0.185	0.430	0.191	0.644	0.462	0.552	0.359
Tensor-d. H.	SAD3, str.-tex.	95	0.171	0.187	0.142	0.322	0.181	0.636	0.423	0.654	0.340
Tensor-d. H.	grad.	25	0.231	0.195	0.144	0.485	0.271	0.777	0.540	1.089	0.467
Tensor-d. H.	AD+grad.	30	0.231	0.196	0.148	0.480	0.266	0.741	0.515	0.914	0.436
Tensor-d. H.	TNCC3	1	0.182	0.178	0.107	0.244	0.167	0.637	0.309	0.455	0.285
Tensor-d. H.	norm. grad.	1.5	0.227	0.179	0.105	0.257	0.172	0.653	0.328	0.479	0.300
Nonlocal-H.	AD	30	0.236	0.223	0.195	0.474	0.253	0.702	0.528	0.679	0.411
Nonlocal-H.	AD, str.-tex.	70	0.189	0.189	0.142	0.361	0.264	0.726	0.432	0.649	0.369
Nonlocal-H.	SAD3	45	0.190	0.211	0.177	0.457	0.181	0.627	0.502	0.526	0.359
Nonlocal-H.	SAD3, str.-tex.	95	0.162	0.178	0.130	0.334	0.164	0.632	0.431	0.613	0.330
Nonlocal-H.	grad.	30	0.226	0.198	0.139	0.493	0.300	0.818	0.563	1.067	0.475
Nonlocal-H.	AD+grad.	20	0.209	0.188	0.148	0.460	0.232	0.766	0.583	1.136	0.323
Nonlocal-H.	**TNCC3**	**1.75**	**0.186**	**0.172**	**0.093**	**0.265**	**0.146**	**0.590**	**0.287**	**0.378**	**0.265**
Nonlocal-H.	norm. grad.	1.25	0.226	0.185	0.103	0.267	0.163	0.654	0.317	0.443	0.295

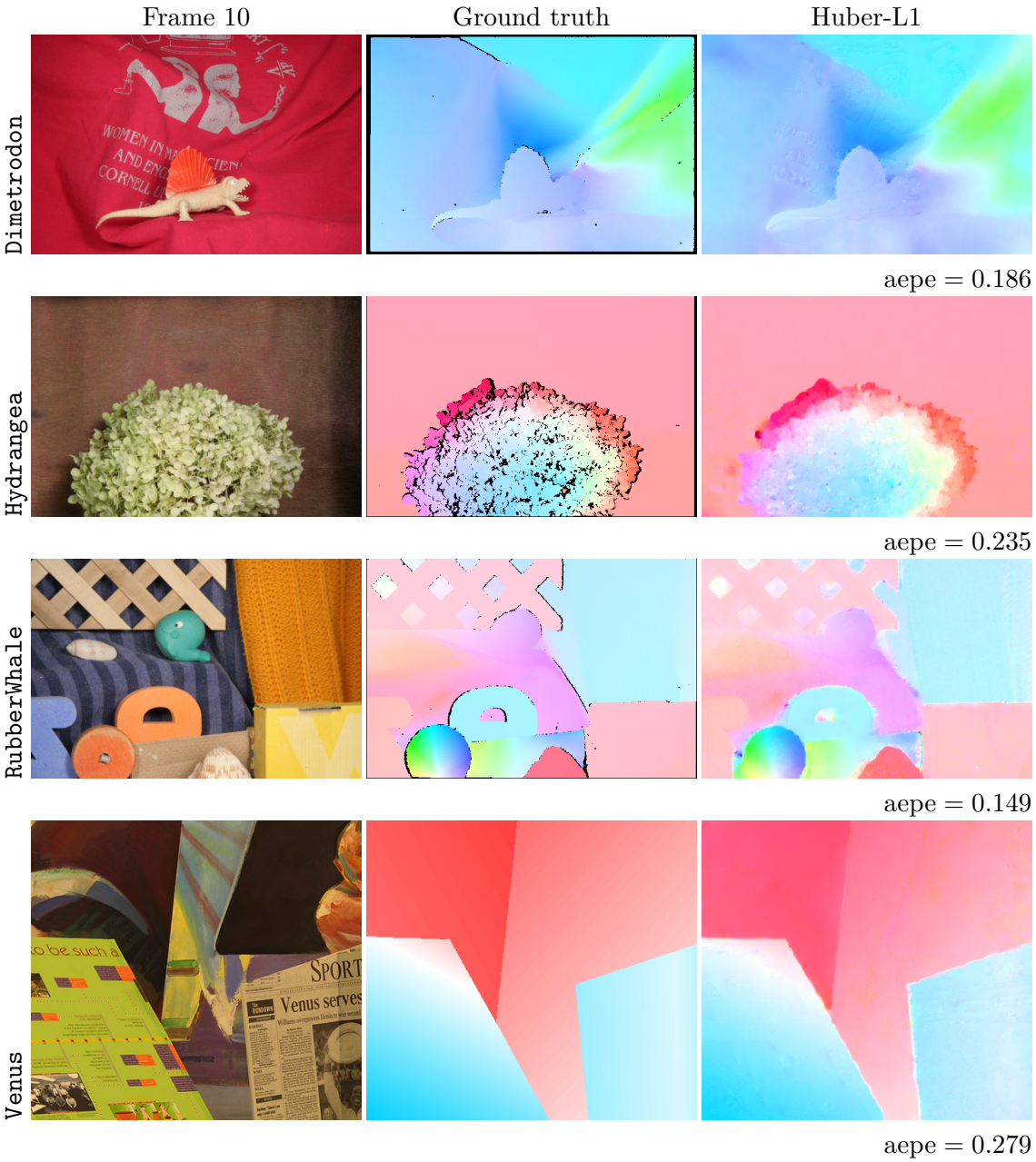

Frame 10
Ground truth
Huber-L1
Dimetrodon
aepe = 0.186
Hydrangea
aepe = 0.235
RubberWhale
aepe = 0.149
Venus
aepe = 0.279

Figure 5.15.

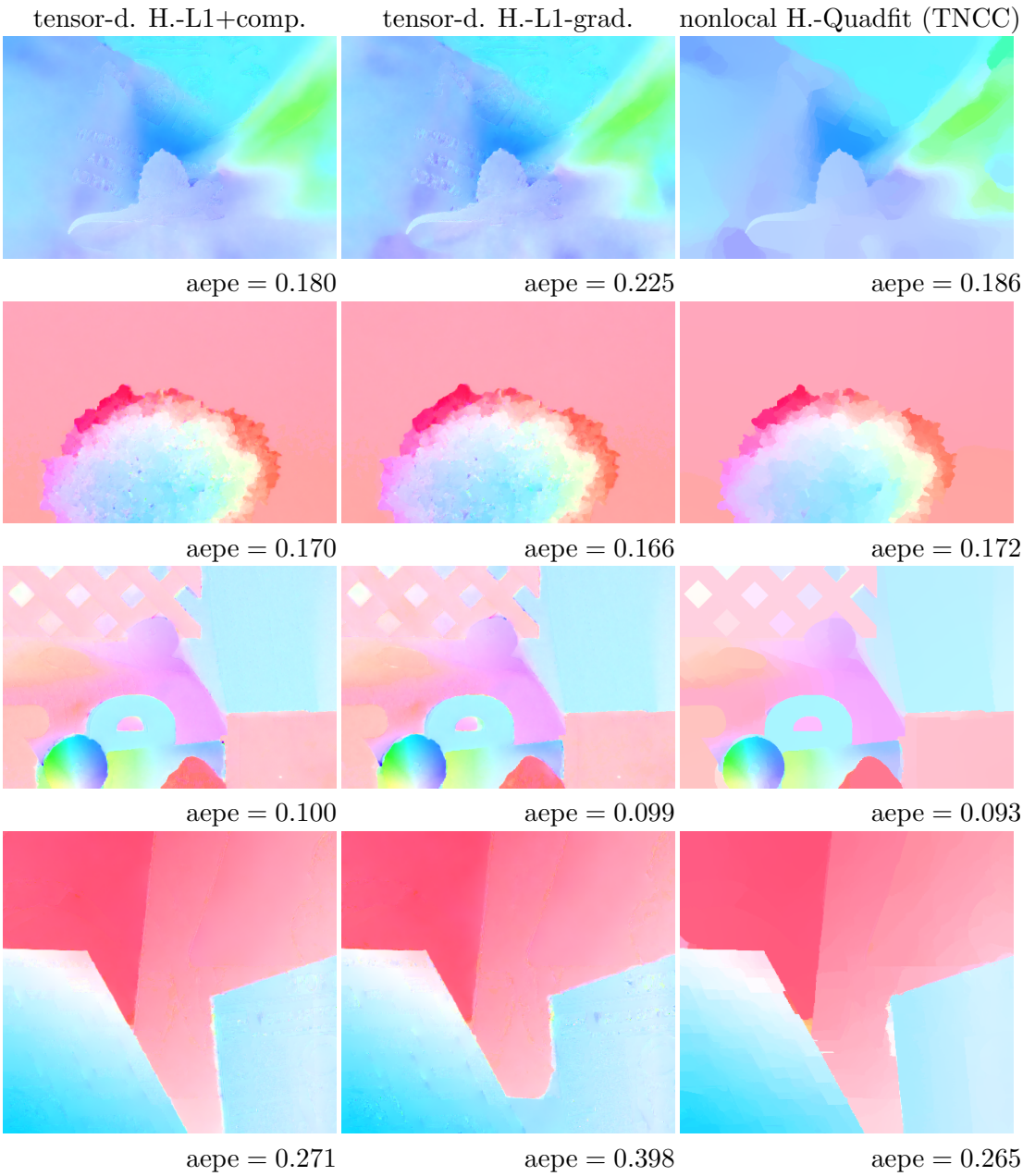


Figure 5.16.: Results of Middlebury's test data set.

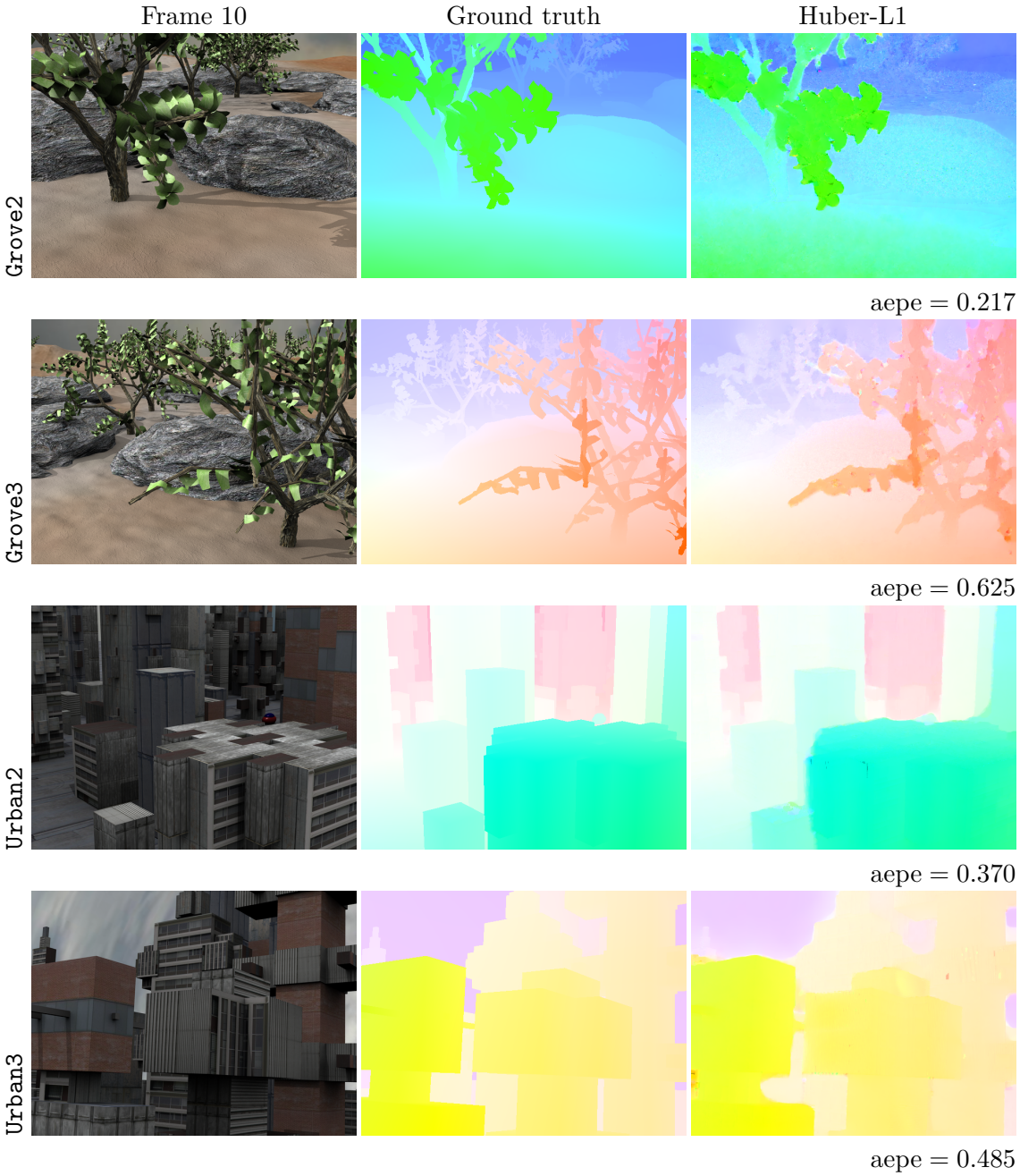

Frame 10
Ground truth
Huber-L1
Grove2
aepe = 0.217
Grove3
aepe = 0.625
Urban2
aepe = 0.370
Urban3
aepe = 0.485

Figure 5.17.

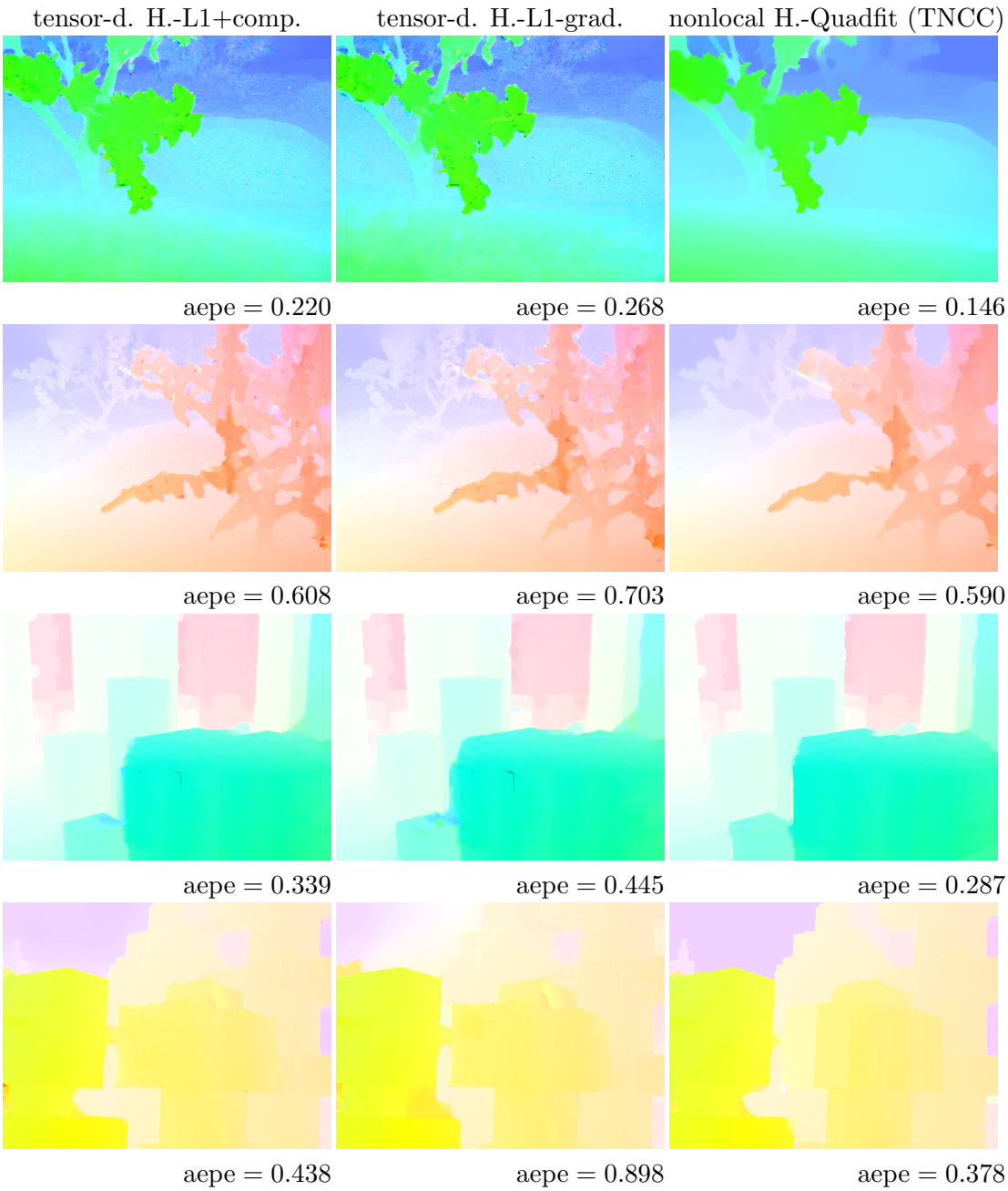


Figure 5.18.: Results of Middlebury's test data set.

5.3.1. Handling Outliers

In the beginning we motivated the optical flow's model improvements also by the demand to handle corrupted video material. By now, most of the material has been clean. Nevertheless, some of the introduced improvements are suitable to treat difficult data with coarse outliers. Figure 5.19 shows an exemplary result of a historical video sequence with typical impairments. The emerging artifacts range from noise and scratches to blobs of completely missing data. To handle such coarse outliers patch-based approaches, realized with the *Quadfit*data term, are well suited. In this particular case we utilized patches of size 22×22 and computed the sum of absolute differences (SAD) to use as a similarity measure for the data term. Imposing such large patches normally oversmoothes the resultant optical flow fields, but utilizing a weighted regularization enables the approach to preserve motion discontinuities and compute a reasonable optical flow field. In contrast to using a point-wise data term as in Figure 5.19d, the optical flow field obtained by the described approach (we used a tensor-directed Huber regularization but also *e.g.* nonlocal regularization could be used here) yield the desired result without any disturbances coming from the impairments.

An additional benefit that we get from the ability to compute correct flow fields without outliers is the possibility to compute a mask with the impairments' locations as in Figure 5.19c. Here, we simply computed the difference image of the reference image (Frame 59) and the warped moving image. Therefore, we utilized the computed optical flow vector to map the second input image towards the reference image. However, let us point out that all the impairments (especially the coarse ones) that occur in the second image are also warped towards the reference frame. The restoration itself is not completed by only computing clean optical flow fields. Nevertheless, a model for computing optical flow that is robust towards outliers is crucial for every restoration pipeline to achieve the desired results. In the following chapter, we will utilize the ability to compute optical flow in an accurate and robust manner and incorporate the optical flow vectors into restoration systems demonstrating the ability to recover image sequences from typical impairments.

(a) Frame 59 **(b)** Frame 60 **(c)** H.-Quadfit (SAD) warped difference image

(d) Huber-L1 **(e)** Huber-L1, warped $60 \rightarrow 59$

(f) tensor-d. H.-Quadfit (SAD) **(g)** tensor-d. H.-Quadfit (SAD), warped $60 \rightarrow 59$

Figure 5.19.: Increasing the neighborhood size of patch based similarity measures gains robustness towards gross outliers. The input images (a) and (b) yield the color-coded optical flow result for (d) the Huber-L1 model and (f) the tensor directed Huber-Quadfit model with SAD data term and the respective warped images (e) and (g). To obtain the clean result we use a matching window of 22×22. With the clean flow field the position of the outliers are obtained by computing the difference of the warped image (g) and the respective input image (a). The difference image is shown in (c).

CHAPTER
SIX

OPTICAL FLOW GUIDED VIDEO PROCESSING

We introduced a versatile framework for optical flow and motion estimation, respectively. Beforehand, we motivated the detailed disquisition on that topic with the importance of the optical flow for video restoration and video processing in general. Still, the ability to handle coarse outliers to estimate an optical flow field is an essential tool for the restoration task itself. This brings us to the next step – the utilization of optical flow vectors in video processing tasks.

Contents

In this chapter, we first extend the well-known TV-ℓ_1 denoising framework to operate on image sequences in a space-time volume. We then use this approach to implement different applications towards video processing. In terms of restoration, we show the results on frame-denoising and the performance if handling coarser outliers. In addition, we show the versatility of the model by generating artificial intermediate images. This is of particular interest when complete frames are lost. But also in the video post-production the approach can be applied to generate artificial slow motion sequences.

As a second approach, we demonstrate the adoption of inpainting towards video sequences. For single frames, exemplar-based inpainting is a well established way to recover image areas without the smoothing effects that might be introduced when information is propagated from the inpainting boarder by *e.g.* a diffusion process.

6.1. Optical Flow Guided TV-ℓ_1

In this chapter we propose a method that is capable of image sequence denoising, restorating impaired frames, reconstructing completely lost frames and interpolating images between frames. It is of note that all this can be realized with a simple extension of the well-known TV-ℓ_1 model and still we are able to produce state-of-the-art results. As our model is closely related to variational image denoising we first pick up the topic from Section 3.3 and give some more insights. Then, we introduce the data representation for an image sequence. After that the optical flow guided extension is discussed including the respective optimization scheme. Finally, we show the method's different fields of application.

6.1.1. Variational Image Denoising

We have already introduced the concept of variational methods in Chapter 3 and some details on variational image denoising have been discussed in Section 3.3. There, we presented the so-called ROF denoising model, proposed in the seminal work of Rudin et al. [1992]. Since its introduction, models of this type were studied intensively. A well-established modification of the classical ROF model is the so-called TV-ℓ_1 model that replaces the ℓ_2 norm in the data term by an ℓ_1 norm:

Starting from the ROF model, we obtain a denoised image v by minimizing the energy functional

$$\min_v \int_\Omega |\nabla v(x)| + \frac{\lambda}{2} (v(x) - f(x))^2 \, \mathrm{d}x \,, \tag{6.1}$$

assuming that v is sufficiently smooth. The unknown (noise-free) data $v(x)$, as well as the given input data $f(x)$, is defined in the image domain Ω. This model is appropriate to remove additive Gaussian noise. When looking at the characteristics of video defects, it immediately becomes obvious that those are not likely to be similar to Gaussian noise. Consider all kinds of missing data like *e.g.* scratches or noise speckles coming from dust on the film during the copying process. Taking a closer look, the outliers are more related to

impulse noise.

Similar to the ROF model, Nikolova [2004], and Chan and Esedoglu [2005] proposed the TV-$\ell_1$model as

$$\min_v \int_\Omega |\nabla v(x)| + \lambda |v(x) - f(x)| \, \mathrm{d}x \,. \tag{6.2}$$

This is a popular approach when it comes to removing impulse noise. Besides the ability to remove strong outliers, the substitution of the ℓ_2-norm in the ROF model with the ℓ_1 norm makes the TV-ℓ_1 model contrast invariant. In addition to image denoising applications, (6.2) can be used for shape denoising and feature selection tasks [Pock, 2008].

6.1.2. TV-ℓ_1 Sequence Denoising

For the moment, the given input data $f(x)$ (and also the model itself) is defined on the two dimensional image grid Ω. We already referred to the issue that applying an approved image denoising algorithm does not necessarily result into a well-performing algorithm for denoising video sequences. Let us, at first, neglect the motion influence for a moment and discuss the data representation. The main idea is to treat a video sequence as a spatio-temporal volume and apply the restoration algorithm not on the 2D images but the 3D volume. For tracking and segmentation it has been shown by Mansouri et al. [2003], and Unger et al. [2009] that the representation of image sequences in a space-time volume is beneficial. For the following model, we will show that this is also true if a spatio-temporal TV-ℓ_1 model is utilized for restoration and video processing tasks. Incorporating the temporal domain into the TV-ℓ_1 model (6.2) results in the minimization task

$$\min_v \int_{\Omega \times T} |\nabla v(x,t)| + \lambda |v(x,t) - f(x,t)| \, \mathrm{d}x \, \mathrm{d}t \,. \tag{6.3}$$

The input sequence $f : \Omega \times T \to \mathbb{R}$ and the sought solution $v : \Omega \times T \to \mathbb{R}$ are defined in a space-time volume.

The benefit of directly encoding the temporal domain into the model does not fully exhaust the potential of such approaches. The drawback of the existing methods is the definition of the temporal gradient. To handle moving objects and the associated (dis-)occlusions, we incorporate optical flow to guide the temporal derivative.

6.1.3. The Optical Flow Guided TV-ℓ_1 Model

In order to incorporate information about objects' movements, we utilize the computed motion vectors to guide the smoothing process. Therefore, the time derivatives are computed along motion trajectories. In order to be robust against occlusions and disocclusions we compute the optical flow in both directions, which we will denote as the forward flow

$$u^+(x,t) = \left(u^{1+}(x,t), u^{2+}(x,t)\right)^T : \Omega \times T \to \mathbb{R}^2$$

and the backward flow

$$u^{-}(x,t) = \left(u^{1-}(x,t), u^{2-}(x,t)\right)^{T} : \Omega \times T \to \mathbb{R}^{2}.$$

We propose to minimize the following optical flow driven TV-ℓ_1 energy functional

$$\min_{v} \int_{\Omega \times T} |\nabla_u v(x,t)| + \lambda(x,t) |v(x,t) - f(x,t)| \, \mathrm{d}x \, \mathrm{d}t. \tag{6.4}$$

As mentioned previously, the input sequence $f(x,t)$, as well as the sought solution $v(x,t)$, are defined in a space-time volume. Unlike in the previous models, we now define the trade-off between regularization and data term in a pointwise manner as $\lambda(x,t)$. This enables the model to neglect the data term for certain areas and only rely on the filling-in effect of the regularizer. There, solely an inpainting operation is performed and only information from neighboring regions are considered. Furthermore, the spatio-temporal gradient operator is defined as

$$\nabla_u = (\partial_{x^1}, \partial_{x^2}, \partial_{t^+}, \partial_{t^-})^{T} \quad , \tag{6.5}$$

where the components ∂_{x^1} and ∂_{x^2} denote standard spatial derivatives. The temporal derivatives are directed by the corresponding optical flow vectors $(u^{+}(x,t), u^{-}(x,t))$ and are denoted as ∂_{t^+} and ∂_{t^-}.

$$\begin{cases} \partial_{x^1} v &= \lim\limits_{h \to 0^+} \dfrac{v(x^1 + h, x^2, t) - v(x,t)}{h} \\ \partial_{x^2} v &= \lim\limits_{h \to 0^+} \dfrac{v(x^1, x^2 + h, t) - v(x,t)}{h} \\ \partial_{t^+} v &= \lim\limits_{h_t \to 0^+} \dfrac{v(x + h_t u^{+}(x,t), t + h_t) - v(x,t)}{h_t \sqrt{1 + |u^{+}(x,t)|^2}} \\ \partial_{t^-} v &= \lim\limits_{h_t \to 0^+} \dfrac{v(x + h_t u^{-}(x,t), t + h_t) - v(x,t)}{h_t \sqrt{1 + |u^{-}(x,t)|^2}} \end{cases} . \tag{6.6}$$

Figure 6.1 shows a graphical illustration of the gradients' directions.

