Advanced Restoration Techniques for Images and Disparity Maps

Inauguraldissertation
der Philosophisch-naturwissenschaftlichen Fakultät
der Universität Bern

vorgelegt von

Siavash Arjomand Bigdeli

von IRAN

Leiter der Arbeit:

Prof. Dr. M. Zwicker
Universität Bern

Prof. Dr. S. Süsstrunk
École polytechnique fédérale de Lausanne

Abstract

With increasing popularity of digital cameras, the field of Computational Photography emerges as one of the most demanding areas of research. In this thesis we study and develop novel priors and optimization techniques to solve inverse problems, including disparity estimation and image restoration.

The disparity map estimation method proposed in this thesis incorporates multiple frames of a stereo video sequence to ensure temporal coherency. To enforce smoothness, we use spatio-temporal connections between the pixels of the disparity map to constrain our solution. Apart from smoothness, we enforce a consistency constraint for the disparity assignments by using connections between the left and right views. These constraints are then formulated in a graphical model, which we solve using mean-field approximation. We use a filter-based mean-field optimization that perform efficiently by updating the disparity variables in parallel. The parallel updates scheme, however, is not guaranteed to converge to a stationary point. To compare and demonstrate the effectiveness of our approach, we developed a new optimization technique that uses sequential updates, which runs efficiently and guarantees convergence. Our empirical results indicate that with proper initialization, we can employ the parallel update scheme and efficiently optimize our disparity maps without loss of quality. Our method ranks amongst the state of the art in common benchmarks, and significantly reduces the temporal flickering artifacts in the disparity maps.

In the second part of this thesis, we address several image restoration problems such as image deblurring, demosaicing and super-resolution. We propose to use denoising autoencoders to learn an approximation of the true natural image distribution. We parametrize our denoisers using deep neural networks and show that they learn the gradient of the smoothed density of natural images. Based on this analysis, we propose a restoration technique that moves the solution towards the local extrema of this distribution by minimizing the difference between the input and output of our denoiser. We

demonstrate the effectiveness of our approach using a single trained neural network in several restoration tasks such as deblurring and super-resolution. In a more general framework, we define a new Bayes formulation for the restoration problem, which leads to a more efficient and robust estimator. The proposed framework achieves state of the art performance in various restoration tasks such as deblurring and demosaicing, and also for more challenging tasks such as noise- and kernel-blind image deblurring.

Keywords. disparity map estimation, stereo matching, mean-field optimization, graphical models, image processing, linear inverse problems, image restoration, image deblurring, image denoising, single image super-resolution, image demosaicing, deep neural networks, denoising autoencoders

Acknowledgements

I thank Matthias Zwicker, my mentor and adviser, for generously offering the opportunity to learn and develop under his supervision. His honest views and broad knowledge of computer science shaped my views on research. I dedicate this work to him.

Many thanks are due to Sabine Süsstrunk and Paolo Favaro for their assistance in reviewing the thesis. I am also very grateful to have been Paolo's student and to collaborate with him during my research.

I would like to express sincere gratitude to Dragana Esser, who selflessly cleared the path for my work during my PhD. Her contributions to my studies and research are more than one could ever imagine or hope for. I also thank members of the Computer Graphics Group for their friendship and support throughout this time. On many occasions, they helped clarify the most challenging topics of my studies. Peter Bertholet, Marco Manzi, and Tiziano Portonier have helped me to arrive at a deeper understanding of Swiss-German culture and literature, and Daljit Singh Dhillon and Shihao Wu have broadened my knowledge of India and China. Last but not least, I would like to thank the Iranian students of the Computer Science Institute - Mehdi Nowrouzi, Ali Marandi, and Mostafa Karimzadeh - for keeping the connection to my home and culture strong.

The infinite support of my parents and my wife are the sole reason that I could finish my studies. My wife Rava has selflessly advanced my work on countless occasions, engaging in endless theoretical discussions about my research, listening to my practice talks, and proofreading my texts into the late hours of the night. I am immeasurably grateful to my family and devote my research to them.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

> Whether [the artist] regards himself as highly individual or not, it seems to me that he or she needs to be exquisitely aware of all the other images, symbols, movements floating around, so that they can be taken into account—so that the artist can anticipate the response of the viewer...
>
> Alvin Toffler

The Digital Revolution in the 20th century has ignited a wave of change in arts. As an influential futurist, Alvin Toffler suggested pop artists to begin interpreting their environment for clues and evidence about the preference of their audience, instead of escaping their context [TM87]. John McHale, as an art theorist, took this debate even further by formulating the environment as a set of *shared codes* and *symbols* to be used effectively as the artist's guide. This analogy is visible in his early work *Forms of Coding*, shown in Figure 1.1. He clearly depicts the role of *shared codes* in the form of a storage that help in decoding an observation to its symbol or a new work of art.