6.1.4. Minimization

Due to the fact that the model is defined to operate on a spatio-temporal domain, the mathematical foundation and its discretization from Section 3.1 must be extended towards the temporal domain. To make the following numerical scheme self-contained we discuss the whole discretization approach although we may repeat some elements that have already been introduced in previously discussed discretizations.

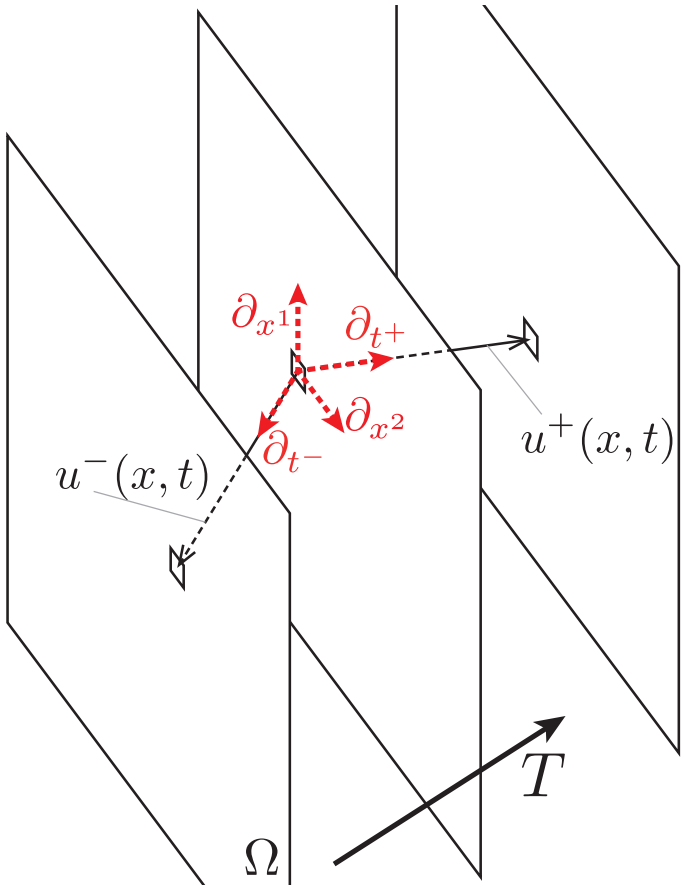


Figure 6.1.: Layout of the spatio-temporal volume and the defined gradient operator (6.5), whose directions are marked as red vectors. The forward $(u^+(x,t))$ and backward $(u^-(x,t))$ optical flow vectors guides the temporal components of this gradient operator.

Discretization

First, we define the image sequence within a three dimensional, regular Cartesian grid of the size $M \times N \times K$:

$$\Omega \times T = \{(i,j,k) : 1 \leq i \leq M, 1 \leq j \leq N, 1 \leq k \leq K\} \tag{6.7}$$

The grid size is 1 and the discrete pixel positions within the defined volume are given by (i,j,k). The finite dimensional vector spaces X and Z are extended towards

$$\begin{aligned} X &= \mathbb{R}^{MNK}, \\ Z &= \mathbb{R}^{4MNK}, \end{aligned} \tag{6.8}$$

equipped with standard scalar products denoted by $\langle\cdot,\cdot\rangle_X$ and $\langle\cdot,\cdot\rangle_Z$. Next, we define the discretized versions of the anisotropic spatio-temporal gradient operator ∇_u (6.5) with h and h_t controlling the spatial and temporal influence of the regularization

$$(D_u v)_{i,j,k} = \Big(\underbrace{(D_{x^1}v)_{i,j,k}, (D_{x^2}v)_{i,j,k}}_{spatial}, \underbrace{(D_{t^+}v)_{i,j,k}, (D_{t^-}v)_{i,j,k}}_{temporal}\Big)^T \ . \tag{6.9}$$

The spatial gradients are discretized using simple finite differences yielding

$$(D_{x^1}v)_{i,j,k} = \begin{cases} \frac{v_{i+1,j,k}-v_{i,j,k}}{h} & \text{if } i < M \\ 0 & \text{if } i = M \end{cases} \ , \tag{6.10}$$

$$(D_{x^2}v)_{i,j,k} = \begin{cases} \frac{v_{i,j+1,k}-v_{i,j,k}}{h} & \text{if } j < N \\ 0 & \text{if } j = N \end{cases} \ . \tag{6.11}$$

If the flow vector points in-between discrete locations on the pixel grid we use a linear interpolation of the surrounding pixels to compute the flow-guided temporal gradients. In the following, we describe the interpolation for the spatial component guided with the forward flow. The same is, of course, valid for the component where the gradient is steered with the backward optical flow. The linear interpolated pixel positions

$$\left(\bar{i}^+, \bar{j}^+, k\right) = \left(i + h_t u^{1+}_{i,j,k}, j + h_t u^{2+}_{i,j,k}, k\right) \ , \tag{6.12}$$

and their coordinates

$$\begin{aligned} \bar{i}^+_1 &= \lfloor \bar{i}^+ \rfloor \ , \quad \bar{i}^+_2 = \bar{i}^+_1 + 1 \ , \\ \bar{j}^+_1 &= \lfloor \bar{j}^+ \rfloor \ , \quad \bar{j}^+_2 = \bar{j}^+_1 + 1 \ . \end{aligned} \tag{6.13}$$

The corresponding weighting factors are computed as the distances

$$\delta^+_i = \bar{i}^+_2 - \bar{i}^+ \quad \text{and} \qquad \delta^+_j = \bar{j}^+_2 - \bar{j}^+ \ . \tag{6.14}$$

The temporal gradient operators then yield

$$(D_{t^+}v)_{i,j,k} = \begin{cases} \frac{a^+ + b^+ + c^+ + d^+ - e^+}{h_t \sqrt{1 + \left|\left|u^+_{i,j,k}\right|\right|_2^2}} & \text{if} \quad a^+, b^+, c^+, d^+, e^+ \in \Omega \times T \\ 0 & \text{else} \ , \end{cases} \tag{6.15}$$

$$(D_{t^-}v)_{i,j,k} = \begin{cases} \frac{a^- + b^- + c^- + d^- - e^-}{h_t \sqrt{1 + \left|\left|u^-_{i,j,k}\right|\right|_2^2}} & \text{if} \quad a^-, b^-, c^-, d^-, e^- \in \Omega \times T \\ 0 & \text{else} \ , \end{cases} \tag{6.16}$$

with

$$\begin{array}{l|l} a^+ = \delta_i^+ \delta_j^+ v_{\bar{i}_1^+, \bar{j}_1^+, k+1} & a^- = \delta_i^- \delta_j^- v_{\bar{i}_1^-, \bar{j}_1^-, k} \\ b^+ = (1-\delta_i^+)\delta_j^+ v_{\bar{i}_2^+, \bar{j}_1^+, k+1} & b^- = (1-\delta_i^-)\delta_j^- v_{\bar{i}_2^-, \bar{j}_1^-, k} \\ c^+ = \delta_i^+ (1-\delta_j^+) v_{\bar{i}_1^+, \bar{j}_2^+, k+1} & c^- = \delta_i^- (1-\delta_j^-) v_{\bar{i}_1^-, \bar{j}_2^-, k} \\ d^+ = (1-\delta_i^+)(1-\delta_j^+) v_{\bar{i}_2^+, \bar{j}_2^+, k+1} & d^- = (1-\delta_i^-)(1-\delta_j^-) v_{\bar{i}_2^-, \bar{j}_2^-, k} \\ e^+ = v_{i,j,k} & e^- = v_{i,j,k+1}\,. \end{array} \tag{6.17}$$

To implement the operator D_u we use a sparse matrix representation. This is of particular interest because we will also need to implement the adjoint operator D_u^* which is defined through the identity $\langle D_u v, p\rangle_Z = \langle v, D_u^* p\rangle_X$. Note that here, the adjoint operator is simply the matrix transpose.

Now, (6.4) can be rewritten in the discrete setting as the following minimization problem

$$\min_v \; ||D_u v||_1 + ||\Lambda(v-f)||_1 \quad , \quad \text{with } \Lambda = \operatorname{diag}(\lambda) \;. \tag{6.18}$$

Primal-Dual Algorithm

Dualizing both ℓ_1 norms in (6.18) yields the saddle-point problem

$$\begin{aligned} \min_v \max_{p,q} \; & \langle D_u v, p\rangle_Z + \Lambda \langle v-f, q\rangle_X \quad , \\ \text{s.t.} \quad & 0 \le v \le 1\,, \\ & ||p||_\infty \le 1\,, \quad \text{and} \\ & -1 \le s \le 1 \end{aligned} \tag{6.19}$$

where $p = \left(p^1_{i,j,k}, p^2_{i,j,k}, p^3_{i,j,k}, p^4_{i,j,k}\right) \in Z$ and $q = (q_{i,j,k}) \in X$ denote the dual variables, and the discrete maximum norm $||p||_\infty$ is defined as

$$\begin{aligned} ||p||_\infty &= \max_{i,j,k} |r_{i,j,k}| \quad , \\ |p_{i,j,k}| &= \sqrt{(p^1_{i,j,k})^2 + (p^2_{i,j,k})^2 + (p^3_{i,j,k})^2 + (p^4_{i,j,k})^2} \;. \end{aligned} \tag{6.20}$$

Applying the primal-dual algorithm (*cf.* Section 3.6.4) yields the following iterates to perform a gradient descent update for the primal variable and a gradient ascent update for

the dual variables:

$$\begin{aligned} p^{n+1} &= \Pi_{\mathcal{B}_1}\left[p^n + \sigma\left(D_u \overline{v}^n\right)\right] \\ q^{n+1} &= \Pi_{[-1,1]}\left[q^n + \sigma\Lambda\left(\overline{v}^n - f\right)\right] \\ v^{n+1} &= v^n - \tau\left(D_u^* p + \Lambda q\right) \\ \overline{v}^{n+1} &= 2v^{n+1} - v^n \end{aligned} \tag{6.21}$$

The projection $\Pi_{\mathcal{B}_\zeta}$ is a simple point-wise projections onto the ball with the radius ζ, and $\Pi_{[a,b]}$ denotes the point-wise truncation to the interval $[a, b]$. τ and σ are the primal and dual update step sizes satisfying $\tau\sigma L^2 \leq 1$, where $L^2 = \|(D_v^*, \Lambda^*)\|^2$.

6.1.5. Application to Sequence Restoration

In the following, we will demonstrate different fields of applications for the proposed method. At first, we will show the denoising result of a sequence where some kind of motion is present. Then, we demonstrate the method's appliance towards sequence restoration by inpainting destructed areas. Finally, the ability to generate complete frames is evaluated on the Middlebury benchmark for frame interpolation and, furthermore, an artificial slow-motion sequence is generated.

6.1.6. Denoising

The following example mainly aims to demonstrate the ability to enhance the smoothing results over a sequence of frames including motion in comparison to classical spatio-temporal approaches that do not take the guided gradient operator into account. Therefore, we compare results of a classical TV-ℓ_1 denoising, extended to the spatio temporal domain, with the proposed method. The smoothed images in Figure 6.2 reveal that the classical TV-ℓ_1 model does not prevent the smoothing over moving edges. In this example the camera is moving and therefore every edge in the image is a moving edge. Especially the details shown in Figure 6.2d and 6.2e demonstrate that the optical flow guided variant also preserves fine details by enforcing the temporal smoothness along motion trajectories.

6.1.7. Inpainting

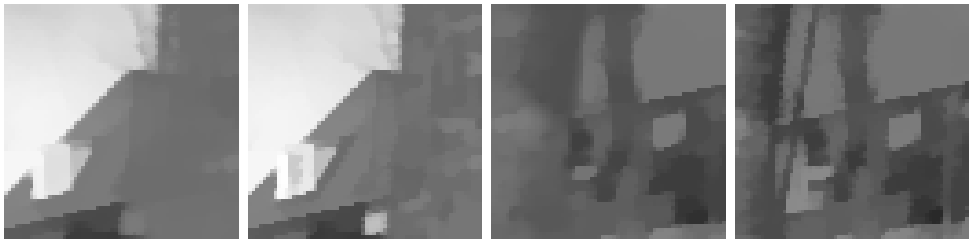
The previous example shows the benefit of guiding the gradient with a spatio-temporal variant of the TV-ℓ_1 when the sequence is smoothed. As we have already stated, the ℓ_1 norm penalizing the data fidelity term makes the model appropriate to remove impulse noise. In the following, we demonstrate that with incorporating optical flow guidance also coarse outliers can be removed from the sequence. A requirement for a successful restoration

(a) Clean input sequence.

(b) TV-ℓ_1 smoothing result without any guidance.

(c) Guided TV-ℓ_1 smoothing result.

(d) Details showing the differences the two approaches. The right images shows the proposed guided variant.



(e) Details showing the differences the two approaches. The right images shows the proposed guided variant.

Figure 6.2.: Denoising the `flowergarden` sequence with the optical flow guided TV-ℓ_1 model enhances the details in the result compared to the variant without optical flow guidance.

is the capability to compute the optical flow vectors. However, for big destructed areas it may occur that it is not possible to compute any reasonable optical flow field. In these cases the vectors must be interpolated in these areas as *e.g.* done in Section 6.1.8 (see also Figure 6.4).

In the presented example in Figure 6.3 the task is to reduce the coarse outliers but still conserve the historical characteristic of the sequence. Therefore, we mainly emphasize the temporal smoothing as the defects occur within single frames (so-called single-frame defects) and valid information can be propagated from neighboring frames. In this experiment, the values for λ are set to $\lambda(x,t) = 2$ if the pixels are considered good and to $\lambda(x,t) = 0$ otherwise. In the areas where the data term is switched of ($\lambda(x,t) = 0$) solely inpainting is performed. To reduce the influence of the spatial smoothing, the weightings of the regularity are chosen as $h = 2.5$ and $h_t = 1$. Especially the desire to reconstruct fine image details makes the task a difficult problem. The results in Figure 6.3 show that due to the reduced spatial smoothing, the overall characteristic of the movie is preserved and all the coarse outliers are removed. In addition, all fine details, like the wheel's spokes, are recovered.

6.1.8. Application to Frame Interpolation

For frame interpolation, we use the ability of regularizing along the temporal trajectory in the space-time volume. When solving (6.4) in such a setting, the intermediate frames are generated and the pixel's movement are reasonably interpolated along the object's motion trajectories. Hence, the optical flow vectors must be propagated in order to define the temporal gradients for the unknown frames. In this case, we assume a locally linear motion around the sought frame and propagate the flow vectors from the previous frame towards the unknown positions as illustrated in Figure 6.4

For a numerical evaluation of interpolating intermediate frames within an image sequence we use sequences of the Middlebury benchmark [Baker et al., 2011], where interpolated frames are available as ground truth. The ground-truth datasets, as well as the input data, are available at the Middlebury website*. For comparison, we show the interpolated image v, the ground-truth $\check{v}$ and the difference image $v_{\text{diff}} = |\check{v} - v|$ in Fig. 6.5-6.7. As a quantitative measurement we give the root mean squared error (RMSE)

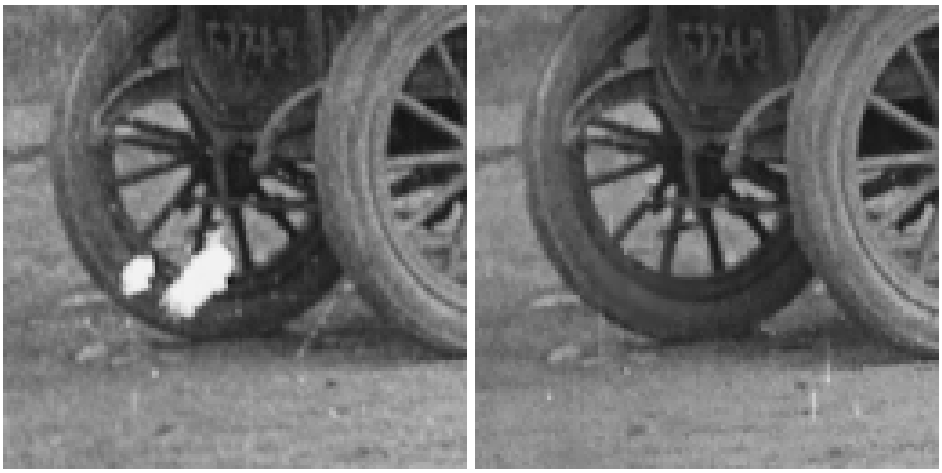
$$e_{\text{RMSE}} = \sqrt{\frac{1}{MN} \sum_{j=1}^{N} \sum_{i=1}^{M} (\check{v}_{i,j} - v_{i,j})^2}\,. \tag{6.22}$$

In Fig. 6.8 an example of the Middlebury database is shown. Here, large displacements, fast motion, small scaled structures and complex occlusions are the major difficulties within these datasets. The magnified regions show that, again, the interpolated movement is reasonable and the (dis-)occluded regions are handled robustly. Having a closer look at

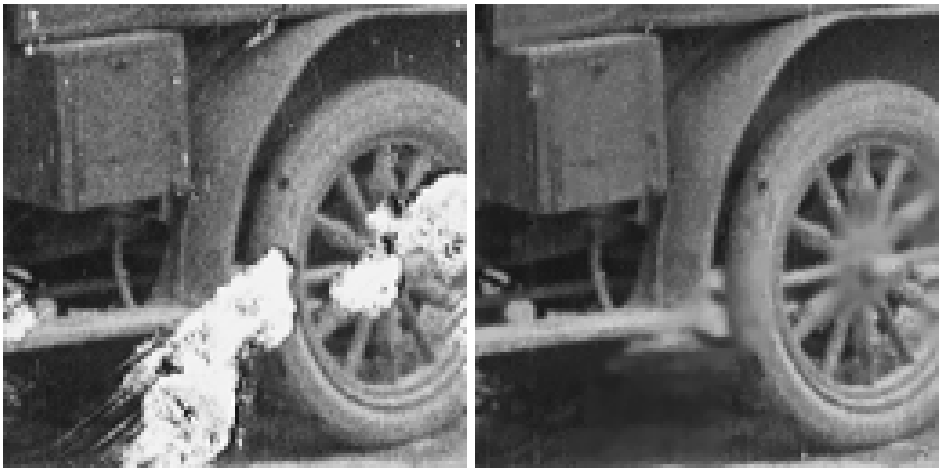
* http://vision.middlebury.edu

(a) Input sequence.

(b) Inpainting result.



(c)



(d)

Figure 6.3.: Inpainting corrupted regions in a defect afflicted sequence (a). The restoration is performed with the aim to obtain a naturally looking sequence (b). In (c) and (d) we show two magnified regions with the corrupted input on the left and the resulting inpainting result on the right.

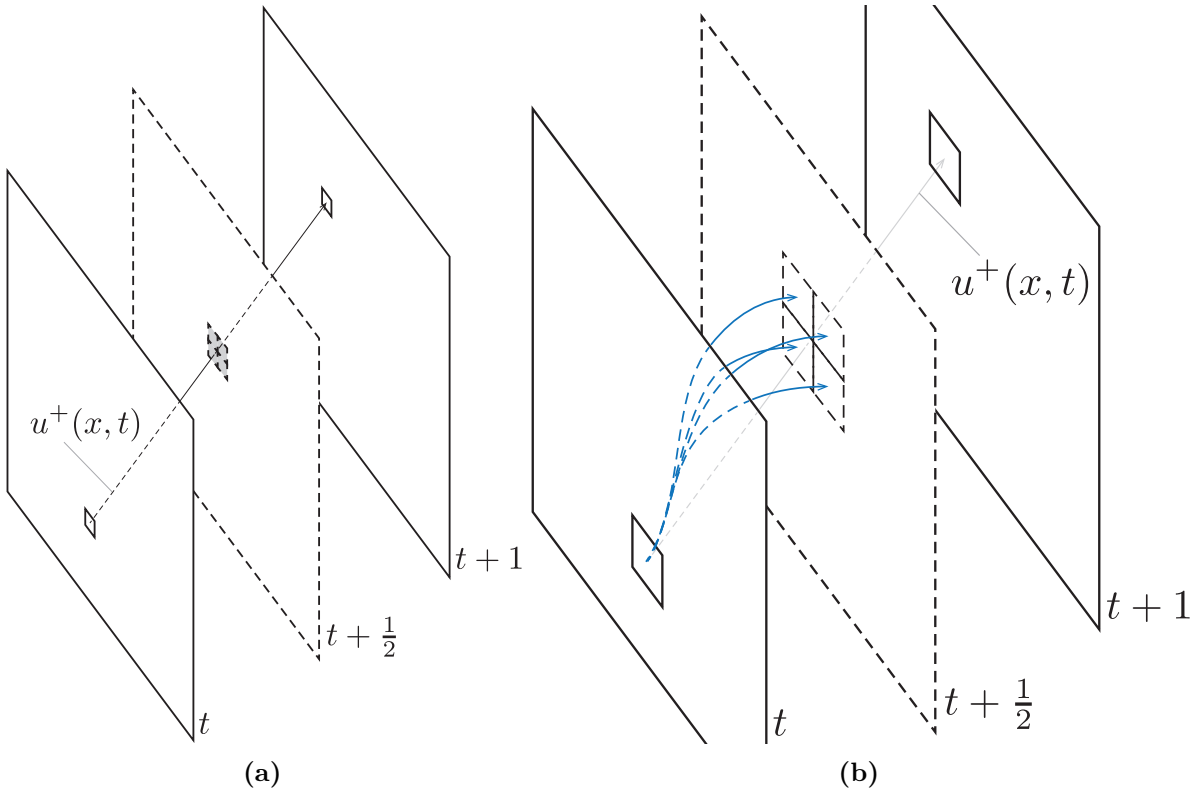
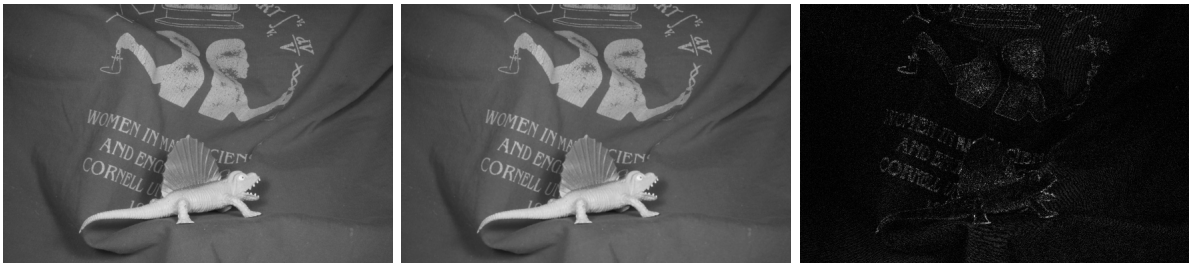


Figure 6.4.: Flow propagation: (a) shows the basic setting of an unknown frame $(t, t+1$: known; $t+\frac{1}{2}$: unknown) and an exemplary forward flow vector. (b) shows the relevant pixels that are used when the optical flow is propagated using the linear interpolation factors as in (6.14).

the ball reveals a drawback of the approach. As the ball has a rather big movement we are not capable to capture it completely accurately with the used optical flow approach and in areas, where no motion vectors are available (or the motion vectors are not 100% correct), the interpolation can produce incorrect results. Besides the benchmark sequence, we also show a result from a historical video sequence where we generated nine intermediate frames. In Figure 6.9 we only show every fifth frame for presentation purpose and to also receive the impression of some motion. The remaining frames are of equal quality. To be more precise, as we assume a linear motion between two known frames the interpolation quality is independent of the number of interpolated frames. Let us point out some key observations from this sequence. Natural motion like the persons' movement in the foreground, including gestures like opening/closing of the mouth, is well captured by the approach and also continued during the intermediate frames. The camera motion aggravates the handling of *e.g.* the poles in the background due to their size and the low contrast to the background. Still, those are well captured and the position remains valid in all the frames. Tiny structures like details in the hair are preserved too and are correctly interpolated towards



(a) Ground truth ($\check{v}$). **(b)** Interpolation result (v). **(c)** Difference image..

Figure 6.5.: `Dimetrodon`-dataset: $e_{\mathrm{RMSE}} = 7.478 \cdot 10^{-3}$

(a) Ground truth ($\check{v}$). **(b)** Interpolation result (v). **(c)** Difference image..

Figure 6.6.: `MiniCooper`-dataset: $e_{\mathrm{RMSE}} = 1.78 \cdot 10^{-02}$

(a) Ground truth ($\check{v}$). **(b)** Interpolation result (v). **(c)** Difference image..

Figure 6.7.: `Walking`-dataset: $e_{\mathrm{RMSE}} = 1.15 \cdot 10^{-2}$

in-between positions also when the two persons move. Most important for viewing such material, the overall impression of the artificially slow-motion sequence is natural and does not exhibit any prominent outliers that draws the attention of the observer. This example shows that the approach is well-suited for typical video sequences. Let us once more point out that the resultant quality is very dependent on the quality of the estimated optical flow vectors. This again shows the importance of a robust and accurate motion estimator in the field of video processing.

6.1.9. Discussion

The purpose of this section is twofold. On the one hand we present an extension to the classical TV-ℓ_1 denoising model and on the other hand it demonstrates the importance of accurate and robust motion estimation in the field of video processing. Let us emphasize once more that the only modification of classical TV-ℓ_1 denoising is the extension towards spatio-temporal input data and incorporating directional information to guide the temporal gradient along motion trajectories. Hence, for video restoration, denoising or interpolation tasks, the motion must be robustly estimated even when outliers occur. If this is not possible, as for the interpolation case, a reasonable interpolation must be performed to impose motion vectors into those areas. The different results in this section show that the presented approach is useful for both, video restoration, and the post production stage (*e.g.* to generate artificial slow motion sequences). Moreover, the optical flow guidance enables the TV-ℓ_1 denoising to restore big regions without introducing blur. Still, when recovering from very big outliers, or when difficult elements are processed, it might happen that some blur is introduced and abates the overall impression. In the details shown in Figure 6.3d such blurring artifacts can be observed. Still, for this example it is not a problem when the sequence is played back because the impairment is only applied on this single frame and the introduced blur is not disruptive. In the following section we will pick up on this problem and introduce an approach that is designed for situations where the optical flow guided TV-ℓ_1 approach would not lead to the desired results.

Figure 6.8.: Interpolating the `Backyard`-sequence.

Figure 6.9.: Interpolating the `Macunaima`-sequence.

6.2. Optical Flow Guided Exemplar-based Inpainting

In this section we discuss an approach that is complementary to the optical flow guided interpolation presented in Section 6.1. While the guided TV-ℓ_1 model is well suited to interpolate single frame defects, the upcoming approach is especially designed for artifacts that occur over a longer period. Such an artifact can *e.g.* be caused by dirt on the lens while recording. As an illustrative example and for demonstrating the capabilities of the following approach, we use a sequence with a *lens hair** causing a static artifact over a complete shot (see Figure 6.10). The additional stripe pattern on the moving person makes the restoration of this sequence especially difficult and the duration of the artifact demands a different approach than the one presented earlier. Using information from the neighboring frame might help in cases where a valid motion is estimated but for static parts of the sequence no information can be propagated towards the unknown region. A further requirement is, that the restoration must be fully automatic as the impairments can appear in every single frame. The only user input should be to create a mask that tells the algorithm where the defect appears. For the example in Figure 6.10 we give the corresponding masks in Figure 6.10d. In the following we will first give some details on exemplar-based inpainting in general and then discuss the extension by incorporating optical flow vectors.

Before we start with any details on this approach, let us point out that the presented basis does not fit the mathematical framework of the rest of the paper. Still, it is a good chance to present a different approach where also optical flow guidance can be used. For the interested reader, the formulation of exemplar-based inpainting within a variational framework has been introduced by Arias et al. [2011]. The approach has a strong relation to the nonlocal total variation where we used the nonlocal Huber pendant for optical flow regularization in Section 5.1.3.2

(a) Frame 003 **(b)** Frame 006 **(c)** Frame 009 **(d)** Inpainting mask

Figure 6.10.: Three exemplary frames out of 52 frames from a sequence with a static artifact over a complete shot of a video sequence. The artifact is caused by a hair on the camera lens. (d) is the mask for the inpainting region.

* The terminology "lens hair" is used for a static missing-data artifact over multiple frames that is caused by a hair on the lens.

6.2.1. Exemplar-based Inpainting

The system we implemented for single frame exemplar-based inpainting is based upon the techniques proposed by Criminisi et al. [2004]. In addition, to improve the performance, we use sampling methods for the possible input patches that are related to the "PatchMatch" approach by Barnes et al. [2009]. For the sake of completeness, we outline the used algorithm and for details on those related approaches we refer to them and the references therein.

The basic concept of exemplar-based inpainting is the search for adequate patches that fit into the desired region. In other words, the unknown region is iteratively filled by looking for patches that fit towards the surrounding image structure. To maintain prominent image edges the filling order is of substantial importance. This is also a great concern within the work of Criminisi et al. [2004]. Given an input image I, we define the domain of the image $\mathcal{I} = \operatorname{dom} I$ as the set of all available pixels. Furthermore, the region $\mathcal{I}_u$ defines the unknown pixels and depicts the area that should be filled by the algorithm. The pixels that are not declared as unknown have some color value and are defined through the set of pixels $\mathcal{I}_s = \mathcal{I} - \mathcal{I}_u$. The contour around the unknown area is called the fill front $\delta\mathcal{I}_u$. A patch, chosen to be filled, is centered at a point $p \in \delta\mathcal{I}_u$, and denoted as Θ_p. The size w_p of the patch is defined by the user. The quality of the inpainting result is very dependent on this patch size. We defer the selection of this parameter for a moment and discuss that separately after the algorithm's details.

In order to define the succession of patches to fill, Criminisi et al. [2004] define a priority measure $P(p)$ that is defined as a combination of a confidence value $C(p)$ and a measure that is related to the given data $D(p)$:

$$P(p) = C(p)D(p) \tag{6.23}$$

The confidence is computed from the confidences of available pixels q from within the source region yielding

$$C(p) = \frac{\sum_q C(q)}{|\Theta(p) \bigcap \mathcal{I}_s|}\,, \quad \text{with} \quad q \in \Theta(p) \bigcap \mathcal{I}_s\,, \tag{6.24}$$

and $|\cdot|$ denoting the number of elements within the given set. The confidence values for the single pixels are updated when the pixel is filled and is initialized as $C(p) = 1$ if p lies in the source region and $C(p) = 0$ if $p \in \mathcal{I}_u$. The ordering with respect to the data information is dependent on image edges. The basic concept is to continue edges that hit the fill front in the unknown area. As human cognition is very sensitive to edges in the image, it is very important that strong edges are continued in a reasonable fashion. Therefore, $D(p)$ is computed as

$$D(p) = \left| \left\langle \nabla I^{\perp}(p), n(p) \right\rangle \right|\,, \tag{6.25}$$

where $\nabla I^{\perp}(p)$ is the so-called isophote and $n(p)$ the normal vector of the fill front. Both, $\nabla I^{\perp}(p)$ and $n(p)$ are defined at the point p. For a geometrical interpretation see Figure 6.11. With the computed priorities the patches can be worked on in the respective order.

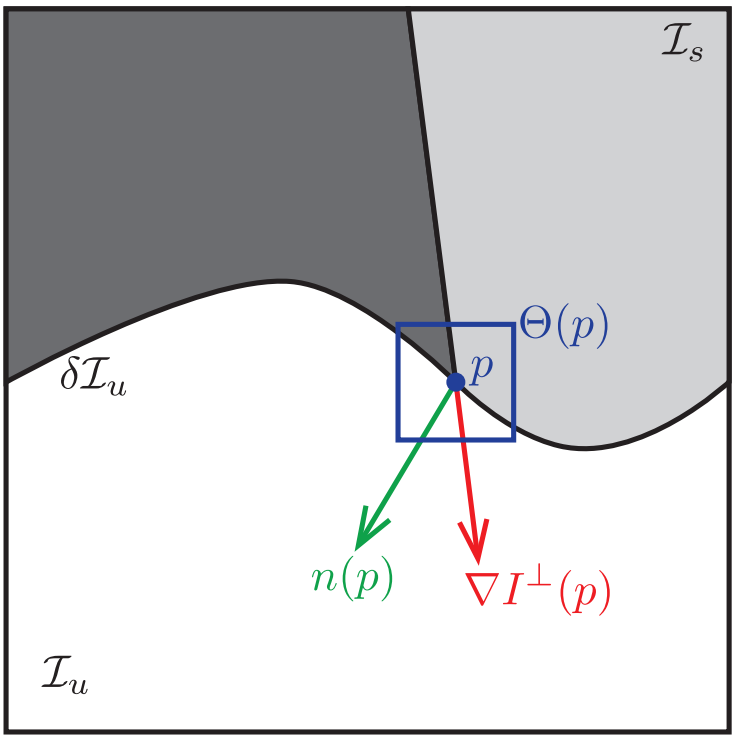


Figure 6.11.: Illustration of an exemplar patch with an isophote *etc.*

Hence, the next step is to find a valid proposal that fits into the desired area. Unlike Criminisi et al. [2004] we do not perform a complete search over all available patches in the image space but in order to accelerate the algorithm, we take a certain amount of random samples to search after a possible match like in [Barnes et al., 2009]. We modified the random sampling of patch candidates in such a way that more nearby patches are taken into account than patches that are further afar. For comparing patch similarities we compute the sum of squared differences (SSD) with respect to pixels that are already filled. Unlike Criminisi et al. [2004] we use the RGB color space as we found the results more applicable compared to the CIE Lab color space. Though, this might depend on the used data. Note, that only patches that completely lie in the source region are considered as candidates to fill the current patch. This is defined as

$$\Theta_{\overline{q}} = \underset{\Theta_q \in \mathcal{I}_s}{\arg\min}\, SSD\left(\Theta_p, \Theta_q\right) . \tag{6.26}$$

Here, Θ_p denotes the patch that is desired to be filled, Θ_q a potential candidate and $\Theta_{\overline{q}}$ the best found patch with respect to the sum of squared difference computed on valid pixels

between Θ_p and $\Theta_{\overline{q}}$.

When an adequate patch is found the desired data is then copied into the unfilled pixels of Θ_p which are denoted as p_u. In addition, the confidence values of the newly filled pixels must be updated. Hence, the confidence value for the newly filled pixels is defined as

$$C(p_u) = C(p)\,. \tag{6.27}$$

A drawback that we found when applying this algorithm to difficult restoration problems is its dependency on the patch size. Often, the border of the patches in the filled areas is noticeable. Even if it does not distract the result for a single frame restoration, the slight changes over time would catch the attention of the viewer. In other words, the restoration of a sequence is more sensitive to small artifacts. As mentioned, even if the borders of the filled patches (*e.g.* where different patches meet) are hardly noticeable in single frames, it can be an disturbing artifact within the sequence. Changing artifacts are especially visible when played back and only slight changes in *e.g.* edges or even different and changing grain can distract the viewer immediately.

Before we proceed to the temporal guidance of the inpainting approach we discuss some further improvements for searching for patches and filling the unknown image regions. First, we will increase the patch size for searching possible patches. Hence, we define an additional overlap o that is added to the current patch for searching candidates. Hence, to find matching patches we use the patch size

$$w^+ = w_p + 2o\,. \tag{6.28}$$

This has the benefit that more information is used to choose a patch and, more importantly, when the current patch meets the border of an already filled patch, the overlapping region ensures that the current patch is filled with information that better fits the previously filled-in information. Furthermore, when an appropriate candidate is found we modify the insertion of the patch by additionally adding a slight blending to already filled pixels. This also reduces the appearance of blocking artifacts at patch borders. When filling the desired patch, the data information is again copied to pixels that are unknown and for known pixels the mentioned blending step is performed. Therefore let $p_i \in \Theta_p$ be the pixels from the current patch and $q_i \in \Theta_{\overline{q}}$ the respective pixels from the found patch. Introducing a real-valued parameter $b \in [0, 1]$ that weights the blending, the filling step for the pixels in the current patch Θ_p is defined as

$$I(p_i) = \begin{cases} I(q_i) & \text{if} \quad p_i \in \Theta_p \bigcap \mathcal{I}_u \\ (1-b)\, I(p_i) + bI(q_i) & \text{if} \quad p_i \in \Theta_p \bigcap \mathcal{I}_s \end{cases}\,. \tag{6.29}$$

The combination of both extensions yields a more consistent inpainting result and improves the quality of the filled regions. In addition, the different patch sizes for searching a matching patch and filling the desired regions increases the robustness of the algorithm

towards this parameter. We set the patch size to $w_p = 9$ and the overlap to $o = 2$ and $w^+ = 13$ respectively. In order to avoid blurring due to the introduced blending step we set $b = 0.5$. When big regions are inpainted the regions might be reduced if some disturbing blurring artifacts occur. Still the temporal consistency of patches is not yet enforced and it can happen that the restored images look reasonable when separately examined, but when the consecutive frames are played back the temporal consistency is not given. In other words, the observer notices some rapid changes from frame to frame in the repaired regions. Therefore, we introduce an additional temporal guidance to improve the visual quality of the repaired video sequence.

6.2.2. Imposing Optical Flow Guidance

The benefit of introducing optical flow guidance in a holistic variational framework like the optical flow guided TV-ℓ_1 denoising in Section 6.1.3 has already been shown. It is a legitimate question, if the advantages can be conveyed to other approaches. The exemplar-based inpainting method is a typical patch-based approach and in the following we will present a simple extension on how to incorporate optical flow vectors to achieve temporal consistency of the repaired region.

As mentioned previously, slight variations in the patch constellation already distract the viewer and destroy the impression of a nicely repaired video sequence although the single frames are of reasonable quality. A further demand is, that the restoration should be automated over the whole sequence and should be able to handle static impairments over a longer part of the sequence. As shown in the example, the impairment will mostly be applied over the whole shot. In the following, we will discuss the incorporation of flow guidance and in addition show a further modification of the optical flow approach to explicitly handle coarse outliers where the position is known.

To include temporal information in a patch-based inpainting approach we enforce the patch that is chosen to fit towards the available source region of the current patch Θ_p to be similar to the patch in the previous frame. In order to account for moving objects we transform the previous image by means of warping with the computed optical flow vectors between the previous and the current frame. Let us start by inpainting the first frame with the approach presented in the previous section if no prior information is available. If a previous frame is available without impairment one can directly start off with the guided approach that we present in the following.

Now, with a restored (or intact) frame available we compute the optical flow between this frame and the next impaired frame in the sequence. To be able to compute valid optical flow vectors also for the inpainting region, one must consider the robustness of the used optical flow method. In this case we can also use the mask that is given by the user for the unknown region. Usually, this mask will be equivalent to the outliers and can be incorporated in the optical flow estimation. Therefore, we expand the used optical flow model to the ability to define the trade-off between regularization and data fidelity in a

pointwise manner as function $\lambda(x)$. Hence, for the typical optical flow model the objective function is then defined as

$$\min_{u \in Y} \mathcal{R}(u(x)) + \lambda(x)\, \mathcal{D}(u(x))\, \mathrm{d}x\,. \tag{6.30}$$

To neglect the data information in the impaired region, we set $\lambda(x) = 0$ for the unknown pixels $x \in \mathcal{I}_u$. As a consequence, only the regularization is active for these areas and the optical flow vectors are inpainted according to the neighboring information. With the computed optical flow vectors the previous frame can be warped towards the current frame. Now we define the currently active patch again as Θ_p and in addition we define the underlying patch of the warped previous image at the same position p as $\overline{\Theta}_p$ which we will refer to as prior patch. If no motion is present in those areas the underlying patch is a repaired version that can be incorporated to achieve temporal consistency of the imposed data information. In addition, if the area behind the impairment is currently moving the flow guidance enables to map the correct patch to the current area and also enforces some natural movement and again visual consistency over time.

Next, we compute the patch priorities according to (6.23). Now, to find a matching patch, we impose the constraint that the found patch $p_{\overline{q}}$ must also be similar to the prior patch $\overline{\Theta}_p$. In addition, we add the weight γ to parametrize the influence of the previous patch. Hence, (6.26) changes to

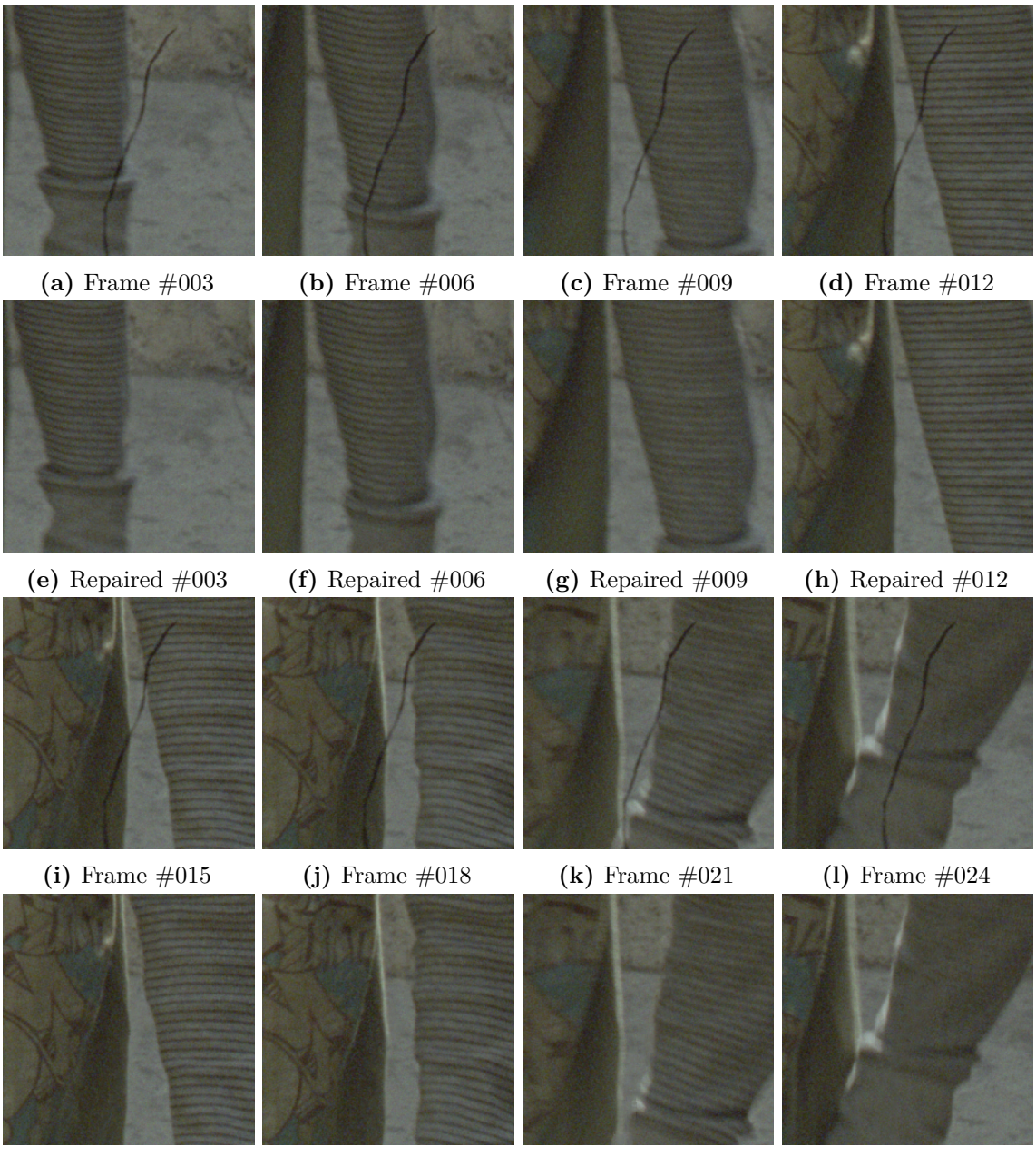
$$\Theta_{\overline{q}} = \underset{\Theta_q \in \mathcal{I}_s}{\arg\min}\ \ (1-\gamma)\, SSD(\Theta_p, \Theta_q) + \gamma\ SSD\left(\overline{\Theta}_p, \Theta_q\right)\,. \tag{6.31}$$

For the results shown in the following section we set $\gamma = 0.2$ for imposing the temporal consistency of the patches. For inserting missing pixels from the found patch we utilize the approach presented in (6.29).

6.2.3. Result

The example that we use to demonstrate the capability of the optical flow guided exemplar-based inpainting model seems not very challenging at first because of the comparably thin impairment. On closer consideration it becomes clear that the restoration of this sequence is anything else than a simple restoration task. First, the impairment is static over the whole sequence. Second, within the surroundings of the defect, objects perform complex motion. Third, the texture of the moving objects, especially the clothes' striped pattern is difficult to repair in such a way that the observer does not recognize the filled area. In addition, due to the fast movement, motion blur appears and even the stripe pattern changes (see *e.g.* Figure 6.12p). Finally, the motion is constantly changing as the object changes its directions and also the camera is slightly moving. Altogether, this task becomes a very difficult restoration problem and, to emphasize this once again, the restoration should be performed fully automatically. By applying the presented approach this restoration task

can be performed and the viewer is not able to notice any repaired region when looking either at the still images or the whole video sequence. Also for repairing single images, the small extensions presented in Section 6.2.1 improves the quality and robustness of those approaches.



(a) Frame #003 **(b)** Frame #006 **(c)** Frame #009 **(d)** Frame #012

(e) Repaired #003 **(f)** Repaired #006 **(g)** Repaired #009 **(h)** Repaired #012

(i) Frame #015 **(j)** Frame #018 **(k)** Frame #021 **(l)** Frame #024

(m) Repaired #015 **(n)** Repaired #018 **(o)** Repaired #021 **(p)** Repaired #024

Figure 6.12.: Details of the impaired and filled areas from the sequence shown in Figure 6.10.

6.2.4. Discussion

In this section we presented an optical flow guided variant of exemplar-based inpainting. It shows that also for methods that are not formulated in a variational framework, adding temporal information increases the results' quality when used for video processing. For the showcase results of restoration of a static artifact the methods yield an appropriate reconstruction and when viewing the sequence, the restored area is not noticed by the viewer. Furthermore, the usage of motion information in a restoration task emphasizes the importance of a robust algorithm for motion estimation.