The role of an artist in producing appealing images is highly reminiscent of the tools being used today for digital imaging. Surprisingly, computer scientists usually take the same approach of pop artists

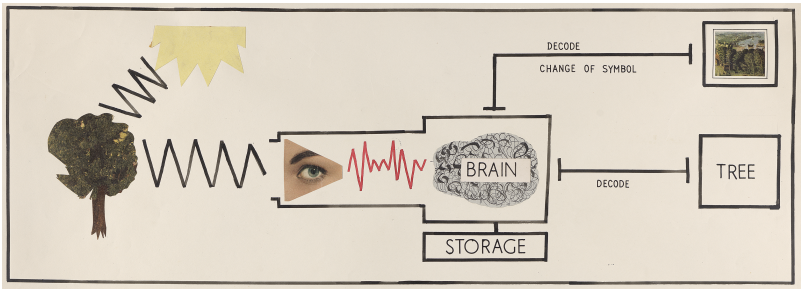


Figure 1.1: *Forms of Coding*, John McHale, 1955, © Yale Center for British Art.

by interpreting and analyzing digital images, and encoding intuitive representations of their distribution. Once obtained, these representations are made accessible to the computational tools that incorporate the knowledge to produce high quality results.

Aside from the aforementioned similarity, considering the application of computers and scientific tools, digital images are also becoming available to a broader range of users, and not limited to the use by artists. With recent technological advances, digital cameras are available almost everywhere today, which leads to further increase in demands for digital imaging tools. The Economist reports that, in the year 2010, more than one billion cameras where installed on mobile phones alone [Eco10]. In addition to mobile phone users, professional photographers, medical scientists, security experts, and many more use digital cameras every day for their specific applications. These cameras are being used continuously to capture images, some of which are observed by the human eye, others are further analyzed by computers. Thus it is necessary for the computer graphics and vision fields to make advancements as well and develop many tools to be used in applications ranging from visual quality enhancement to machine scene understanding. The two fields of computer graphics and vision intersect in the computational photography subdomain (also known as image processing), which covers most of the

techniques that are applied to digital images and to meta-data related to them. These techniques are being applied from the moment the image is being captured until it is analyzed or displayed, using both hardware and software tools.

The most common challenge in the field of computational photography is the presence of **under-constrained** or **ill-posed** problems. This is due to practical and physical limitations, that in most problems, fewer observations are being made than there are unknowns. As a trivial example, consider the case where one would measure the length of a single edge in a rectangle and try to compute its area. However in computing the area of a rectangle, two measurements (edge lengths) are required. In digital image processing, the unknowns are typically the RGB color values of pixels. A physical limitation of digital cameras is that the light intensity sensors only record the amount of incoming light for a specific color range. Therefore in practice, most digital images are captured using the Bayer's pattern (using red, green, and blue color filters). In this case, each pixel on the camera image uses a sensor to measures the intensity of a single color channel (e.g. red); thus, the other two color intensity values (e.g. green and blue) remain unknown for that pixel. This leads to a challenging problem known as *image demosaicing*, where it is necessary to infer the unknown color intensities from the captured values. Other examples of under-constrained problems include removing undesired artifacts from images (e.g. image denoising), image editing and enhancement (e.g. Photoshop). More general under-constrained tasks are designed to recover additional information (such as object segmentation masks, disparity maps, etc.) from digital images, and are often used by other computer vision tools (e.g. for pedestrian detection in self-driving cars).

An intuitive way to solve an under-constrained problem and infer its unknowns is to use **prior knowledge (a "prior" for short)** about the problem and the underlying data distribution. In essence, the prior encodes information about the problem (definition and parameters), and also about the plausibility and probability of data samples. This leads to a declarative approach, where the reasoning (algorithm) is

separated from the basic knowledge (prior). We take the same declarative approach in this thesis to address several under-constrained problems related to computational photography. We encode knowledge about the problems in usable formats (priors) and introduce optimization techniques that benefit from these priors for efficient reasoning. We apply the above approach to *disparity map estimation* and a more general class of *image restoration* problems, which we describe briefly below.