APPENDIX
A

OPTICAL FLOW: A COMPARISON

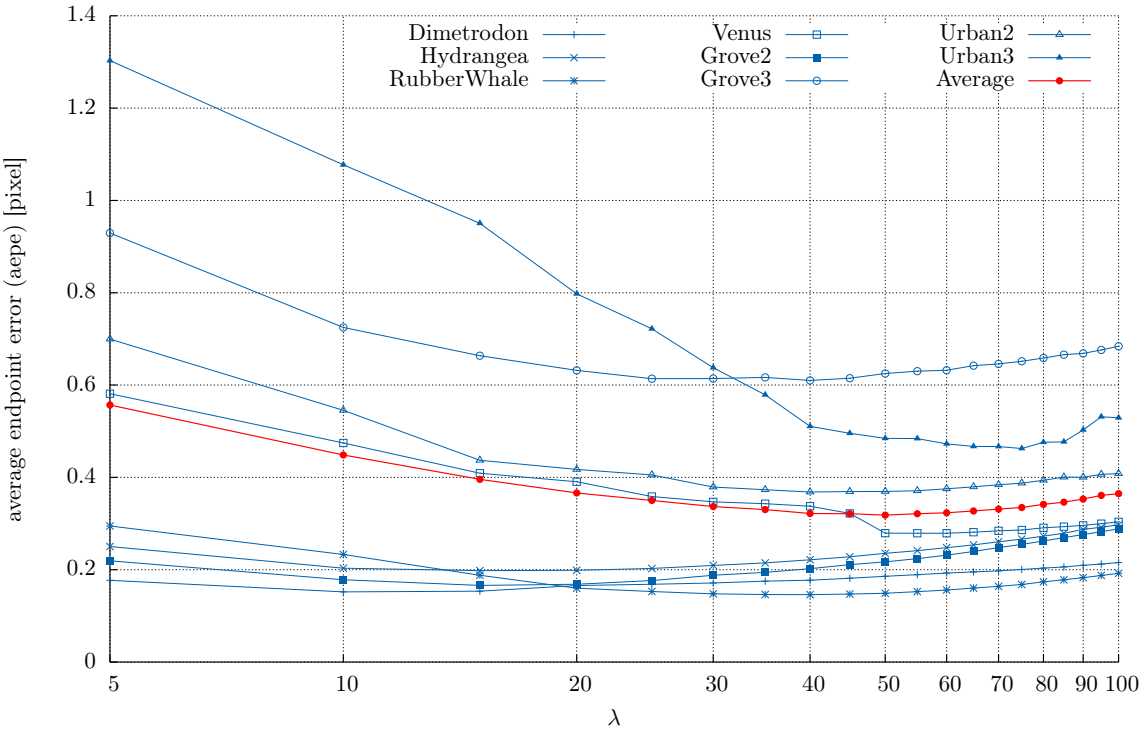


Figure A.1.: Huber-ℓ_1

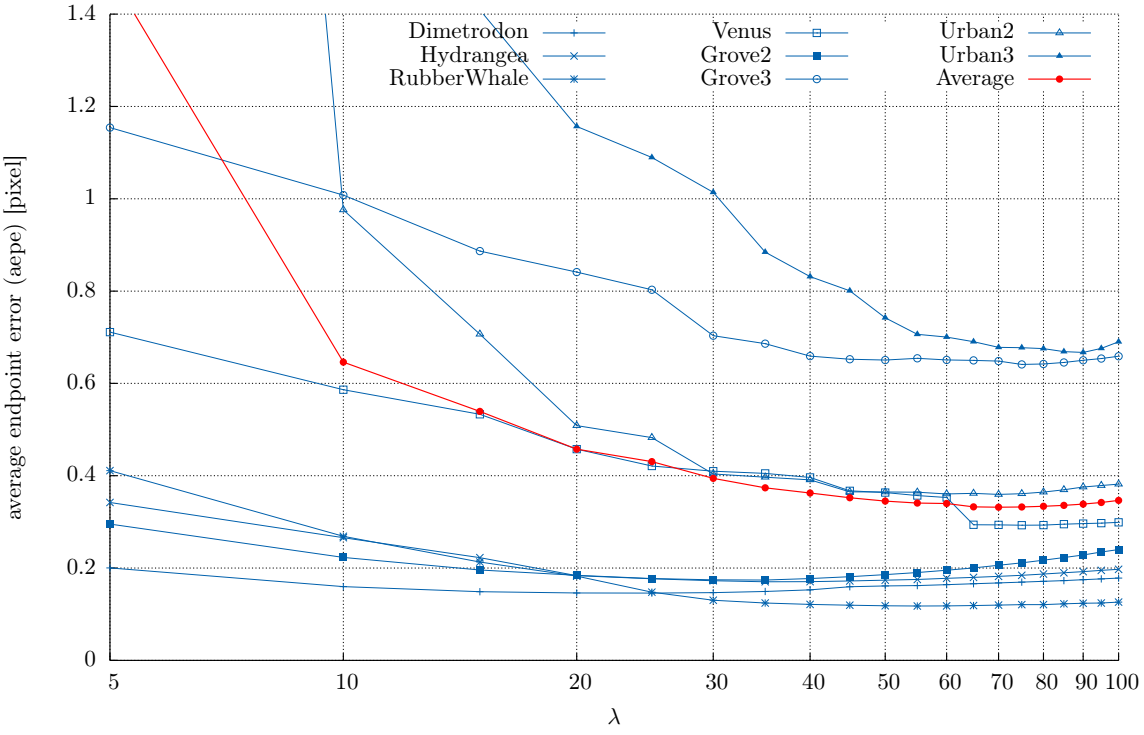


Figure A.2.: Huber-ℓ_1; structure-texture decomposed input

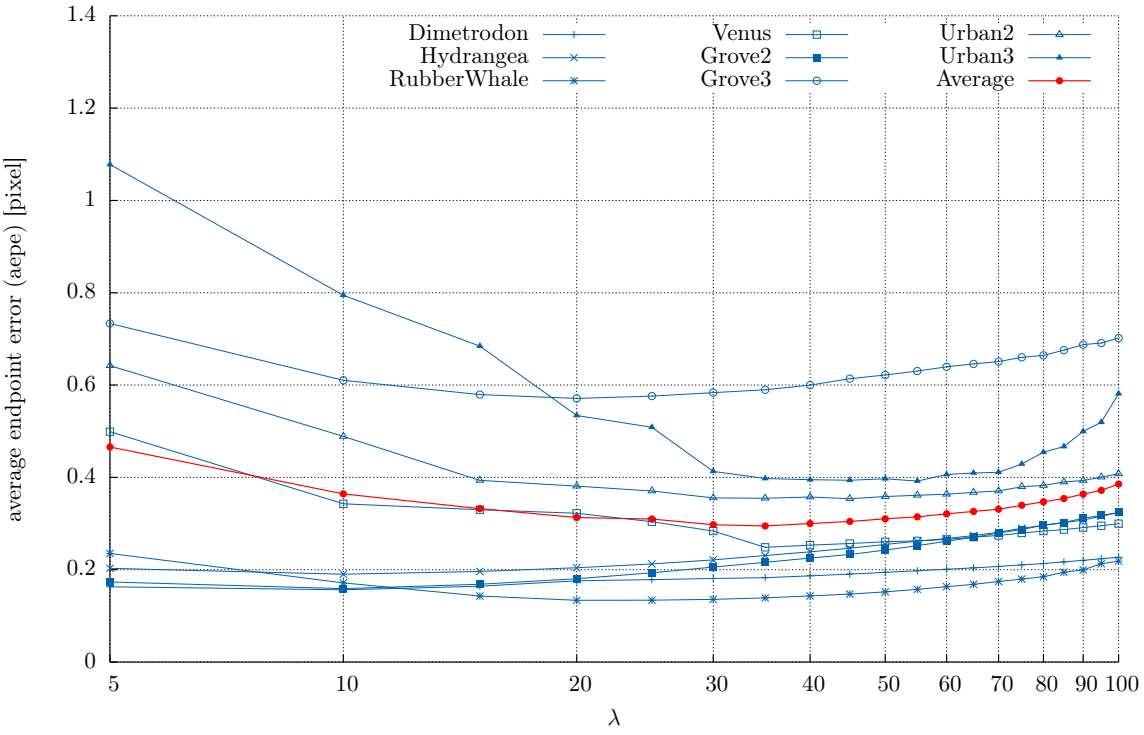


Figure A.3.: weighted Huber-ℓ_1

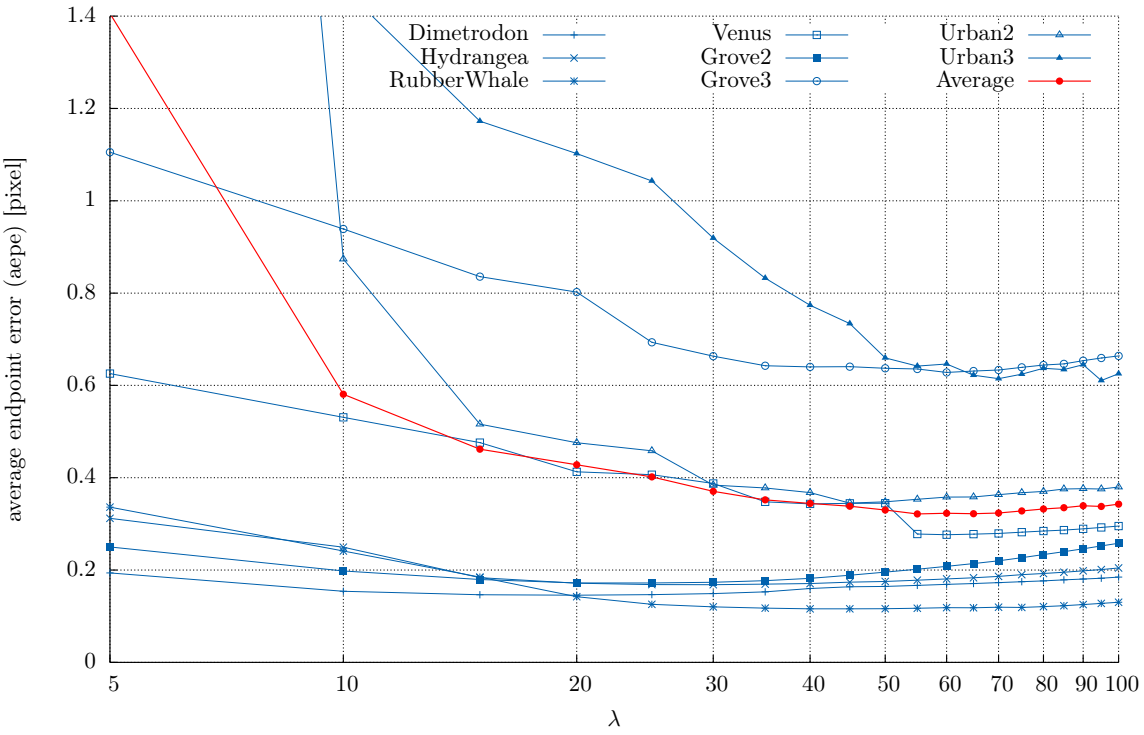


Figure A.4.: weighted Huber-ℓ_1; structure-texture decomposed input

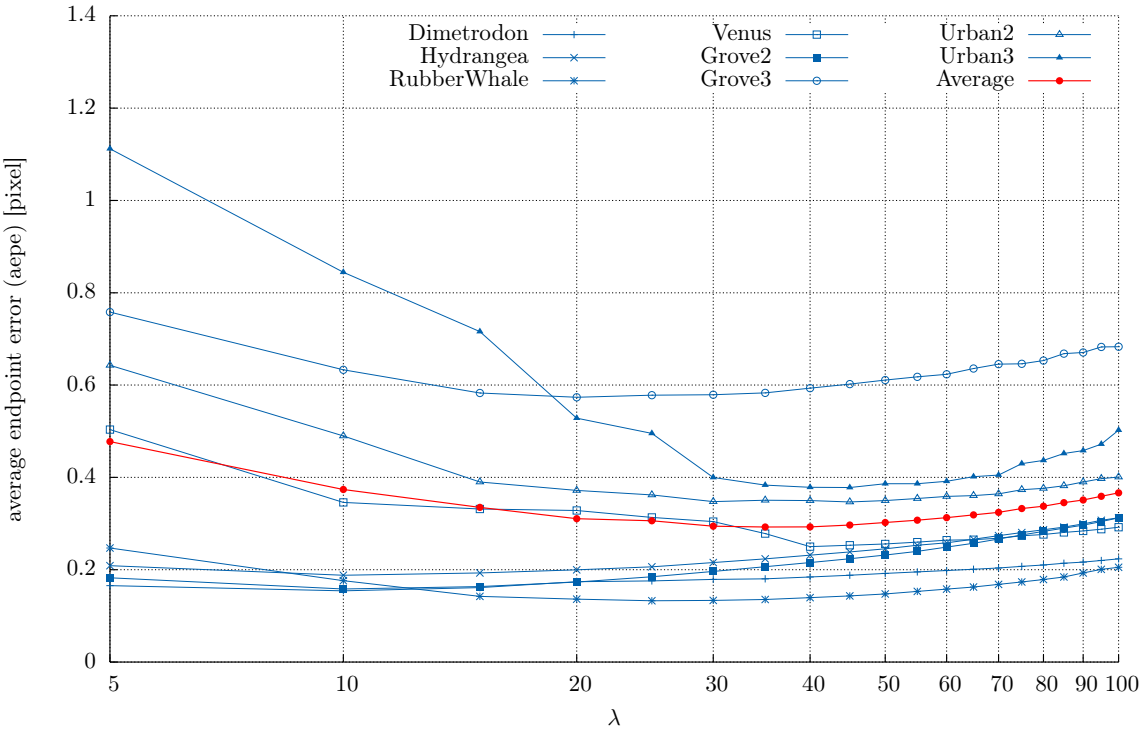


Figure A.5.: directed Huber-ℓ_1

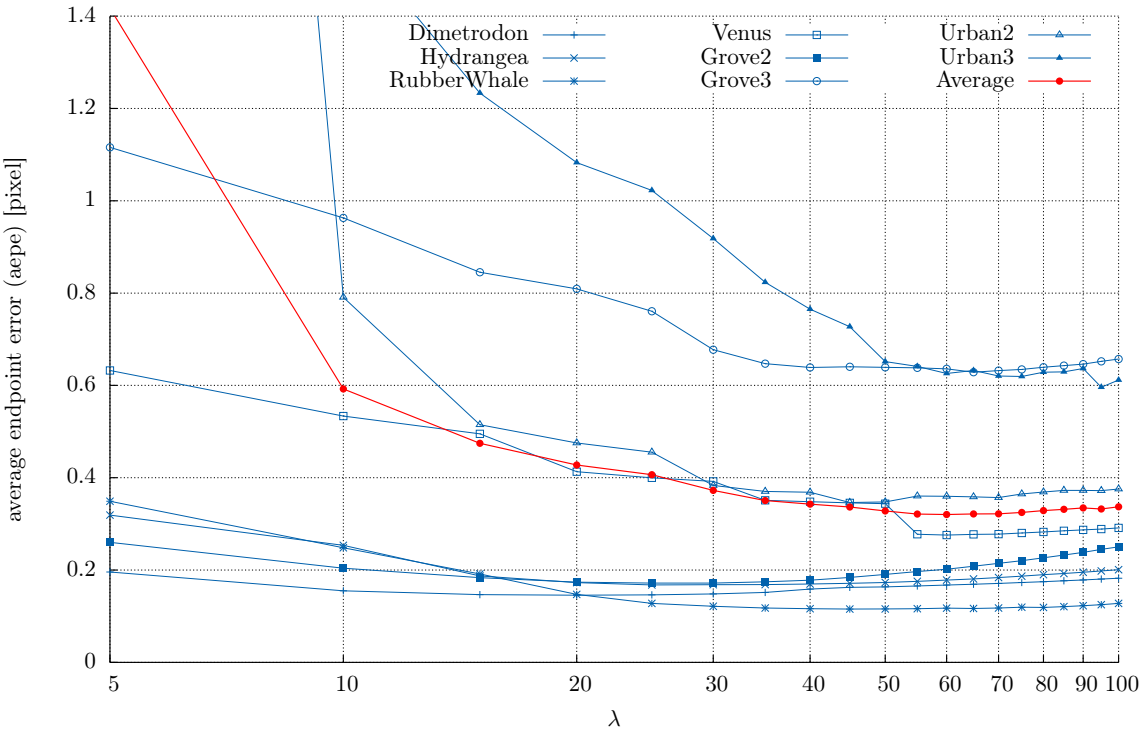


Figure A.6.: directed Huber-ℓ_1; structure-texture decomposed input

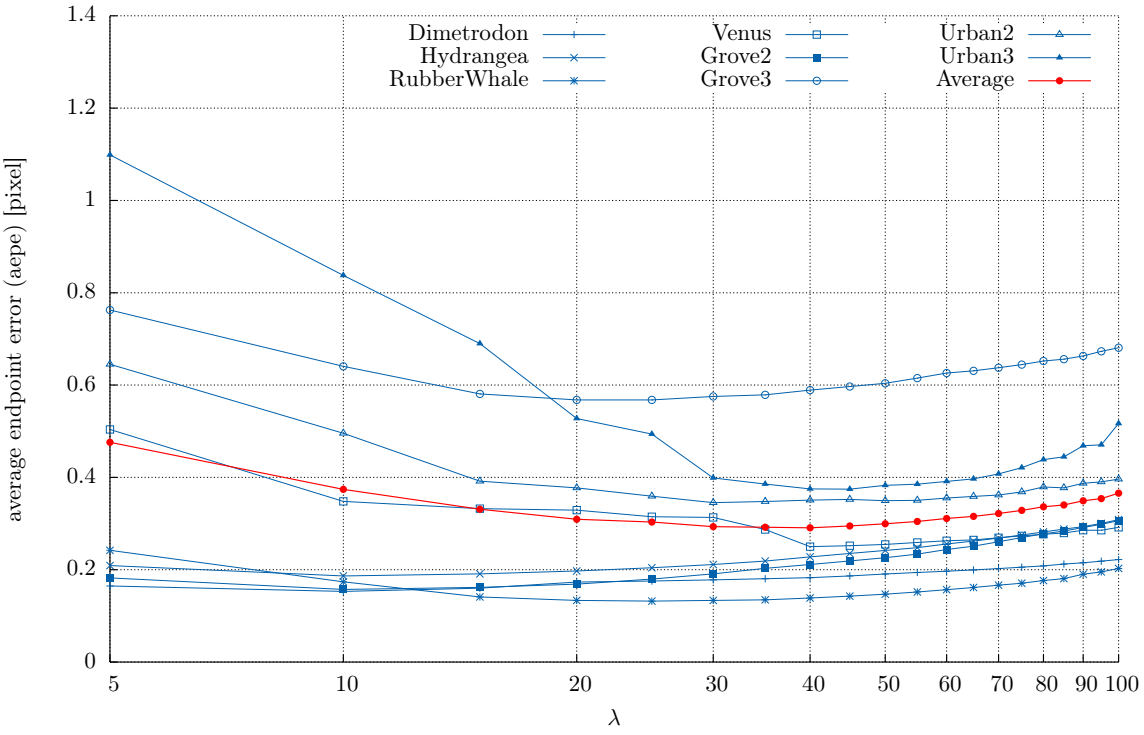


Figure A.7.: tensor directed Huber-ℓ_1

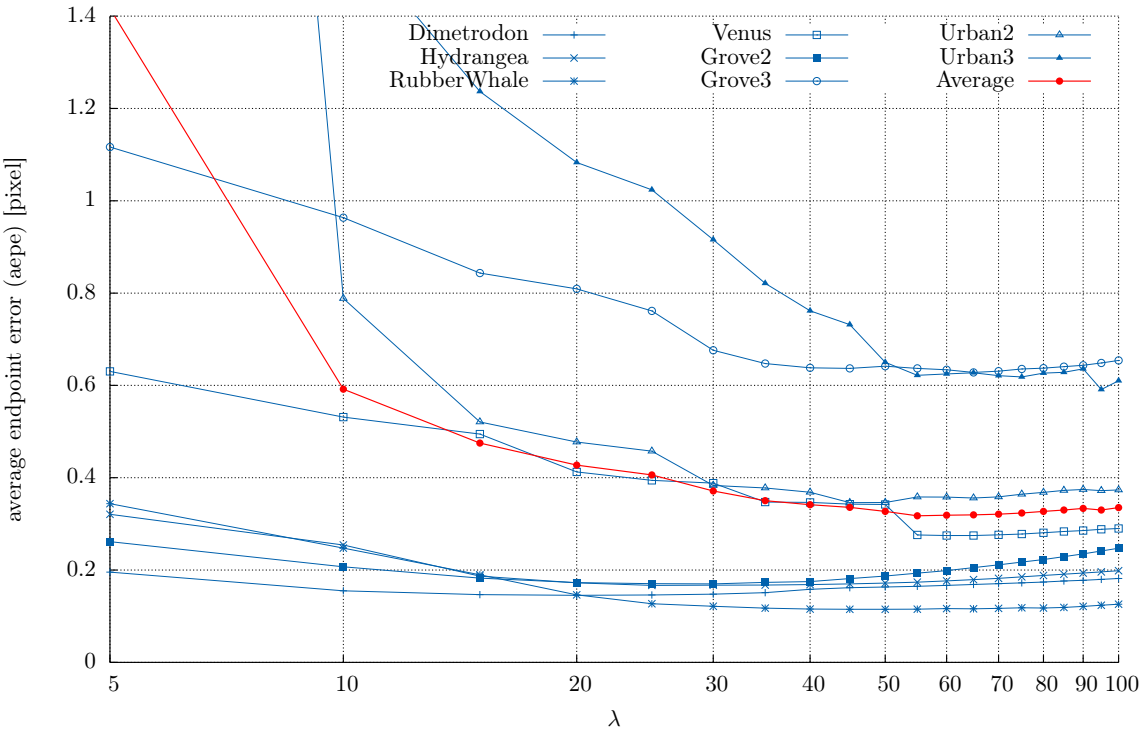


Figure A.8.: tensor directed Huber-ℓ_1; structure-texture decomposed input

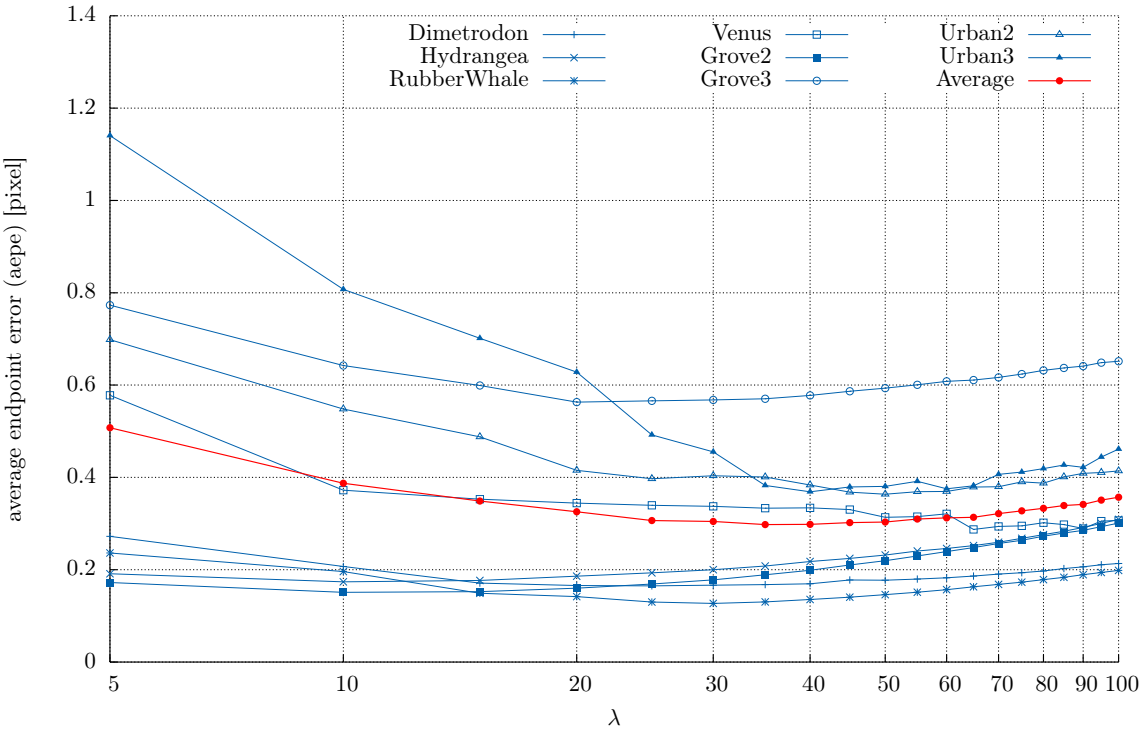


Figure A.9.: Nonlocal Huber-ℓ_1 ($w = 2$)

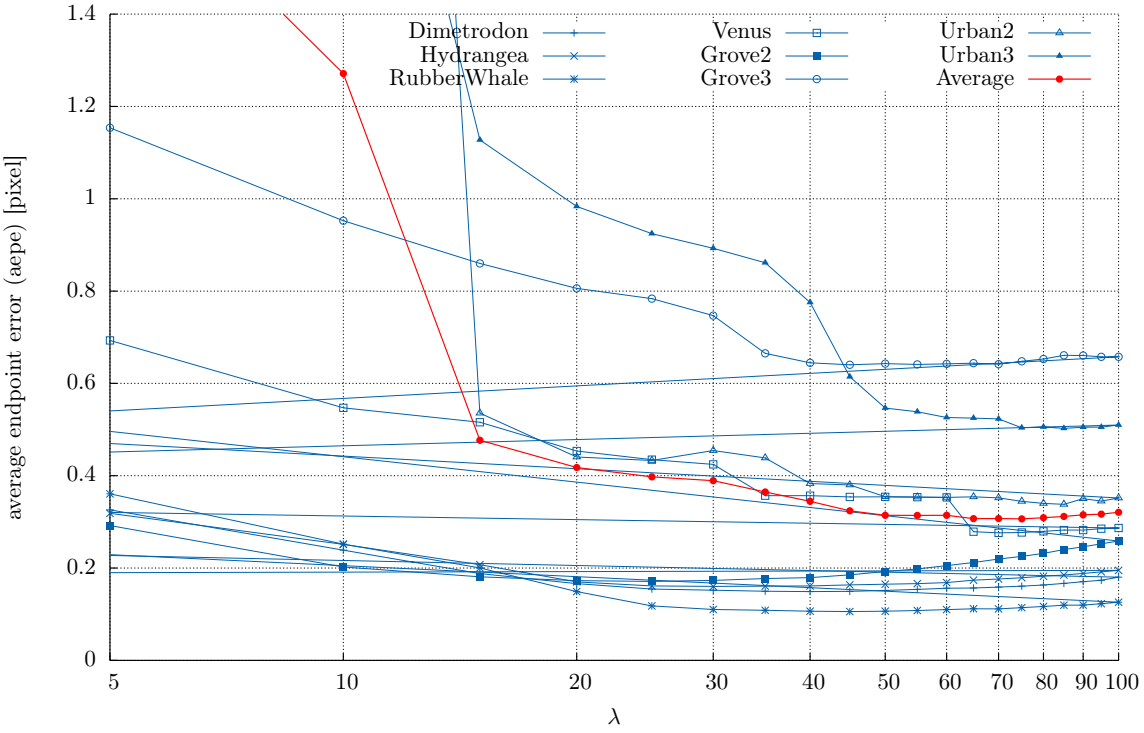


Figure A.10.: Nonlocal Huber-ℓ_1 ($w = 2$); structure-texture decomposed input

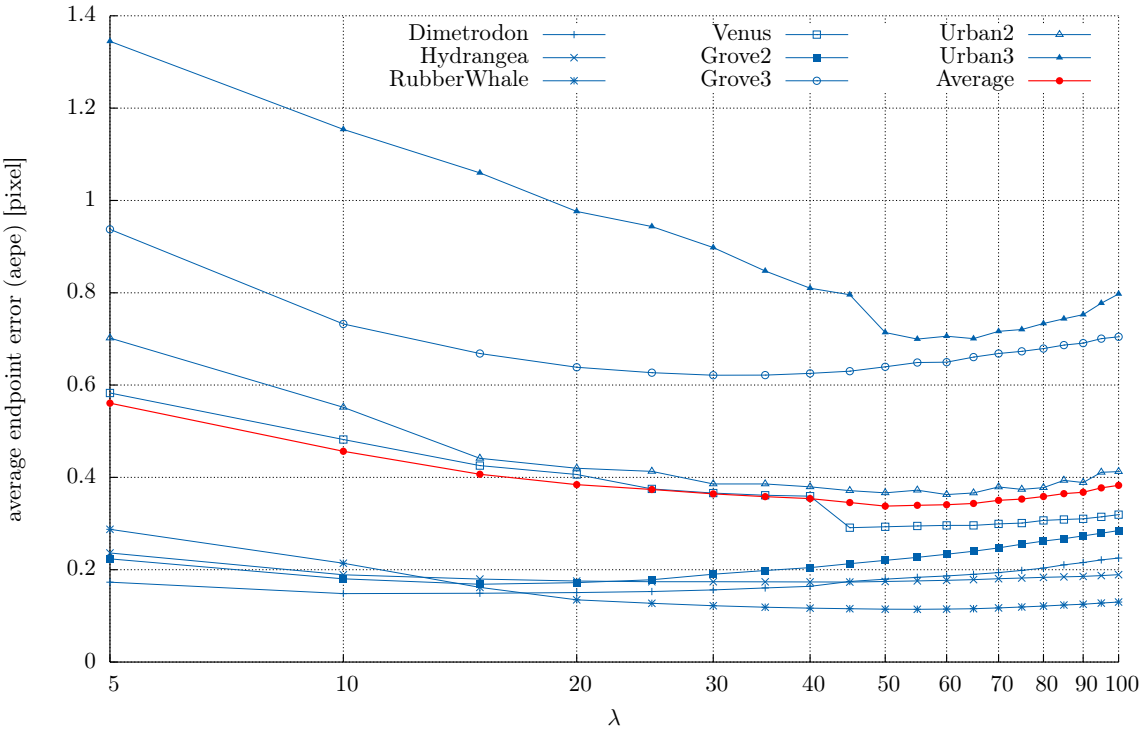


Figure A.11.: Huber-ℓ_1; compensating data violations

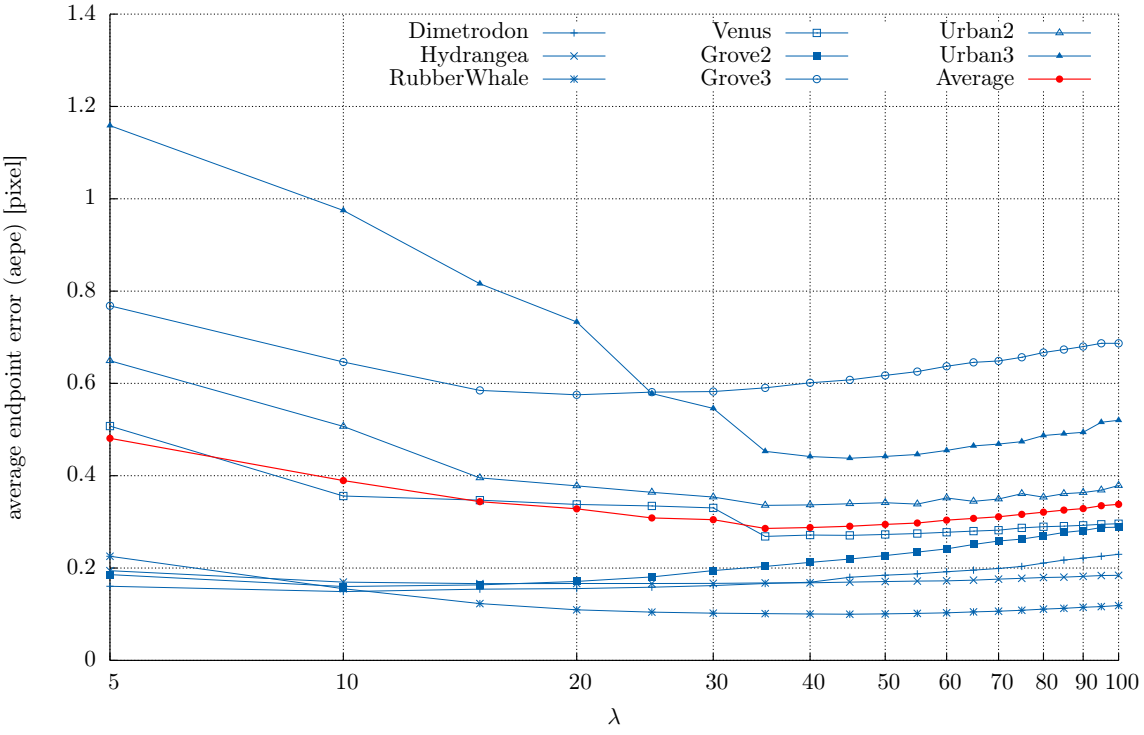


Figure A.12.: tensor directed Huber-ℓ_1; compensating data violations

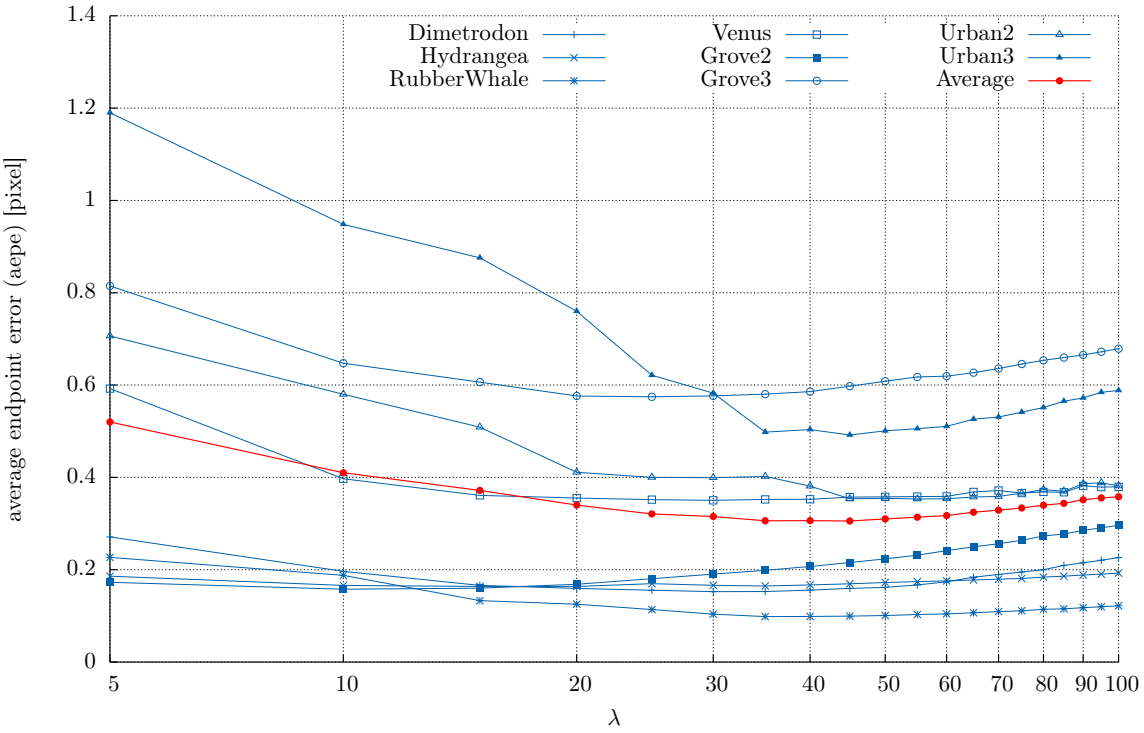


Figure A.13.: nonlocal Huber-ℓ_1 ($w = 2$); compensating data violations

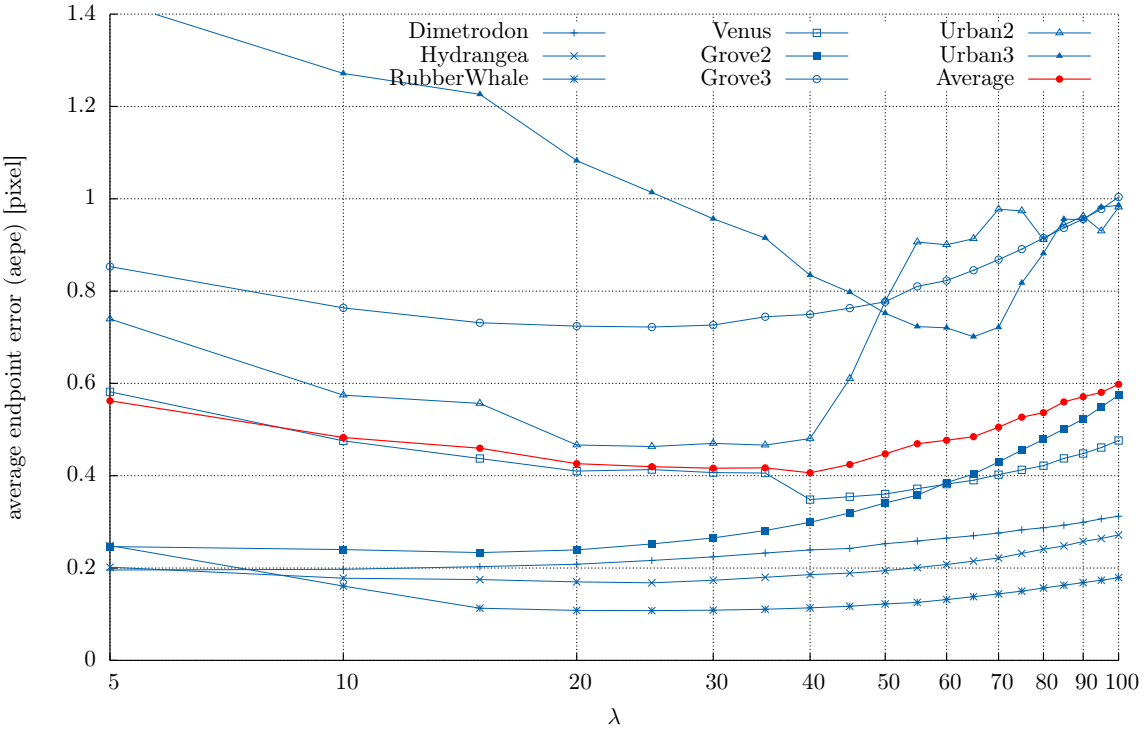


Figure A.14.: Huber-grad

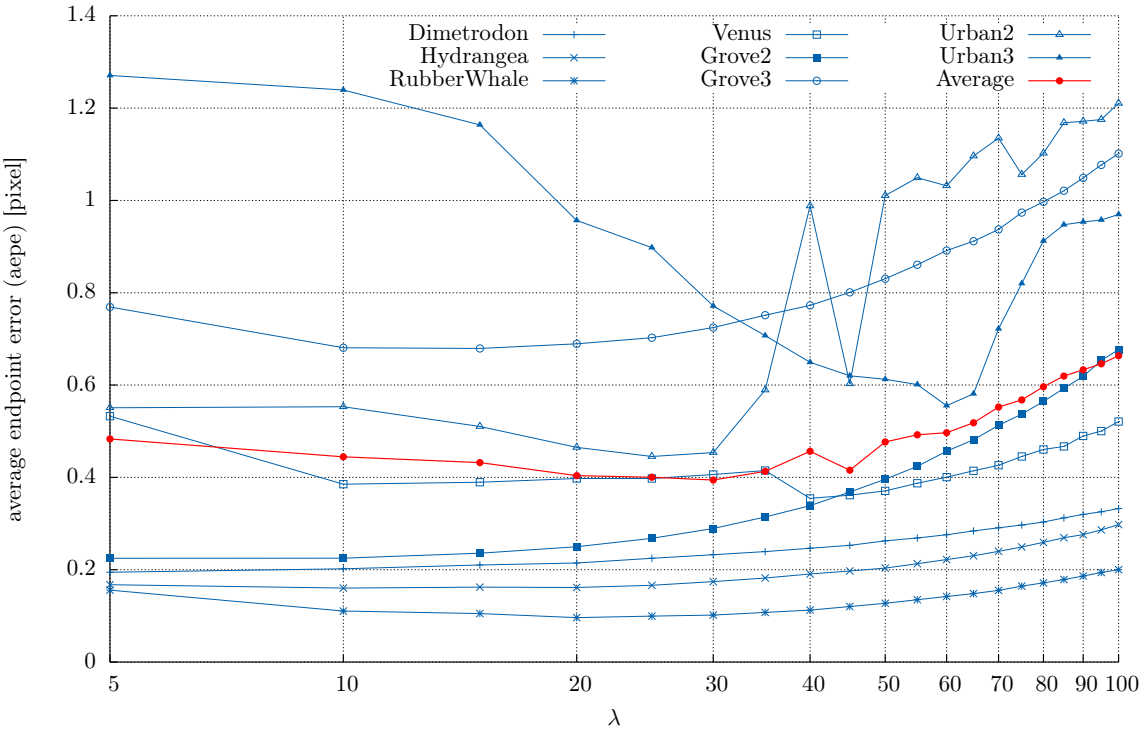


Figure A.15.: tensor directed Huber-grad

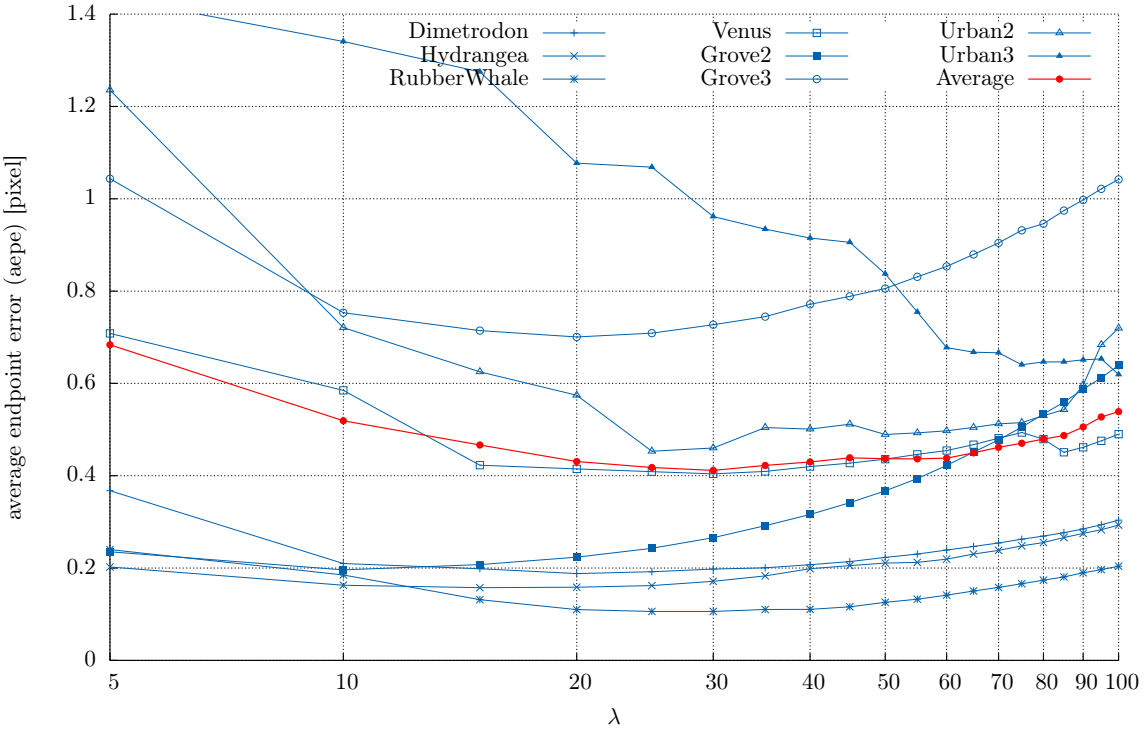


Figure A.16.: nonlocal Huber-grad

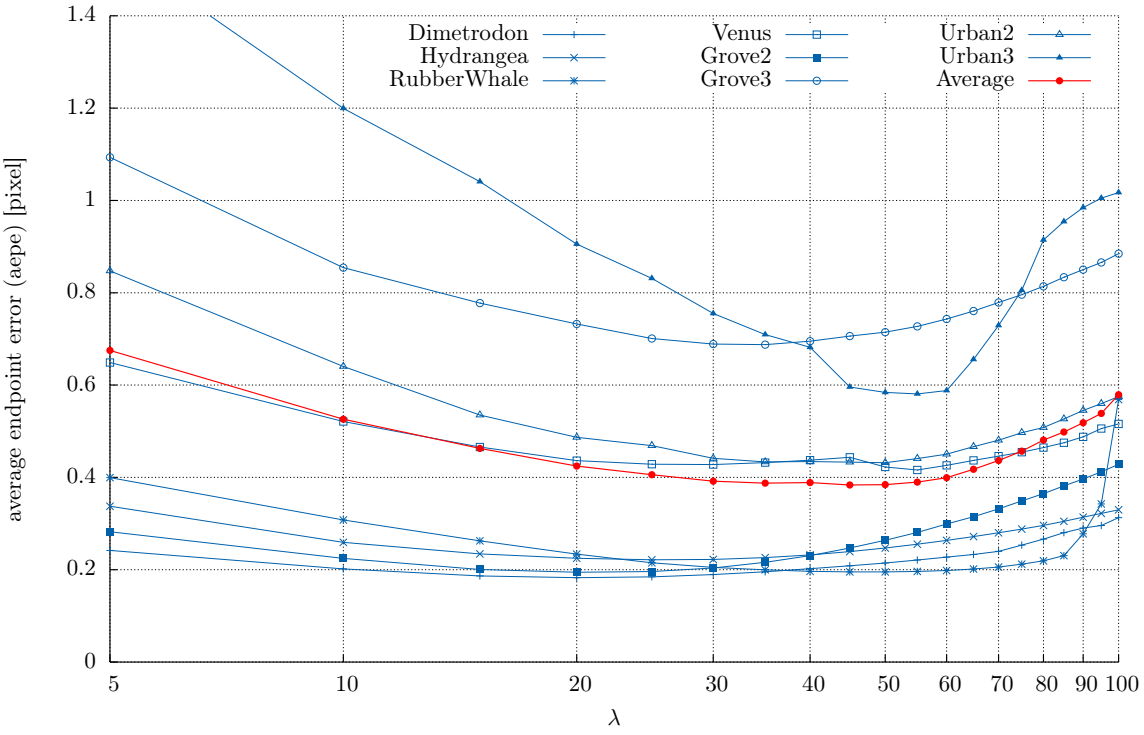


Figure A.17.: Huber-Quadfit (AD)

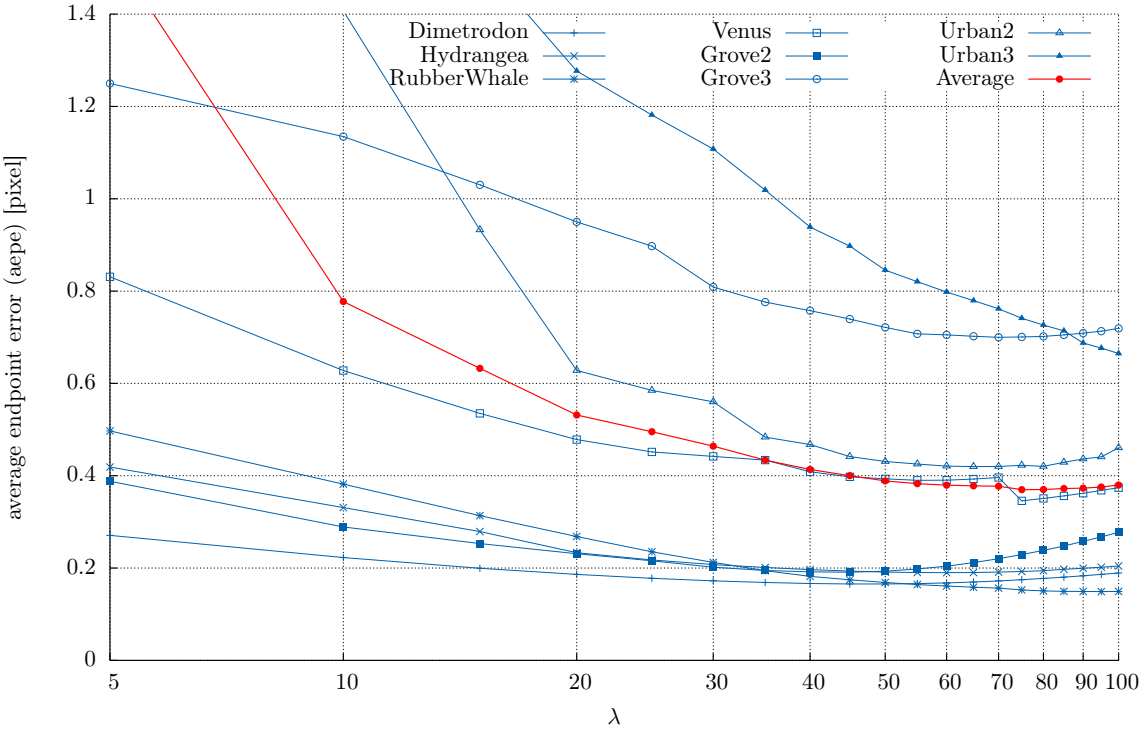


Figure A.18.: Huber-Quadfit (AD); structure-texture decomposed input

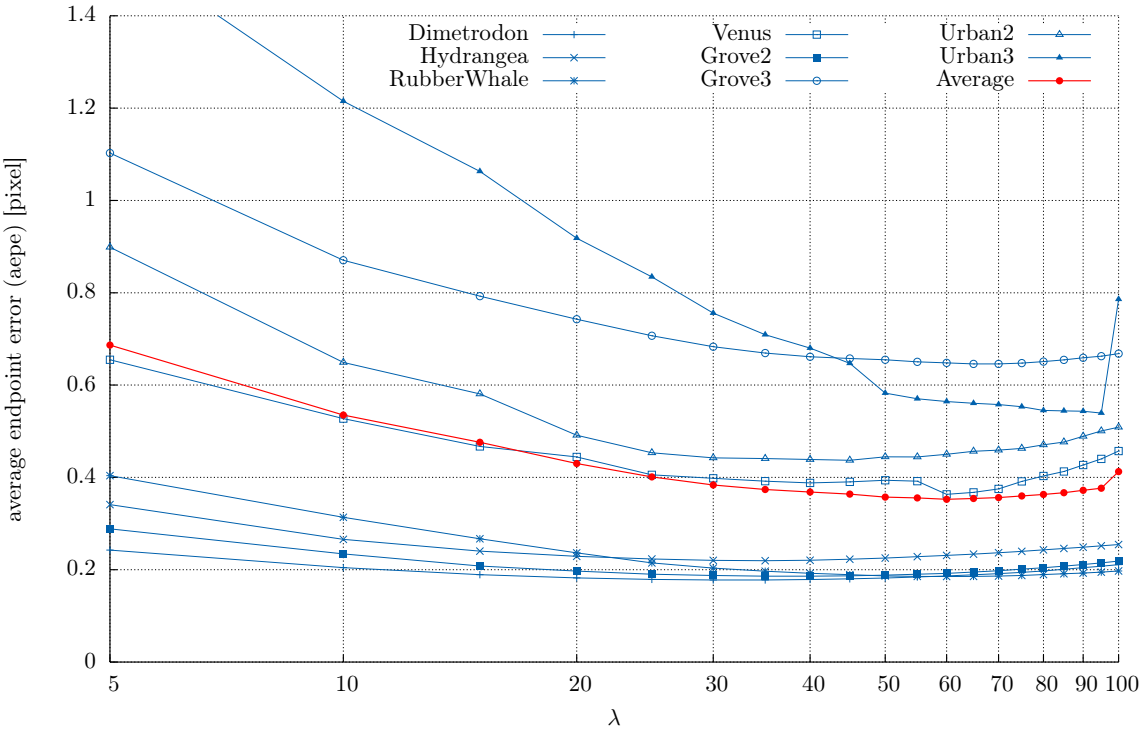


Figure A.19.: Huber-Quadfit (AD)

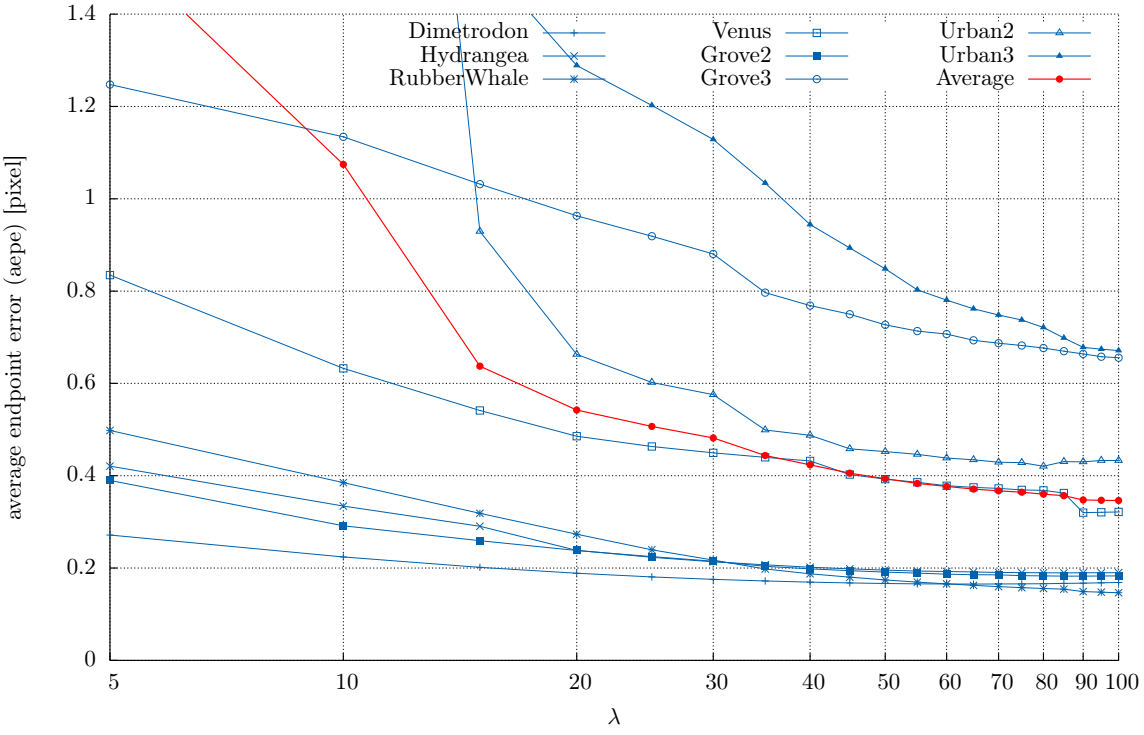


Figure A.20.: Huber-Quadfit (SAD3); structure-texture decomposed input

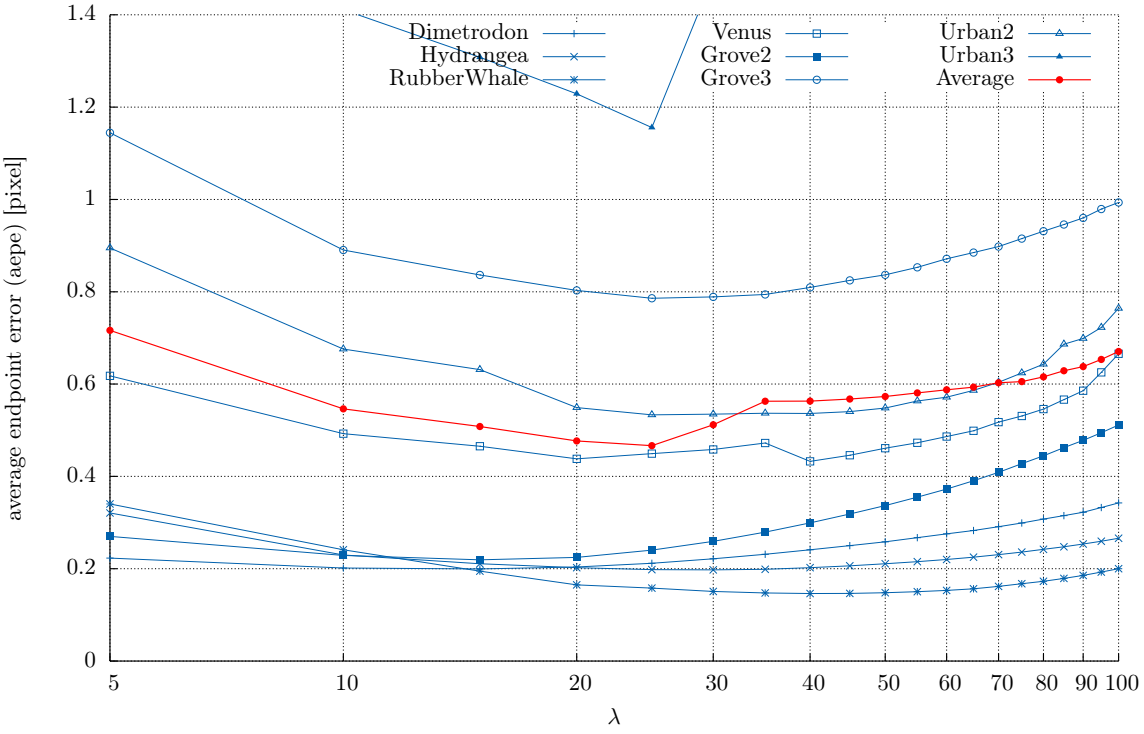


Figure A.21.: Huber-Quadfit (GRAD)

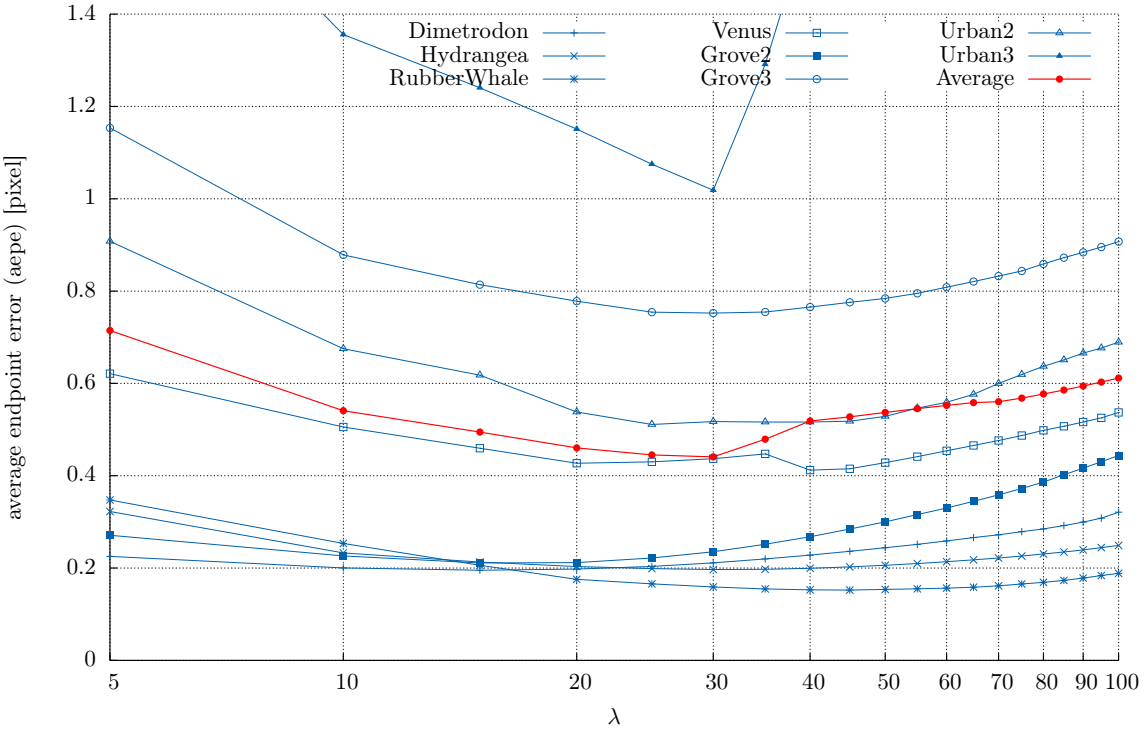


Figure A.22.: Huber-Quadfit (GRAD+INTENSITY)