Disparity map estimation. In digital image processing, the term *disparity* refers to the change in pixel location between two views of the same scene. Due to the parallax between the two views, the disparity value at each pixel encodes information about its distance to the camera(s). The disparity map estimation (also known as stereo matching) task is one of the classical problems in computer vision, which requires finding and matching correspondences between two views (stereo pairs), that were captured from the same scene at the same time. Finding the correspondence between two views can help to understand the (relative) depth of the objects in the scene, which is similar to the human visual system. Disparity maps are used in challenging computer vision tasks such as detecting obstacles in autonomous vehicles. In computer graphics, disparity maps are used in many different applications such as novel view synthesis, object removal and insertion, as well as depth-aware refocusing.

Disparity map estimation is an ill-posed problem; matching the corresponding pixels or patches between the two views is ambiguous. Figure 1.2 shows an example of the disparity matching ambiguity, where a patch in one of the views usually has more than one correspondence in the other view. In this example illustration each of the blue boxes in the left view can be matched with all other boxes in the right view. The goal of the disparity map estimation is to use the surrounding context, by connecting parts of the same object to reduce the ambiguity between these correspondences. Hence, the proposed technique should incorporate the knowledge, that all these patches belong to a larger object (i.e. the handrail), and use it to infer a

Figure 1.2: Visualization of the disparity ambiguity in matching pixels from the left view to the pixels in the right view (Middlebury Stereo Dataset [SHK+14]). Each of the blue boxes in the left view could be matched with all other boxes in the right view.

unique match for each of these patches.

Image restoration. Often the quality of a captured image is degraded by undesired artifacts; therefore, image restoration tools are required to remove and reduce these artifacts. These restoration tools are mostly designed to remove degradation artifacts such as noise, blurriness, and holes in captured images. Additional to software restoration tools, almost all of the consumer cameras use basic hardware restoration tools to produce images with acceptable quality for users. Therefore, image restoration is one of the most important tools in the area of computational photography.

Similar to the disparity map estimation problem, image restoration is an ill-posed problem and requires prior knowledge of natural images. An example for this is shown in Figure 1.3: the left image is captured with a low resolution camera sensor resulting in the middle image. Due to the extreme loss of information, one can relate the low resolution image to many other high resolution images, shown on the right hand side. In other words, these images would appear the same way if captured with the low resolution sensor. This makes our inference ambiguous; given the low resolution image, we cannot

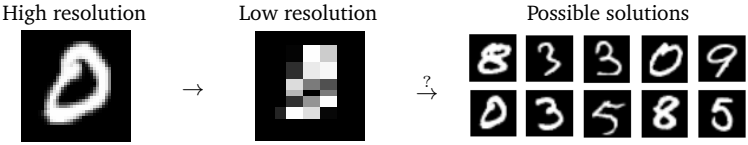


Figure 1.3: An example visualization for image restoration ambiguity (MNIST Dataset [LeC98]). The image on the left is downsampled to have lower spatial resolution, which results to the image in the middle. The task of restoring a high resolution image from the low resolution observation is called the *super-resolution* problem. Due to the loss of information, one can supper-resolve the low resolution image to many other images shown on the right side.

decide which of these high resolution images is our solution. Image restoration techniques use prior knowledge about image statistics (sharp edges, etc.) to find a more plausible solution.

In summary, the goal of this thesis is to develop novel priors that effectively encode prior knowledge about natural images and their distribution. Additionally, we aim to develop optimization techniques that leverage these priors and efficiently solve inverse problems, including disparity estimation and image restoration.

1.1 Contributions

This thesis contributes in the following two problems:

Temporally coherent disparity maps [BBZ16]. We propose a robust disparity map estimation technique that can be used for stereo sequences as well as single stereo pairs. A key contribution of this technique is the strong smoothness constraints that reduce the ambiguity of matching pixels between the views. To avoid degenerate solutions, we preserve depth discontinuities by incorporating a depth dependent similarity measure between pairs of pixels in an image. We

generalize this smoothness constraint to the temporal dimension and use it for stereo sequences to produce temporally coherent results. We show the robustness of our technique by comparing to the state of the art results for both cases of stereo sequences and single stereo pairs. We develop a new optimization technique with convergence guarantees and demonstrate that, in practice, our algorithm is stable. Furthermore, we provide a GPU implementation that can compute disparity maps of a sequence very efficiently.