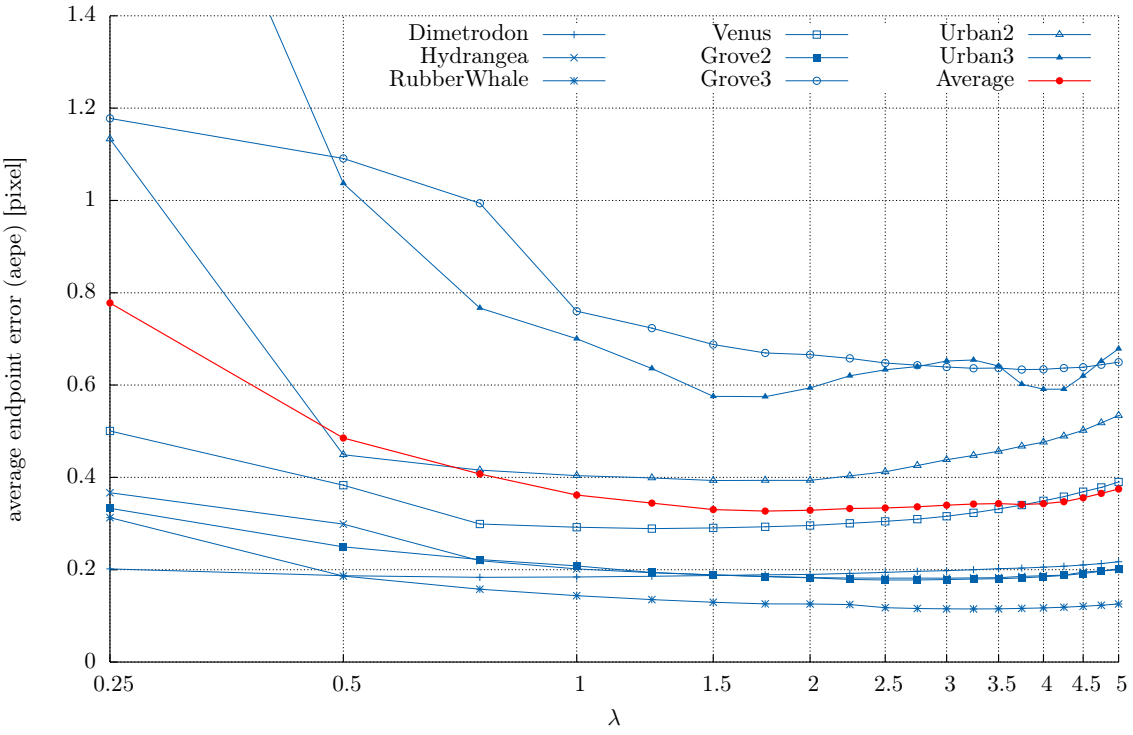


Figure A.23.: Huber-Quadfit (NCC3)

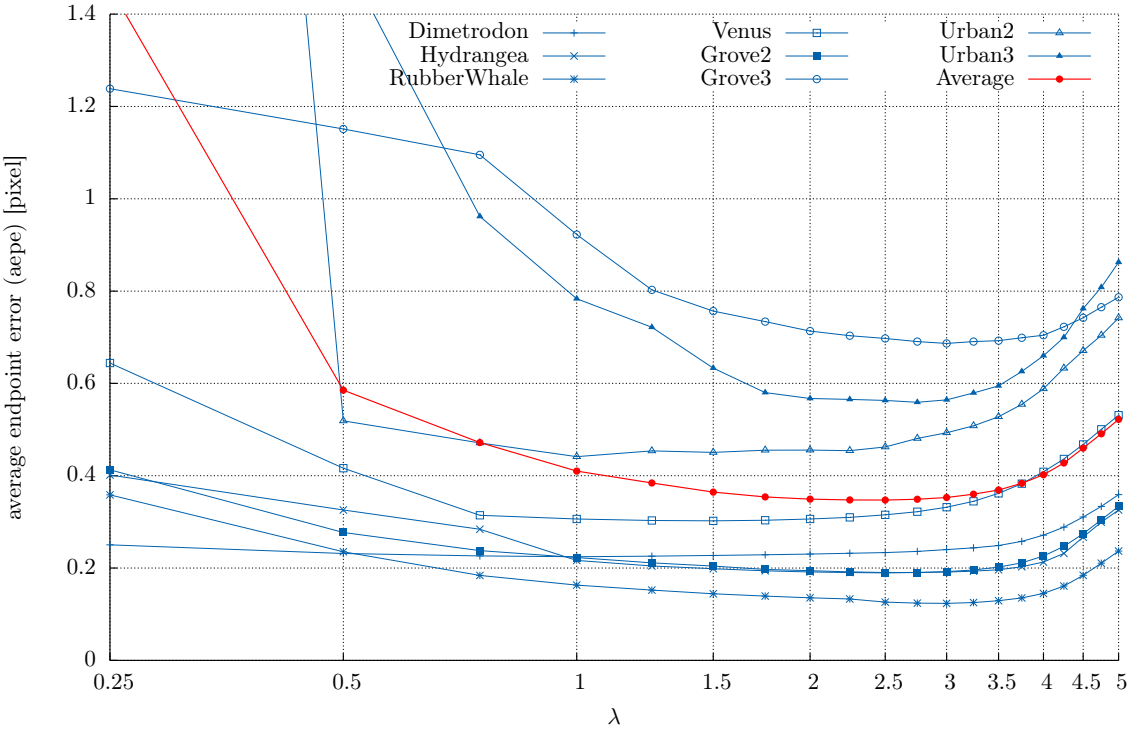


Figure A.24.: Huber-Quadfit (normalized GRAD)

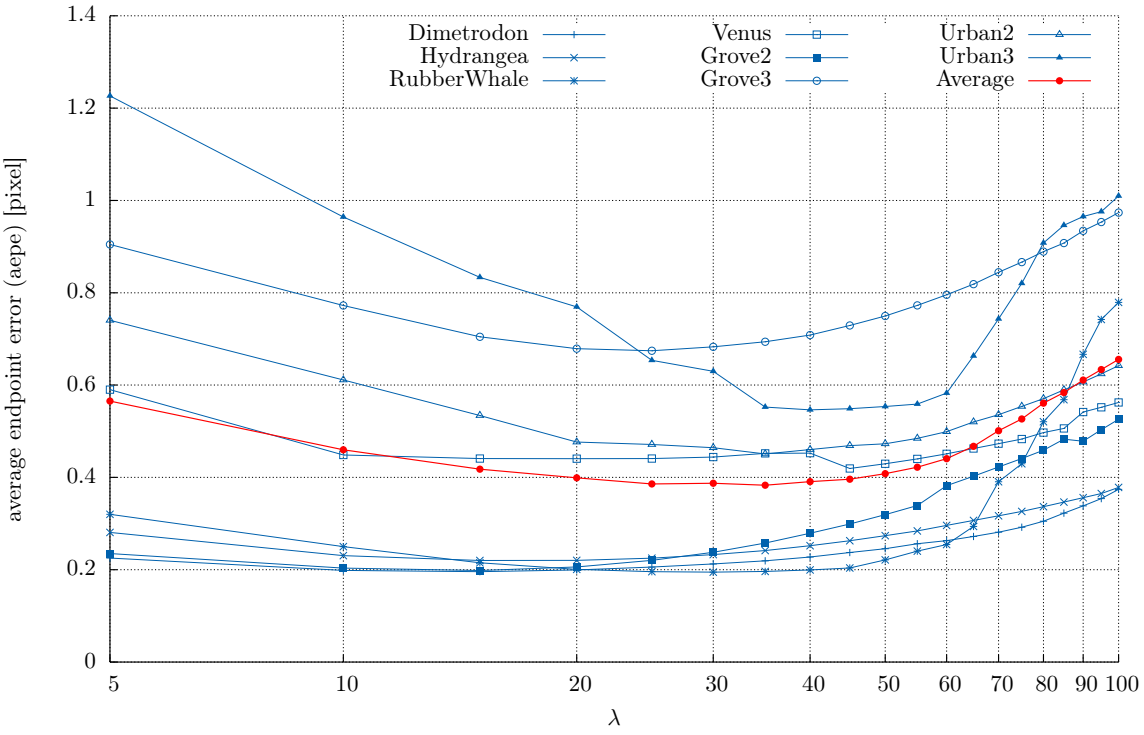


Figure A.25.: tensor directed Huber-Quadfit (AD)

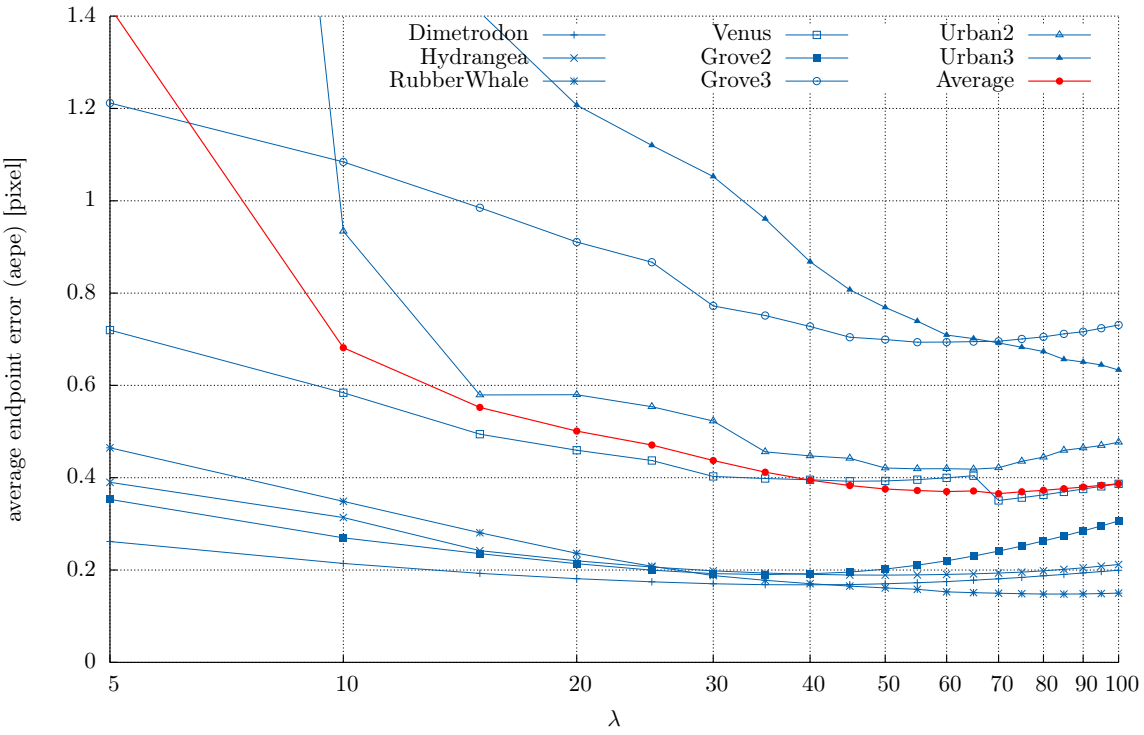


Figure A.26.: tensor directed Huber-Quadfit (AD); structure-texture decomposed input

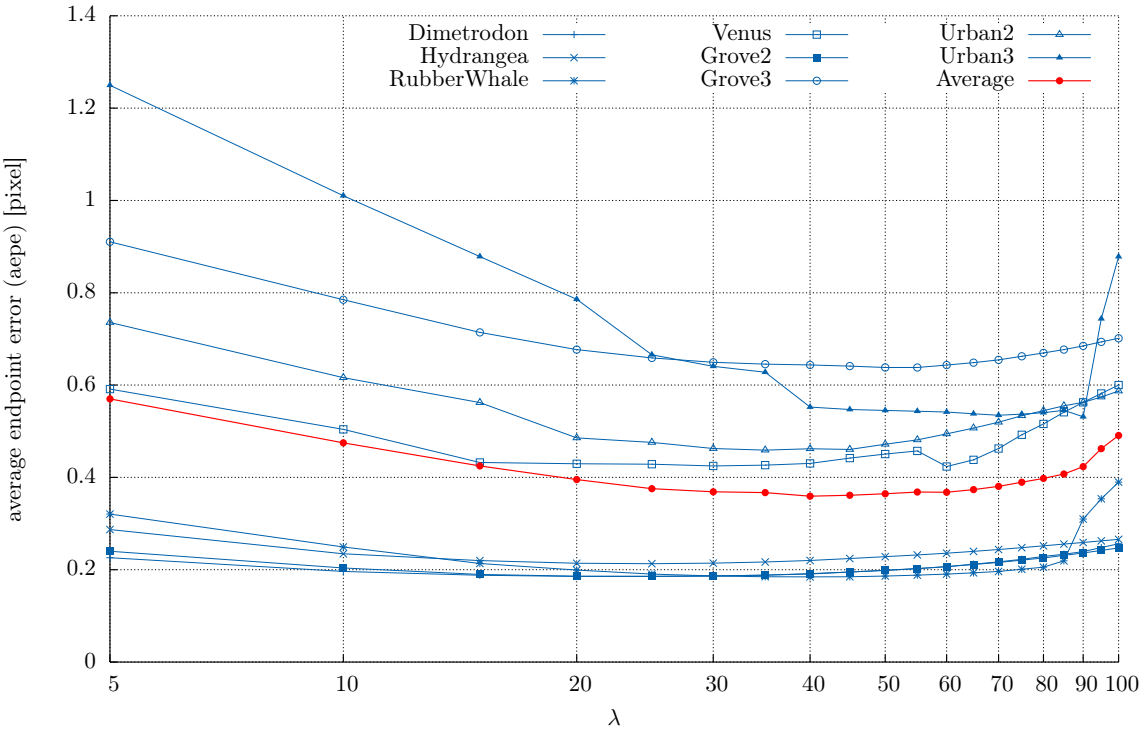


Figure A.27.: tensor directed Huber-Quadfit (SAD3)

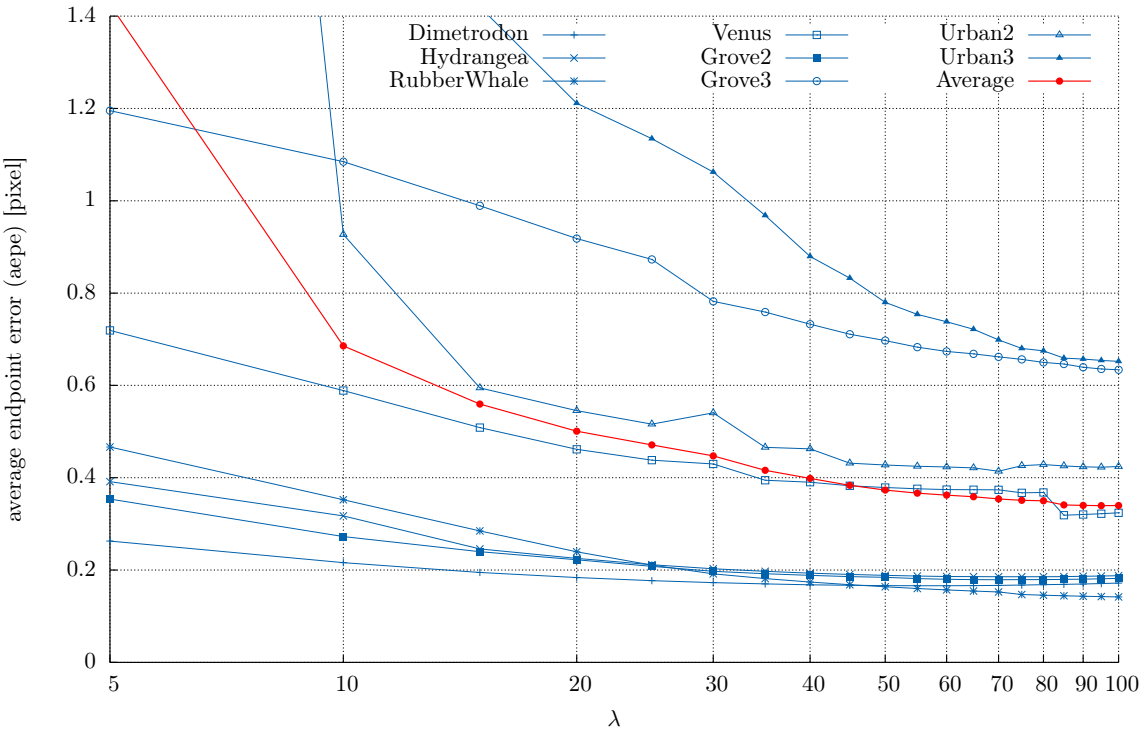


Figure A.28.: tensor directed Huber-Quadfit (SAD3); structure-texture decomposed input

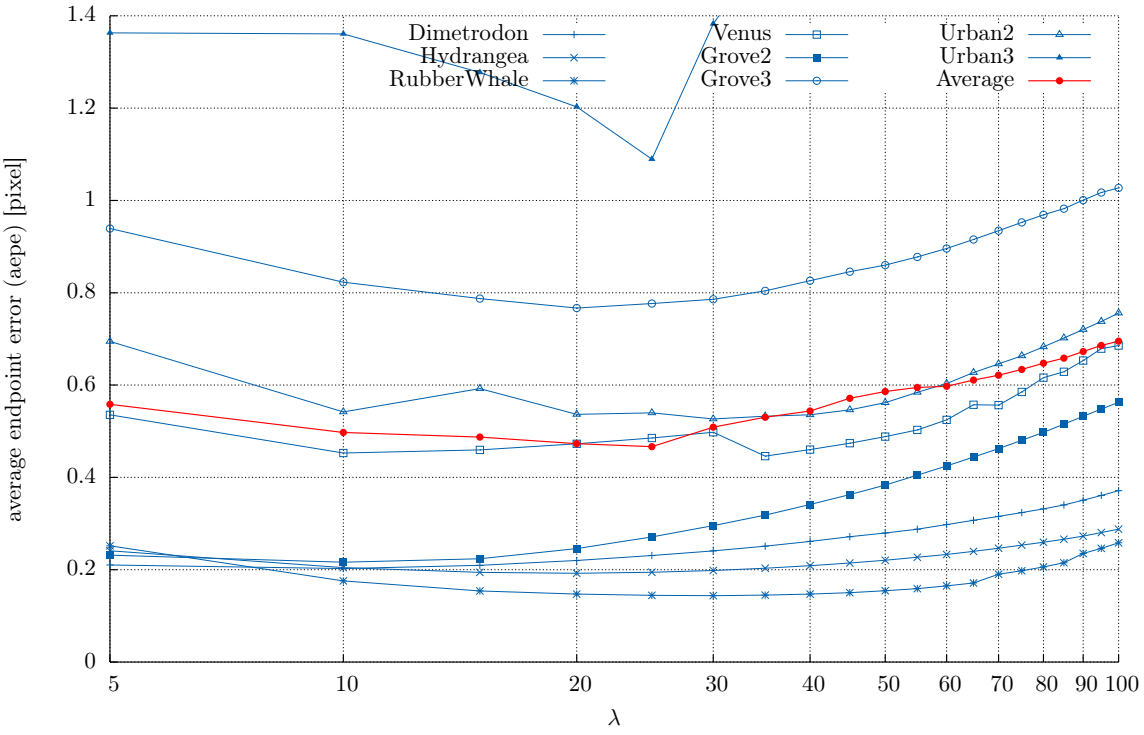


Figure A.29.: tensor directed Huber-Quadfit (GRAD)

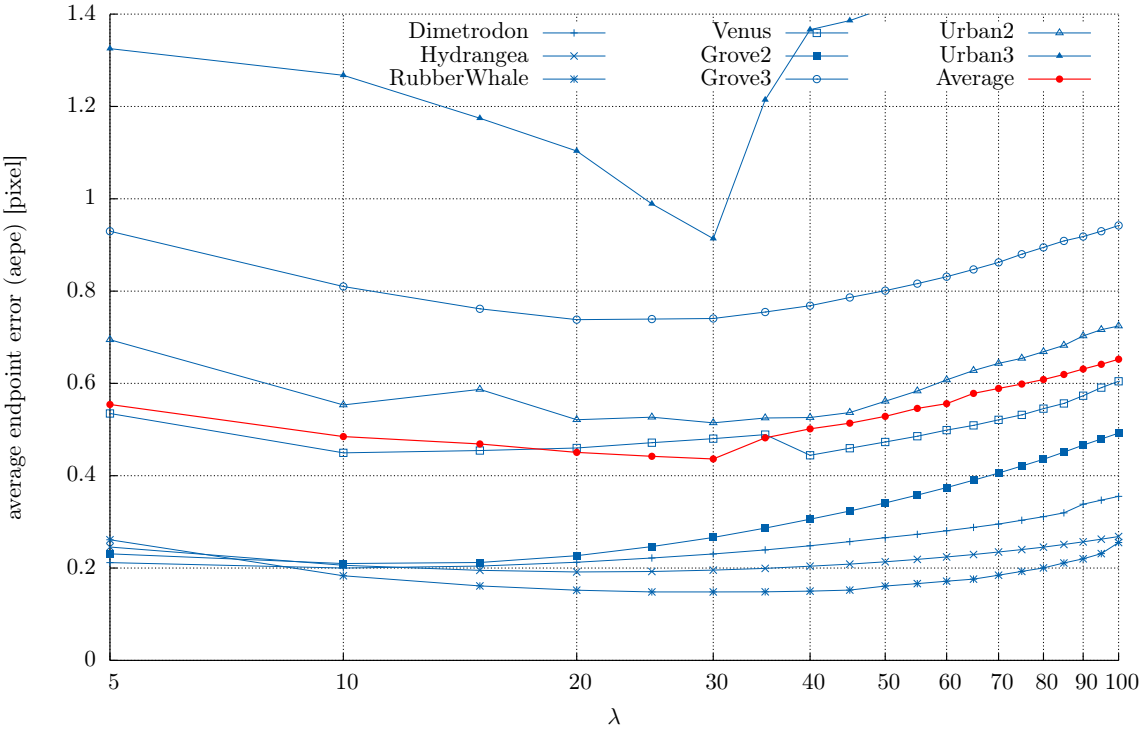


Figure A.30.: tensor directed Huber-Quadfit (GRAD+INTENSITY)

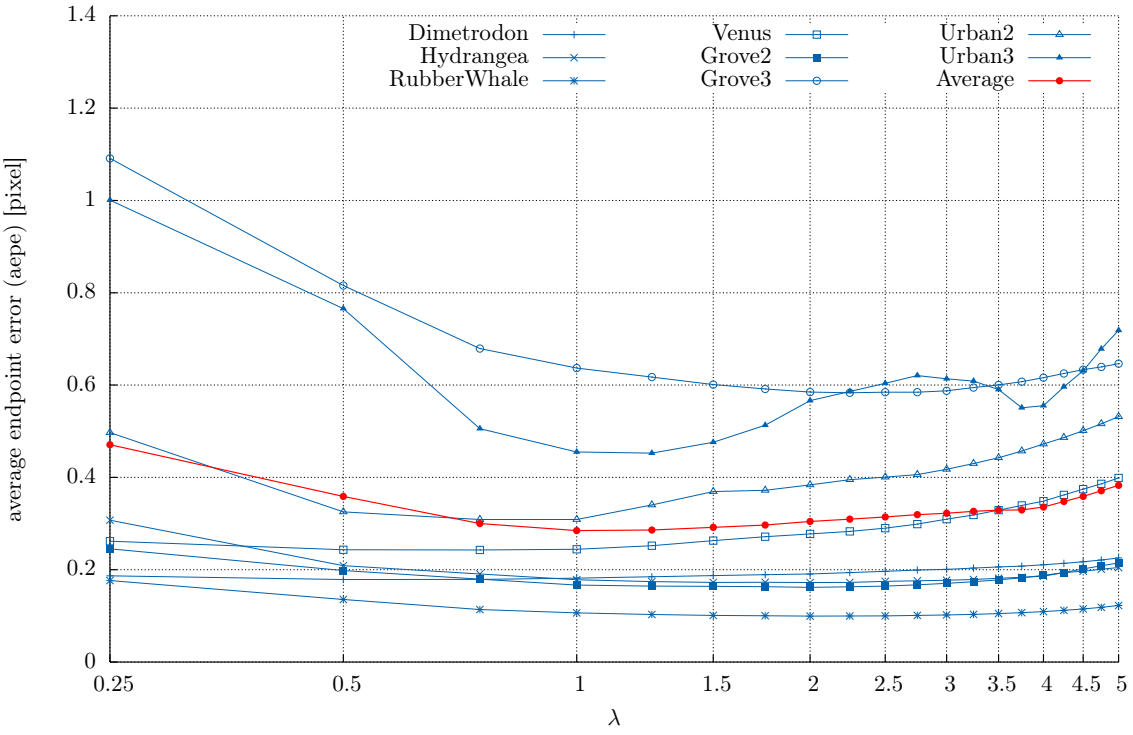


Figure A.31.: tensor directed Huber-Quadfit (NCC3)

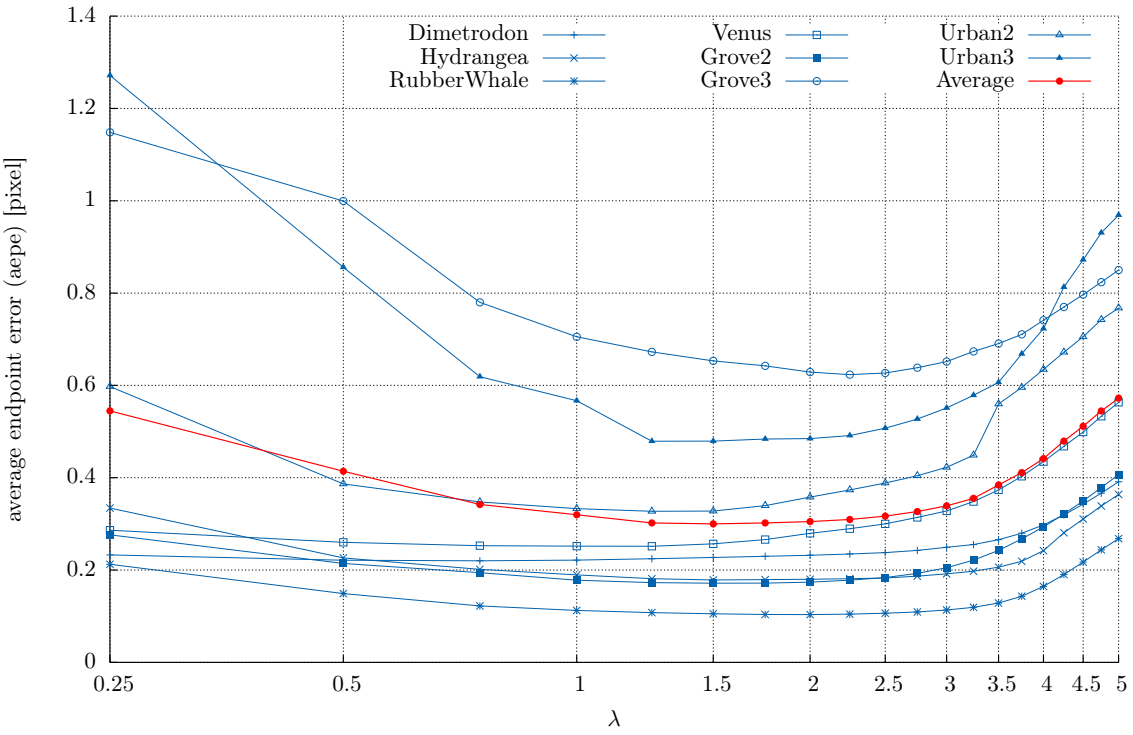


Figure A.32.: tensor directed Huber-Quadfit (normalized GRAD)

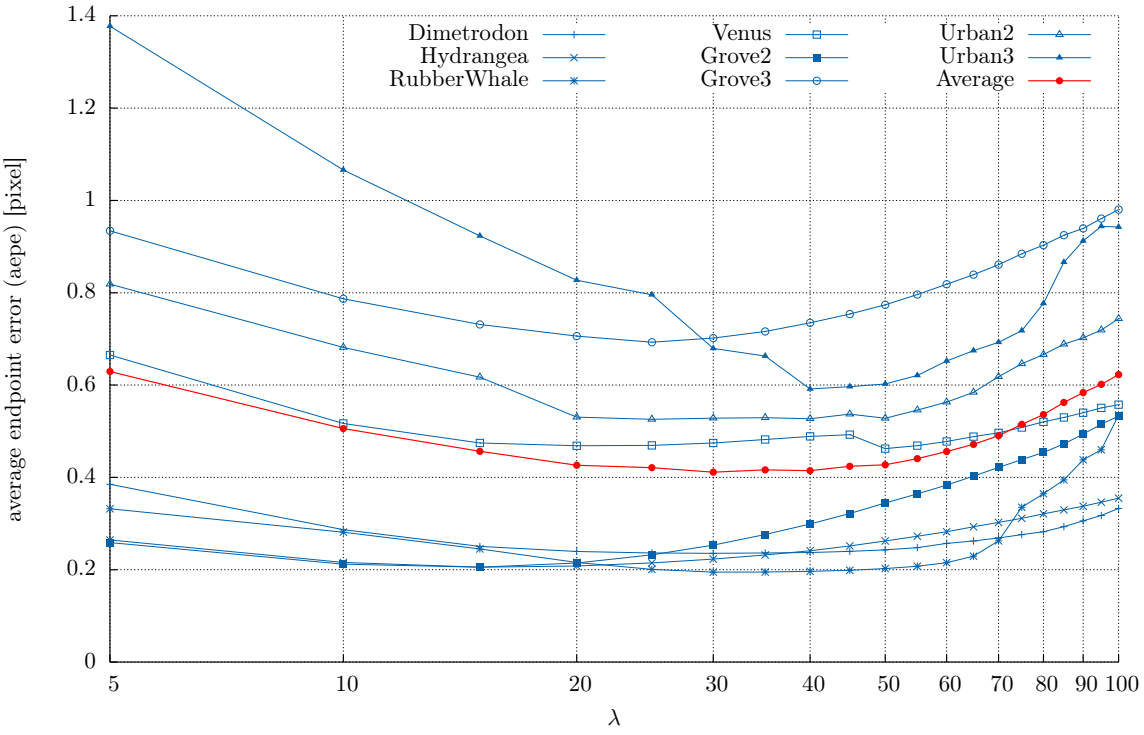


Figure A.33.: nonlocal Huber-Quadfit (AD)

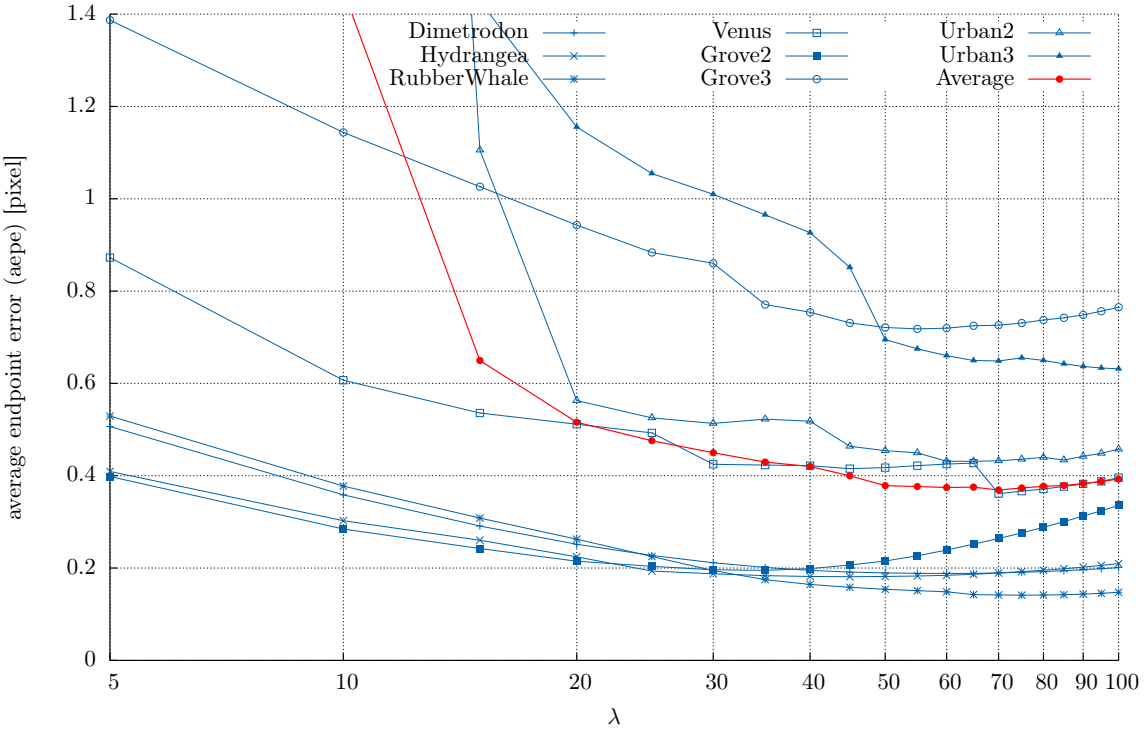


Figure A.34.: nonlocal Huber-Quadfit (AD); structure-texture decomposed input

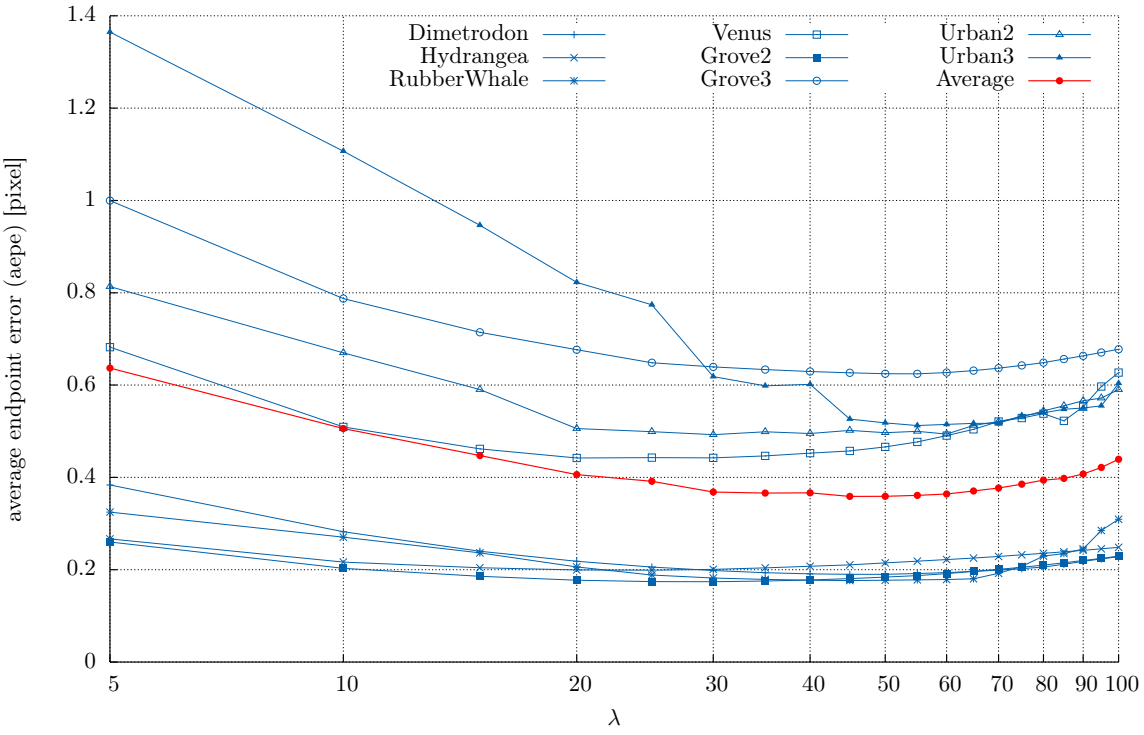


Figure A.35.: nonlocal Huber-Quadfit (SAD3)

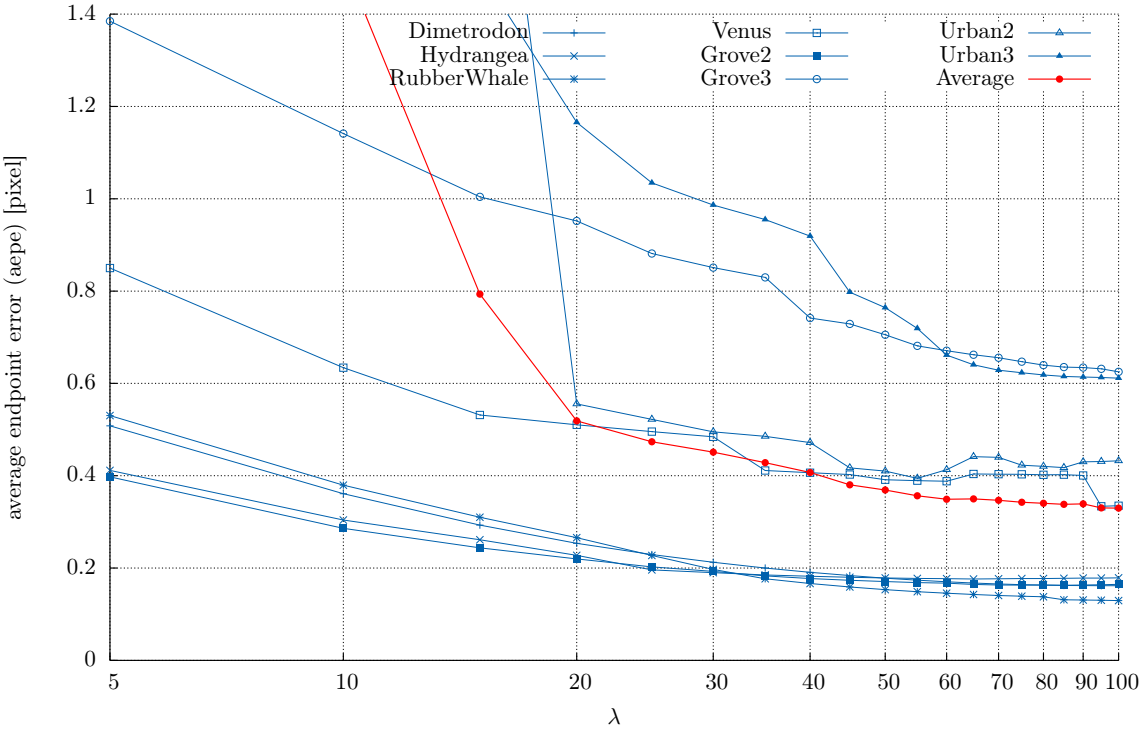


Figure A.36.: nonlocal Huber-Quadfit (SAD3); structure-texture decomposed input

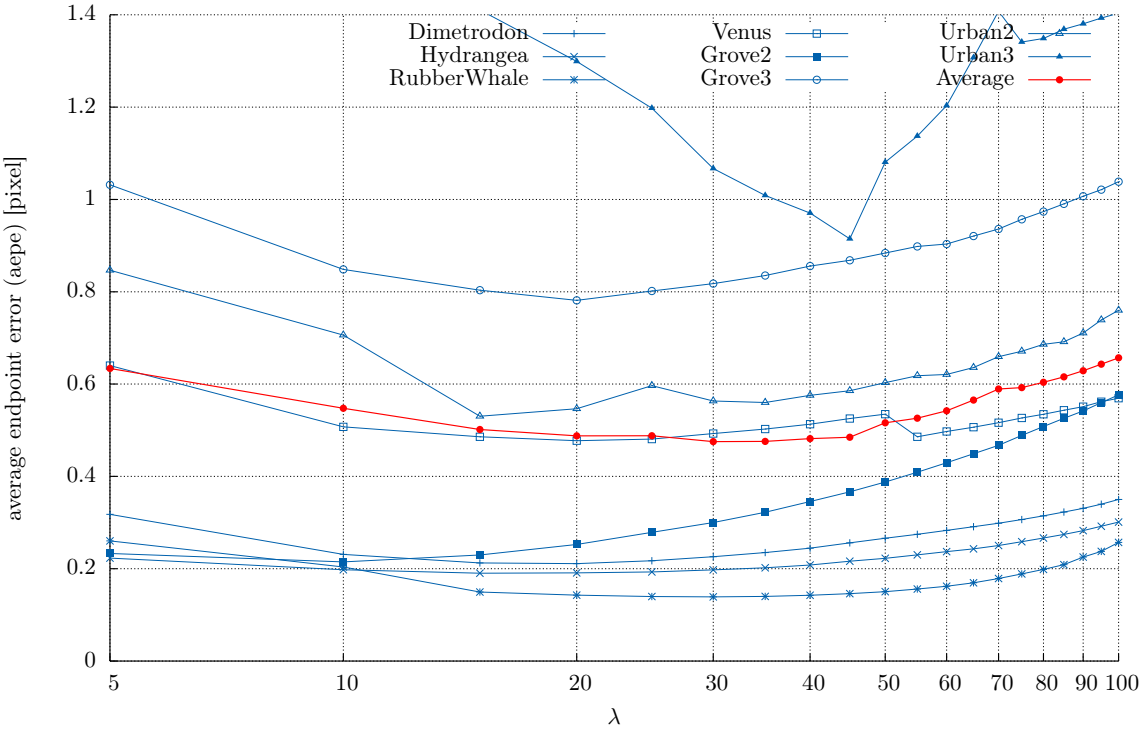


Figure A.37.: nonlocal Huber-Quadfit (GRAD)

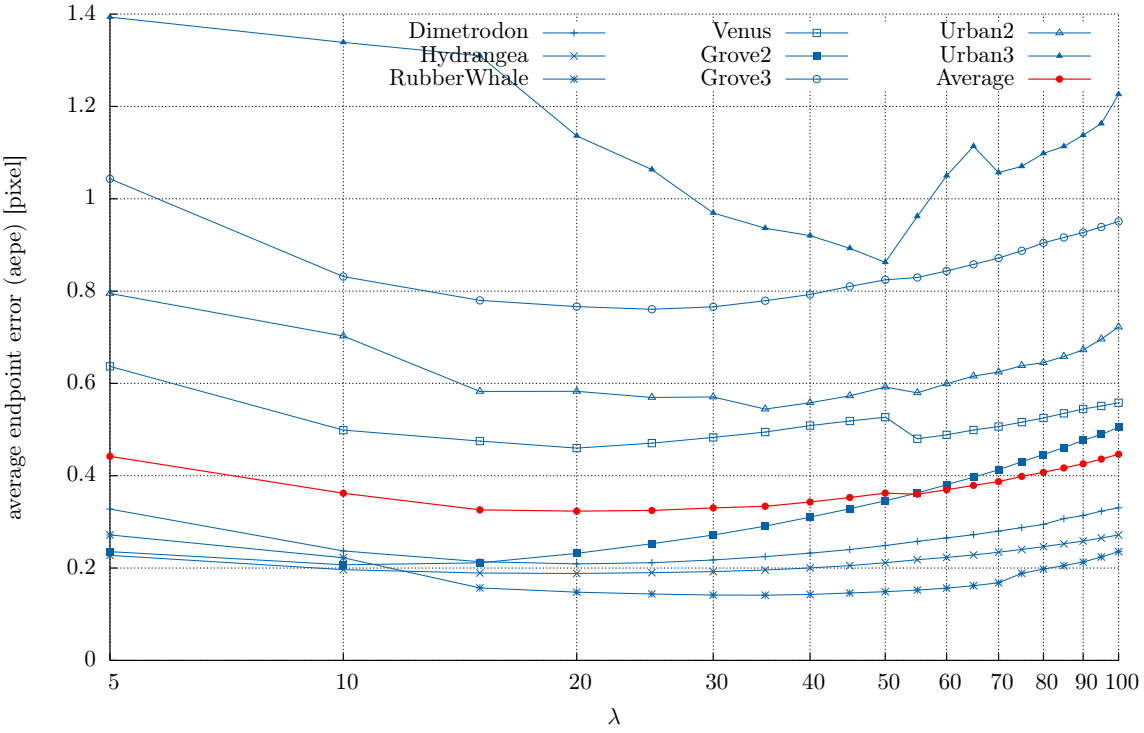


Figure A.38.: nonlocal Huber-Quadfit (GRAD+INTENSITY)

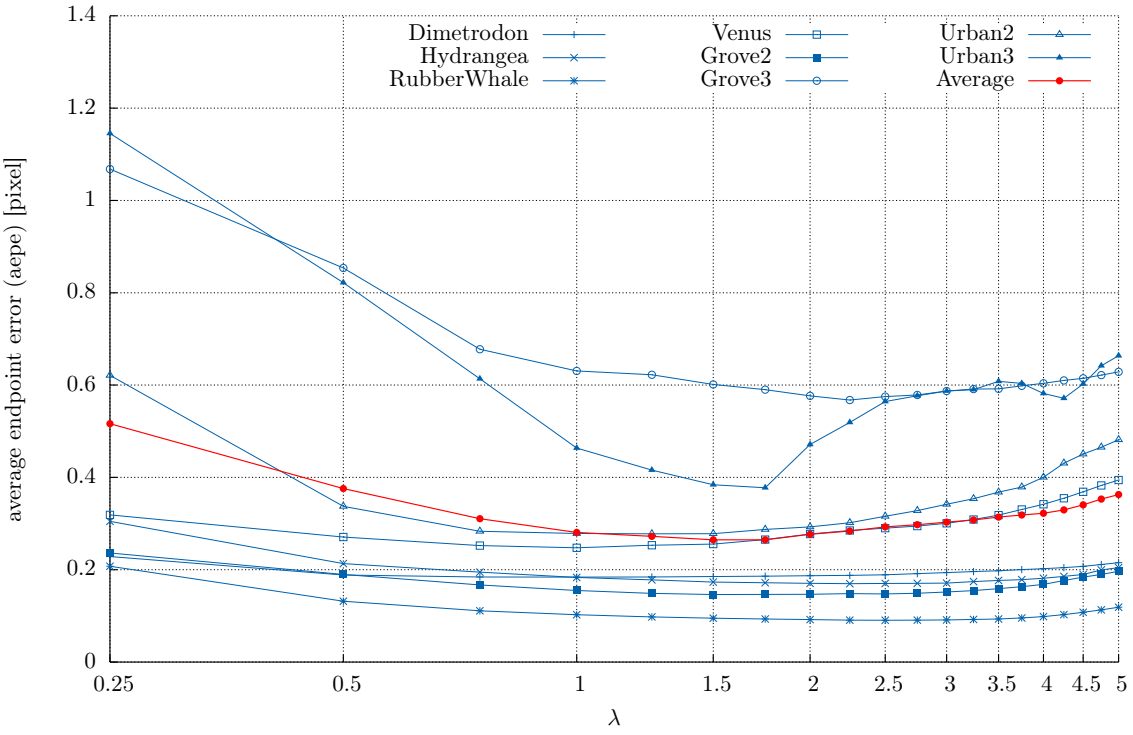


Figure A.39.: nonlocal Huber-Quadfit (NCC3)

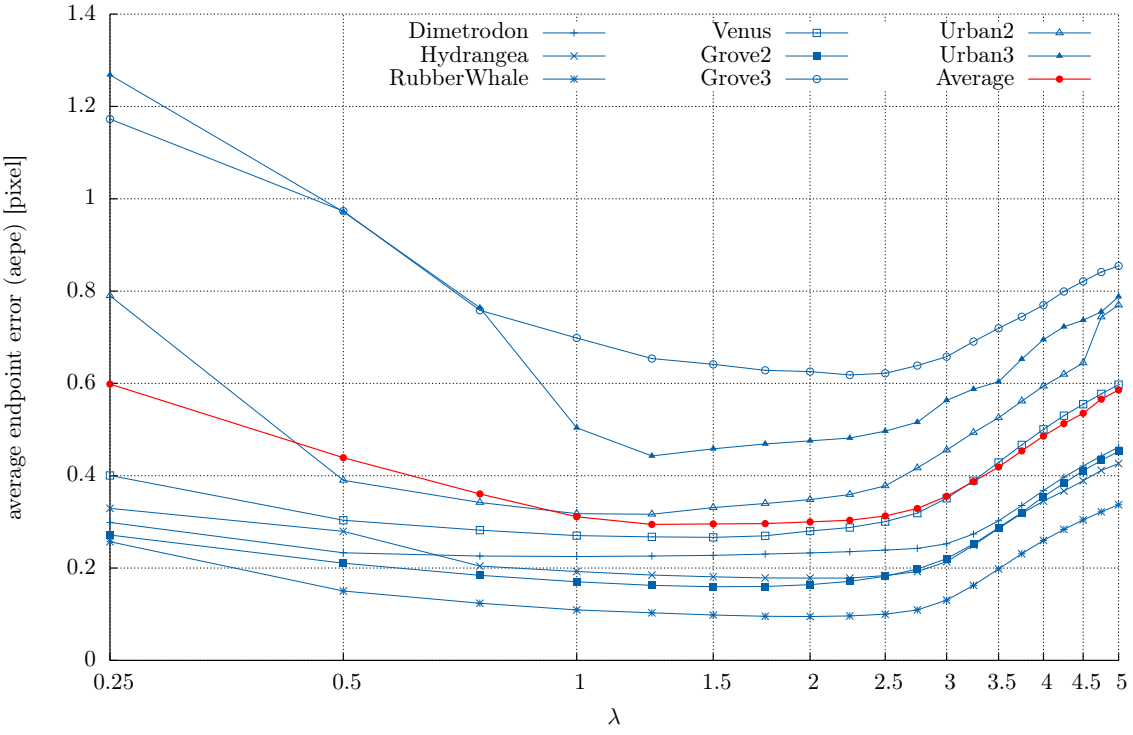


Figure A.40.: nonlocal Huber-Quadfit (normalized GRAD)

APPENDIX
B

NOTATION

List of Symbols

$\mathbb{R}$	Set of real numbers.
x	Spatial image position; in the 2D case $x = \left(x^1, x^2\right)^T$. When multiple spatial positions are needed we utilize the notation $x_1, x_2, x_3, \ldots$. Then the coordinate indices are denoted as $x_i = \left(x_i^1, x_i^2\right)^T$.
Ω	Image domain, typically a rectangular and regular Cartesian grid of size $M \times N$.
M, N	Image dimensions (grid size).
X	Two dimensional (2D), real-valued vector space of size $M \times N$.
Y	Vector valued 2D vector space with two components of size $2 \times M \times N$.
Z	Vector valued 2D vector space with four components of size $4 \times M \times N$.
C	Closed convex set.
D	BV gradient.
∇	Gradient operator.
div	Divergence operator.
$\partial_x a$	Partial derivative of a w.r.t. to x.
$\delta_x^- a$	Backward finite differences of a w.r.t. to x.
$\delta_x^+ a$	Forward finite differences of a w.r.t. to x.
v	Variable to optimize. For *e.g.* denoising it represents the denoised image, $v \in X$.
$\check{v}$	Desired solution. For *e.g.* denoising it represents the clean image, $\check{v} \in X$.
u	Optical flow vector $u = \left(u^1, u^2\right)^T,\ u \in Y$.
$\check{u}$	Ground truth optical flow vector, $\check{u} \in Y$.
c	"Compensation" variable.
q	Dual variable w.r.t. to v or c, $q \in Y$.
p	Dual variable w.r.t. to u, $p \in Z$.
$\mathcal{R}(\cdot)$	Regularization term.
$\mathcal{D}(\cdot)$	Data fidelity term.
ψ	Penalty function.
$\delta(\cdot)$	Similarity measure.
$\mathcal{C}^1$	Set of continuously differentiable functions.

$\mathcal{C}^\infty$	Set of functions that are differentiable for all degrees of differentiation.
$\operatorname{epi} F$	Epigraph of function F.
I_x	Shorthand for the spatial derivative $\partial_x I\left(x(t), t\right)$.
I_t	Shorthand for the temporal derivative $\partial_t I\left(x(t), t\right)$.
$\mathcal{N}\left(x\right)$, $\mathcal{N}_x$	Set of neighboring pixels around x.
$\operatorname{tr}\left(A\right)$	The trace of matrix A.
$\det\left(A\right)$	The determinant of matrix A.
$\operatorname{diag}\left(a\right)$	Diagonal matrix with elements a.
TODO	scalar product

BIBLIOGRAPHY

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. (cited on page 40)

Alvarez, L., Esclarín, J., Lefebure, M., and Sánchez, J. (1999). A PDE model for computing the Optical Flow. In *Proceedings XVI Congreso de Ecuaciones Diferenciales y Aplicaciones C.E.D.Y.A. XVI*, volume 1, pages 1349–1356. (cited on page 47)

Alvarez, L., Weickert, J., and Sánchez, J. (2000). Reliable Estimation of Dense Optical Flow Fields with Large Displacements. *International Journal of Computer Vision*, 39(1):41–56. (cited on page 56)

Anandan, P. (1989). A computational framework and an algorithm for the measurement of visual motion. *International Journal of Computer Vision*, 2(3):283–310. (cited on page 56)

Arias, P., Facciolo, G., Caselles, V., and Sapiro, G. (2011). A Variational Framework for Exemplar-Based Image Inpainting. *International Journal of Computer Vision*, 93(3):319–347. (cited on page 129)

Aujol, J.-F., Gilboa, G., Chan, T., and Osher, S. (2006). Structure-Texture Image Decomposition—Modeling, Algorithms, and Parameter Selection. *International Journal of Computer Vision*, 67(1):111–136. (cited on page 48)

Aurich, V. and Weule, J. (1995). Non-Linear Gaussian Filters Performing Edge Preserving Diffusion. In *DAGM*, volume 17, pages 538–545. Springer-Verlag. (cited on page 71)

Baker, S., Scharstein, D., Lewis, J. P., Roth, S., Black, M. J., and Szeliski, R. (2011). A Database and Evaluation Methodology for Optical Flow. *International Journal of Computer Vision*, 92(1):1–31. (cited on pages 44, 101, and 122)

Barash, D. (2002). Fundamental relationship between bilateral filtering, adaptive smoothing, and the nonlinear diffusion equation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(6):844–847. (cited on page 72)

Barash, D. (2004). A common framework for nonlinear diffusion, adaptive smoothing, bilateral filtering and mean shift. *Image and Vision Computing*, 22(1):73–81. (cited on page 72)

Barnes, C., Shechtman, E., Finkelstein, A., and Goldman, D. B. (2009). PatchMatch. *ACM Transactions on Graphics*, 28(3):1. (cited on pages 130 and 131)

Bertsekas, D. P. (1982). *Constrained Optimization and Lagrange Multiplier Methods*. (cited on page 35)

Bertsekas, D. P., Nedi, A., and Ozdaglar, A. E. (2003). *Convex Analysis and Optimization*, volume 3 of *Athena Scientific Optimization and Computation Series*. Athena Scientific. (cited on pages 28, 31, 32, and 33)

Biemond, J., Looijenga, L., Boekee, D., and Plompen, R. (1987). A pel-recursive Wiener-based displacement estimation algorithm. *Signal Processing*, 13(4):399–412. (cited on page 12)

Black, M. and Anandan, P. (1991). Robust dynamic motion estimation over time. In *Proceedings. 1991 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 296–302. IEEE Comput. Sco. Press. (cited on pages 44 and 46)

Black, M. and Anandan, P. (1993). A framework for the robust estimation of optical flow. In *International Conference on Computer Vision*, pages 231–236. IEEE Computer Society Press. (cited on pages 46 and 48)

Black, M. and Jepson, A. (1996). Estimating optical flow in segmented images using variable-order parametric models with local deformations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(10):972–986. (cited on page 44)

Black, M. J. (1992). *Robust Incremental Optical Flow*. phdthesis, Yale University. (cited on pages 44 and 46)

Black, M. J. and Anandan, P. (1996). The Robust Estimation of Multiple Motions: Parametric and Piecewise-Smooth Flow Fields. *Computer Vision and Image Understanding*, 63(1):75–104. (cited on pages 46 and 56)

Bougleux, S., Elmoataz, A., and Melkemi, M. (2007). Discrete Regularization on Weighted Graphs for Image and Mesh Filtering. In Sgallari, F., Murli, A., and Paragios, N., editors, *Scale Space and Variational Methods in Computer Vision*, volume 4485 of *Lecture Notes in Computer Science*, pages 128–139, Berlin, Heidelberg. Springer Berlin Heidelberg. (cited on page 72)

Boyd, S. and Vandenberghe, L. (2004). *Convex Optimization*. (cited on pages 28, 31, 32, and 33)

Bresson, X., Esedoḡlu, S., Vandergheynst, P., Thiran, J.-P., and Osher, S. (2005). Global minimizers of the active contour/snake model. In *Free Boundary Problems (FBP): Theory and Applications*. (cited on page 65)

Bresson, X., Esedoḡlu, 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. (cited on page 65)

Brox, T., Bruhn, A., Pappenberg, N., and Weickert, J. (2004). High Accuracy Optical Flow Estimation Based on a Theory for Warping. In *European Conference on Computer Vision*, pages 1–12. (cited on pages 4, 46, 49, 56, 72, and 83)

Brox, T. and Malik, J. (2010). Object segmentation by long term analysis of point trajectories. In *ECCV*, pages 282–295. (cited on page 56)

Buades, A., Coll, B., and Morel, J.-M. (2005a). A Non-Local Algorithm for Image Denoising. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, pages 60–65. IEEE. (cited on page 72)

Buades, A., Coll, B., and Morel, J. M. (2005b). A Review of Image Denoising Algorithms, with a New One. *Multiscale Modeling & Simulation*, 4(2):490–530. (cited on page 72)

Buades, A., Coll, B., and Morel, J.-M. (2006). The staircasing effect in neighborhood filters and its solution. *IEEE Transactions on Image Processing*, 15(6):1499–1505. (cited on page 72)

Burt, P. and Adelson, E. (1983). The Laplacian Pyramid as a Compact Image Code. *IEEE Transactions on Communications*, 31(4):532–540. (cited on page 56)

Burt, P. J. (1981). Fast filter transform for image processing1. *Computer Graphics and Image Processing*, 16(1):20–51. (cited on page 56)

Chambolle, A. (2004). An Algorithm for Total Variation Minimization and Applications. *Journal of Mathematical Imaging and Vision*, 20(1):89–97. (cited on pages 36, 37, and 46)

Chambolle, A., Caselles, V., Novaga, M., Cremers, D., and Pock, T. (2010). *An Introduction to Total Variation for Image Analysis*. Radon Series on Computational and Applied Mathematics. Walter de Gruyter. (cited on page 27)

Chambolle, A. and Pock, T. (2010). A First-Order Primal-Dual Algorithm for Convex Problems with Applications to Imaging. *Journal of Mathematical Imaging and Vision*, 40(1):120–145. (cited on pages 36, 52, 79, and 80)

Chan, T. F. and Esedoglu, S. (2005). Aspects of Total Variation Regularized L1 Function Approximation. *SIAM Journal on Applied Mathematics*, 65(5):1817. (cited on page 115)

Chan, T. F., Golub, G. H., and Mulet, P. (1996). A Nonlinear Primal-Dual Method For Total Variation-Based Image Restoration. In Berger, M.-O., Deriche, R., Herlin, I., Jaffré, J., and Morel, J.-M., editors, *ICAOS*, volume 219 of *Lecture Notes in Control and Information Sciences*, Berlin, Heidelberg. Springer Berlin Heidelberg. (cited on page 36)

Cohen, I. (1993). Nonlinear Variational Method for Optical Flow Computation. In *Proceedings of the 8th SCIA*, number 1, pages 523–530. (cited on page 46)

Criminisi, A., Perez, P., and Toyama, K. (2004). Region Filling and Object Removal by Exemplar-Based Image Inpainting. *IEEE Transactions on Image Processing*, 13(9):1200–1212. (cited on pages 130 and 131)

Crowley, J. L. (1981). *A Representation for Visual Information*. Cmu-ri-tr-82-7, Robotics Institute, Carnegie-Mellon University. (cited on page 56)

Dennis, T. (1980). Nonlinear temporal filter for television picture noise reduction. In *Electronic Circuits and Systems, IEE Proceedings G*, volume 127, pages 52–56. (cited on pages 12 and 16)

Dubois, E. and Sabri, S. (1984). Noise Reduction in Image Sequences Using Motion-Compensated Temporal Filtering. *IEEE Transactions on Communications*, 32(7):826–831. (cited on pages 12 and 16)

Eckstein, J. and Bertsekas, D. P. (1992). On the Douglas—Rachford splitting method and the proximal point algorithm for maximal monotone operators. *Mathematical Programming*, 55(1-3):293–318. (cited on page 35)

Efstratiadis, S. N. and Katsaggelos, A. K. (1990). A model-based pel-recursive motion estimation algorithm. In *International Conference on Acoustics, Speech, and Signal Processing*, pages 1973–1976. IEEE. (cited on pages 12 and 16)

Elmoataz, A., Lezoray, O., and Bougleux, S. (2008). Nonlocal discrete regularization on weighted graphs: a framework for image and manifold processing. *IEEE transactions on image processing : a publication of the IEEE Signal Processing Society*, 17(7):1047–60. (cited on page 72)

Erdem, A. T., Sezan, M. I., and Ozkan, M. K. (1992). Motion-compensated multiframe Wiener restoration of blurred and noisy image sequences. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 293–296. IEEE. (cited on pages 12 and 16)

Esser, E., Zhang, X., and Chan, T. F. (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. (cited on page 36)

Fleet, D. J. and Jepson, A. D. (1990). Computation of component image velocity from local phase information. *International Journal of Computer Vision*, 5(1):77–104. (cited on page 101)

Gennert, M. A. and Negahdaripour, S. (1987). Relaxing the Brightness Constancy Assumption in Computing Optical Flow. In *MIT AI Memo*. MIT A.I. Memo No. 975. (cited on page 49)

Gilboa, G. and Osher, S. (2007). Nonlocal Operators with Applications to Image Processing. Technical report, UCLA CAM Report 07-23. (cited on page 72)

Hadamard, J. (1902). Sur les problèmes aux dérivées partielles et leur signification physique. *Princeton University Bulletin*, 13:49–52. (cited on page 22)

Haussecker, H. and Fleet, D. (2001). Computing optical flow with physical models of brightness variation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(6):661–673. (cited on page 50)

Heckbert, P. (1989). Fundamentals of Texture Mapping and Image Warping. Master's thesis, U.C. Berkeley, Berkeley, California, United States. (cited on page 56)

Höller, F. (2012). Ablauf einer typischen digitalen Filmrestauration. *FKT: Die Fachzeitschrift für Fernsehen, Film und elektronische Medien*, pages 185—-187. (cited on page 1)

Horn, B. K. (1986). *Robot Vision*. (cited on pages 40 and 49)

Horn, B. K. and Schunck, B. G. (1981). Determining optical flow. *Artificial Intelligence*, 17(1-3):185–203. (cited on pages 41, 44, 45, 46, 48, and 58)

Huang, J. and Mumford, D. (1999). Statistics of natural images and models. In *Proceedings. 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Cat. No PR00149)*, volume 1, pages 541–547. IEEE Comput. Soc. (cited on page 24)

Huang, T., editor (1981). *Image sequence analysis*. Springer series in information sciences. Springer-Verlag. (cited on page 16)

Huber, P. J. (1973). Robust Regression: Asymptotics, Conjectures and Monte Carlo. *The Annals of Statistics*, 1(5):799–821. (cited on page 60)

Katsaggelos, A., Driessen, J., Efstratiadis, S., and Lagendijk, R. (1989). Spatio-temporal motion compensated noise filtering of image sequences. In *Proceedings of SPIE - The International Society for Optical Engineering*, volume 1199 pt 1, pages 61–70. (cited on page 16)

Keller, J. B. (1976). Inverse Problems. *The American Mathematical Monthly*, 83(2):pp. 107–118. (cited on page 22)

Kim, Y.-H., Martínez, A. M., and Kak, A. C. (2005). Robust motion estimation under varying illumination. *Image and Vision Computing*, 23(4):365–375. (cited on page 49)

Kindermann, S., Osher, S., and Jones, P. W. (2004). Deblurring and Denoising of Images by Nonlocal. Technical report, UCLA CAM Report 04-75. (cited on page 72)

Kokaram, A. C. (2003). *Motion Picture Restoration.* phdthesis, University of Cambridge. (cited on pages 1, 2, 12, 13, and 16)

Lee, J.-S. (1983). Digital image smoothing and the sigma filter. *Computer Vision, Graphics, and Image Processing*, 24(2):255–269. (cited on page 71)

Leung, S. and Osher, S. (2005). Variational, Geometric, and Level Set Methods in Computer Vision. In Paragios, N., Faugeras, O., Chan, T., and Schnörr, C., editors, *VARIATIONAL, GEOMETRIC, AND LEVEL SET METHODS IN COMPUTER VISION*, volume 3752 of *Lecture Notes in Computer Science*, pages 149–160, Berlin, Heidelberg. Springer Berlin Heidelberg. (cited on page 65)

Lindeberg, T. (1994). *Scale-Space Theory in Computer Vision.* Kluwer Academic Publisher. (cited on page 56)

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. (cited on page 35)

Longuet-Higgins, H. C. and Prazdny, K. (1980). The Interpretation of a Moving Retinal Image. *Proceedings of the Royal Society B: Biological Sciences*, 208(1173):385–397. (cited on page 40)

Lucas, B. D. and Kanade, T. (1981). An Iterative Image Registration Technique with an Application to Stereo Vision. (cited on pages 41, 42, and 44)

Mansouri, A.-R., Mitiche, A., and Aron, M. (2003). PDE-based region tracking without motion computation by joint space-time segmentation. In *International Conference on Image Processing*, pages 113—-116. (cited on page 115)

Martinez, D. M. (1987). *Model-based motion estimation and its application to restoration and interpolation of motion pictures.* phdthesis, Massachusetts Institute of Technology. (cited on page 16)

Martinez, D. M. and Lim, J. S. (1985). Implicit motion compensated noise reduction of motion video scenes. In *ICASSP '85. IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 10, pages 375–378. Institute of Electrical and Electronics Engineers. (cited on pages 12 and 16)

Martinez, D. M. and Lim, J. S. (1989). Spatial interpolation of interlaced television pictures. In *International Conference on Acoustics, Speech, and Signal Processing*, pages 1886–1889. IEEE. (cited on page 12)

McMann, R. H., Kreinik, S., Moore, J. K., Kaiser, A., and Rossi, J. (1978). A Digital Noise Reducer for Encoded NTSC Signals. *SMPTE Journal*, 87(3):129–133. (cited on page 16)

Memin, E. and Perez, P. (1996). Robust discontinuity-preserving model for estimating optical flow. In *Proceedings of 13th International Conference on Pattern Recognition*, pages 920–924 vol.1. IEEE. (cited on page 46)

Mémin, E. and Pérez, P. (2002). Hierarchical estimation and segmentation of dense motion fields. *International Journal of Computer Vision*, 46(2):129–155. (cited on page 56)

Mileva, Y., Bruhn, A., and Weickert, J. (2007). Illumination-Robust Variational Optical Flow with Photometric Invariants. In *DAGM*, pages 152–162. (cited on page 50)

Moravec, H. P. (1979). Visual mapping by a robot rover. In *Proceedings of the 6th international joint conference on Artificial intelligence - Volume 1*, IJCAI'79, pages 598–600, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc. (cited on page 44)

Nagel, H.-H. (1983). Constraints for the Estimation of Displacement Vector Fields From Image Sequences. In *Proceedings of the Eighth international joint conference on Artificial intelligence*, volume 2, pages 945–951. (cited on page 47)

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*, 11(1):13–30. (cited on page 49)

Nagel, H.-H. and Enkelmann, W. (1986). An investigation of smoothness constraints for the estimation of displacement vector fields from image sequences. *IEEE Trans. on Pattern Analysis and Machine Intelligence*. (cited on page 56)

National Film Preservation Foundation (2004). *The Film Preservation Guide: The Basics for Archives, Libraries, and Museums*. National Film Preservation Foundation. (cited on page 8)

Negahdaripour, S. (1998). Revised definition of optical flow: integration of radiometric and geometric cues for dynamic scene analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(9):961–979. (cited on page 49)

Nemirovski, A. (2004). Prox-Method with Rate of Convergence O(1/t) for Variational Inequalities with Lipschitz Continuous Monotone Operators and Smooth Convex-Concave Saddle Point Problems. *SIAM Journal on Optimization*, 15(1):229. (cited on page 36)

Nesterov, Y. (2004a). *Introductory lectures on convex optimization: A basic course*, volume 87. Springer. (cited on page 33)

Nesterov, Y. (2004b). Smooth minimization of non-smooth functions. *Mathematical Programming*, 103(1):127–152. (cited on page 36)

Nikolova, M. (2004). A Variational Approach to Remove Outliers and Impulse Noise. *Journal of Mathematical Imaging and Vision*, (3):99–120. (cited on page 115)

Otte, M. and Nagel, H.-H. (1995). Estimation of optical flow based on higher-order spatiotemporal derivatives in interlaced and non-interlaced image sequences. *Artificial Intelligence*, 78(1-2):5—-43. (cited on page 101)

Ozkan, M. K. (1991). LMMSE restoration of blurred and noisy image sequences. In *Proceedings of SPIE*, volume 1606, pages 743–754. SPIE. (cited on page 16)

Ozkan, M. K. (1992). Motion-adaptive weighted averaging for temporal filtering of noisy image sequences. In *Proceedings of SPIE*, volume 1657, pages 201–212. SPIE. (cited on page 16)

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*, 67(2):141–158. (cited on pages 46, 49, and 56)

Paris, S., Kornprobst, P., Tumblin, J., and Durand, F. (2009). Bilateral Filtering: Theory and Applications. *Foundations and Trends in Computer Graphics and Vision*, 4(1):1–75. (cited on page 72)

Perona, P. and Malik, J. (1990). Scale-space and edge detection using anisotropic diffusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(7):629–639. (cited on page 72)

Pock, T. (2008). *Fast Total Variation for Computer Vision*. phdthesis, Graz University of Technology. (cited on page 115)

Pock, T. and Chambolle, A. (2011). Diagonal preconditioning for first order primal-dual algorithms in convex optimization. In *International Conference on Computer Vision (ICCV 2011)*. (cited on pages 38, 83, and 88)

Pock, T., Cremers, D., Bischof, H., and Chambolle, A. (2009). An algorithm for minimizing the Mumford-Shah functional. In *2009 IEEE 12th International Conference on Computer Vision*, pages 1133–1140. IEEE. (cited on page 36)

Rockafellar, R. T. (1976). Monotone Operators and the Proximal Point Algorithm. *SIAM Journal on Control and Optimization*, 14(5):877. (cited on page 35)

Rockafellar, R. T. (1997). *Convex Analysis*. Princeton Landmarks in Math. Princeton University Press. (cited on pages 28, 31, 32, and 33)

Rockafellar, R. T. and Wets, R. J.-B. (1997). *Variational Analysis*. Grundlehren der Mathematischen Wissenschaften. Springer. (cited on pages 28, 31, 32, and 33)

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. (cited on pages 25, 46, 79, and 114)

Sand, P. and Teller, S. (2008). Particle Video: Long-Range Motion Estimation Using Point Trajectories. *International Journal of Computer Vision*, 80(1):72–91. (cited on page 72)

Schnörr, C. (1994). Segmentation of visual motion by minimizing convex non-quadratic functionals. In *Proceedings of 12th International Conference on Pattern Recognition*, volume 1, pages 661–663. IEEE Comput. Soc. Press. (cited on page 47)

Shulman, D. and Hervé, J.-Y. (1989). Regularization of discontinuous flow fields. In *[1989] Proceedings. Workshop on Visual Motion*, pages 81–86. IEEE Comput. Soc. Press. (cited on pages 27, 44, 46, 49, and 60)

Slesareva, N., Bruhn, A., and Weickert, J. (2005). Optic Flow Goes Stereo: a Variational Method for Estimating Discontinuity-Preserving Dense Disparity Maps. *DAGM*, pages 33–40. (cited on page 48)

Slide, A. (1992). *Nitrate won't wait: a history of film preservation in the United States*. McFarland & Co. (cited on page 8)

Smith, S. M. and Brady, J. M. (1997). SUSAN – A New Approach to Low Level Image Processing. *International Journal of Computer Vision*, 23(1):45–78. (cited on page 72)

Steinbrücker, F., Pock, T., and Cremers, D. (2009a). Large displacement optical flow computation without warping. In *2009 IEEE 12th International Conference on Computer Vision*, pages 1609–1614. IEEE. (cited on page 50)

Steinbrücker, F., Pock, T., Cremers, D., and Steinbruecker, F. (2009b). Advanced Data Terms for Variational Optic Flow Estimation. In *Proceedings Vision, Modeling and Visualization (VMV)*, volume 1. (cited on page 50)

Tikhonov, A. N. (1963). Solution of incorrectly formulated problems and the regularization method. *Soviet Math. Dokl.*, 4:1035–1038. (cited on pages 24 and 45)

Tistarelli, M. (1996). Multiple constraints to compute optical flow. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(12):1243–1250. (cited on page 49)

Tomasi, C. and Manduchi, R. (1998). Bilateral filtering for gray and color images. In *International Conference on Computer Vision*, pages 839–846. Narosa Publishing House. (cited on page 72)

Tretiak, O. J. and Pastor, L. (1984). Velocity Estimation from Image Sequences with Second Order Differential Operators. In *ICPR*, pages 16–19. (cited on page 49)

Trobin, W. (2009). *Local, Semi-global, and Global Optimization for Motion Estimation*. phdthesis, Graz University of Technology, Institute for Computer Graphics and Vision. (cited on page 45)

Tukey, J. W. (1977). *Exploratory Data Analysis*, volume 2 of *Quantitative applications in the social sciences*. Addison-Wesley. (cited on page 16)

Unger, M., Mauthner, T., Pock, T., and Bischof, H. (2009). Tracking as Segmentation of Spatial-Temporal Volumes by Anisotropic Weighted TV. In *Proceedings of the 7th International Conference on Energy Minimization Methods in Computer Vision and Pattern Recognition*. Springer-Verlag. (cited on page 115)

Uras, S., Girosi, F., Verri, A., and Torre, V. (1988). A computational approach to motion perception. *Biological Cybernetics*, 60(2):79–87. (cited on page 49)

Valgaerts, L., Bruhn, A., and Weickert, J. (2008). A Variational Model for the Joint Recovery of the Fundamental Matrix and the Optical Flow. In *DAGM*, pages 314–324. (cited on page 48)

van de Weijer, J. and Gevers, T. (2004). Robust optical flow from photometric invariants. In *2004 International Conference on Image Processing, 2004. ICIP '04.*, volume 3, pages 1835–1838. IEEE. (cited on page 50)

Van Roosmalen, P. (1999). *Restoration of archived film and video*. phdthesis, Delft University of Technology. (cited on pages 1, 11, 12, and 13)

Verri, A. and Poggio, T. (1989). Motion field and optical flow: qualitative properties. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(5):490–498. (cited on pages 40 and 49)

Wang, J. A. and Adelson, E. H. (1994). Representing moving images with layers. *IEEE transactions on image processing : a publication of the IEEE Signal Processing Society*, 3(5):625–38. (cited on page 44)

Wedel, A., Pock, T., Braun, J., Franke, U., and Cremers, D. (2008). Duality TV-L1 flow with fundamental matrix prior. In *2008 23rd International Conference Image and Vision Computing New Zealand*, pages 1–6. IEEE. (cited on pages 48 and 86)

Wedel, A., Pock, T., Zach, C., Bischof, H., and Cremers, D. (2009). An Improved Algorithm for TV-L 1 Optical Flow. In *Statistical and Geometrical Approaches to Visual Motion Analysis, International Dagstuhl Seminar*, volume 1, pages 23–45. (cited on page 48)

Weickert, J., Bruhn, A., Brox, T., and Papenberg, N. (2006). A Survey on Variational Optic Flow Methods for Small Displacements. *Mathematical models for registration and applications to medical imaging*, pages 103–136. (cited on page 45)

Weickert, J. and Schnörr, C. (2001). A Theoretical Framework for Convex Regularizers in PDE-Based Computation of Image Motion. (cited on pages 4, 45, and 47)

Weiss, Y. and Freeman, W. T. (2007). What makes a good model of natural images? In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE. (cited on page 24)

Werlberger, M., Pock, T., and Bischof, H. (2010). Motion Estimation with Non-Local Total Variation Regularization. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. (cited on pages 3, 4, and 91)

Werlberger, M., Pock, T., Unger, M., and Bischof, H. (2011a). Optical Flow Guided TV-L1 Video Interpolation and Restoration. In *Proceedings Energy Minimization Methods in Computer Vision and Pattern Recognition*, Lecture Notes in Computer Science, pages 273–286. Springer. (cited on pages 3 and 5)

Werlberger, M., Trobin, W., Pock, T., Wedel, A., Cremers, D., and Bischof, H. (2009). Anisotropic Huber-L1 Optical Flow. In *Proceedings of the British Machine Vision Conference (BMVC)*, London, UK. (cited on pages 3 and 4)

Werlberger, M., Unger, M., Pock, T., and Bischof, H. (2011b). Efficient Minimization of the Non-Local Potts Model. In *Proceedings International Conference on Scale Space and Variational Methods in Computer Vision*, Lecture Notes in Computer Science. Springer. (cited on page 3)

Xiao, J. J., Cheng, H., Sawhney, H. S., Rao, C., and Isnardi, M. (2006). Bilateral Filtering-Based Optical Flow Estimation with Occlusion Detection. In *ECCV*. (cited on page 72)

Xu, L., Jia, J., and Matsushita, Y. (2010). Motion detail preserving optical flow estimation. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1293–1300. IEEE. (cited on page 56)

Xu, L., Jia, J., and Matsushita, Y. (2011). Motion Detail Preserving Optical Flow Estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–14. (cited on page 56)

Yang, Q., Wang, L., Yang, R., Stewénius, H., and Nistér, D. (2009). Stereo matching with color-weighted correlation, hierarchical belief propagation, and occlusion handling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(3):492–504. (cited on page 72)

Yaroslavsky, L. P. (1985). *Digital Picture Processing — an Introduction.* (cited on page 71)

Yoon, K.-J. and Kweon, I. S. (2006). Adaptive support-weight approach for correspondence search. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(4):650–6. (cited on pages 72 and 73)

Zach, C., Pock, T., and Bischof, H. (2007). A duality based approach for realtime TV-L1 optical flow. In *DAGM*, pages 214–223. (cited on pages 4, 46, and 52)

Zhou, D. and Schölkopf, B. (2004). A regularization framework for learning from graph data. In *ICML Workshop on Statistical Relational Learning*, pages 132—-137. (cited on page 72)

Zhou, D. and Schölkopf, B. (2005). Regularization on Discrete Spaces. In *DAGM*, volume 80, pages 361–368. Springer. (cited on page 72)

Zimmer, H., Bruhn, A., Weickert, J., Valgaerts, L., Salgado, A., Rosenhahn, B., and Seidel, H.-P. (2009). Complementary Optic Flow. In *Energy Minimization Methods in Computer Vision and Pattern Recognition (EMMCVPR)*, pages 207–220. (cited on page 49)