Natural priors for image restoration [BZ17, BZFJ17]. A key contribution in this thesis is to propose novel and generic frameworks for various restoration tasks. We use analytical techniques to make interesting observations about the natural image distribution using empirical probability estimators. Specifically, we show that it is possible to learn the (gradients of) natural image distribution using empirical Bayesian lease squares denoisers. We parametrize these denoisers using deep neural networks to efficiently approximate the non-parametric approach, and we use them as our priors in two frameworks for image restoration. We design our first framework to address several image restoration tasks using a single neural network as the prior. Our second framework is designed to handle more general and ambiguous cases where the degradation noise variance and blur kernel are unknown. Finally, we show competitive results for various restoration tasks compared to the state of the art methods.

1.2 Thesis Organization

This thesis starts with a chapter on the proposed disparity map estimation technique, followed by three chapters on the proposed image restoration techniques. The thesis is structured as follows:

In **Chapter 2** we introduce our disparity map constraints for single stereo pairs and stereo sequences. We further present the disparity estimation framework in two schemes of parallel and sequential variable update. And we finalize this chapter by presenting quantitative

results of the proposed technique.

Chapter 3 gives an introduction to the image restoration problem followed by a brief review of the related work. We present the key ideas of image distribution parametrization using neural networks that we use later for image restoration.

In **Chapter 4** we describe our first approach to general image restoration using our prior. We explain the intuition behind our technique as well as the details of our algorithm. At the end of this chapter, we provide visual and numerical comparisons of our technique to other state of the art methods.

Chapter 5 presents our second formulation of the image restoration problem with a more generic approach. We complete this chapter by giving detailed explanation of our algorithm and comparison to the state of the art techniques.

Chapter 6 concludes our research and results and briefly describes possible future work.

Chapter 2

Temporally Coherent Disparity Maps

Disparity map estimation is one of the most classical problems in computer vision with many applications regarding the depth of the objects in the scene. Detecting obstacles in autonomous cars is an example application, that crucially relies on temporal coherency of the disparity maps. In applications like video production for 3D displays, also, temporally coherent disparity maps are crucial. While human observers are more forgiving about incorrect disparities, they easily notice flickering artifacts due to temporally incoherent disparity maps. While some disparity estimation methods leverage information over several frames of stereo video sequences, most do not attempt to produce temporally coherent disparity maps.

We address this problem by proposing a technique that produces temporally coherent disparity maps over stereo video sequences. We formulate an energy minimization problem consisting of unary, smoothness, and consistency terms, which we solve using the mean-field approximation of a densely connected conditional random field (CRF). Figure 2.1 shows a comparison of disparity maps from three techniques that support spatio-temporal disparity estimation, includ-

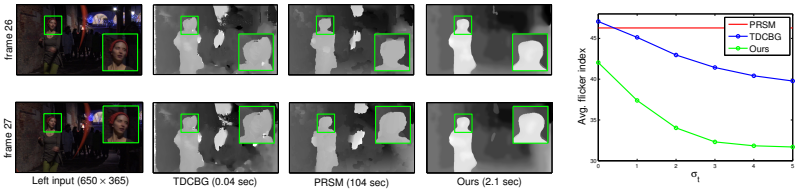


Figure 2.1: Our optimization includes the temporal dimension to achieve temporally coherent disparity maps in linear time. Here we compare disparity maps from TDCBG [ROD+10] using a temporal window of eight frames, PRSM [VSR15] using three frames, and our method using 21 frames. We also indicate the computation time per frame for each method. On the right we show the average disparity flicker index in this sequence. Our algorithm and TDCBG [ROD+10] allow controlling temporal smoothness using a temporal support parameter σ_t. Sequence courtesy of Media Leader Srl (www.medialeadersrl.com).

ing TDCBG [ROD+10], PRSM [VSR15], and our method. We use the maximum temporal support for each method, which is eight consecutive frames for TDCBG, three frames for PRSM, and 21 frames for our approach. On the right side of Figure 2.1 we show the average disparity flicker index in this sequence. The flicker index is a quantitative measurement of the temporal smoothness of a signal, and we compute it according to the IESNA standard [DHMS00]. Our algorithm and TDCBG [ROD+10] allow controlling temporal smoothness with a user specified parameter σ_t. Our proposed algorithm achieves the lowest flicker index, can be computed in linear complexity in terms of image resolution and number of frames, and our GPU implementation requires only a few seconds per frame.

We propose two efficient filtering techniques to solve the mean-field approximation, using parallel and sequential updates. Both have linear complexity in terms of the number of pixels in the input. Parallel updates allow us to process all pixels in a stereo sequence independently, enabling fast GPU implementations. In contrast to sequential

updates, parallel updates are not guaranteed to converge. We provide a detailed comparison between both techniques, and show that with proper initialization, parallel updates obtain the same quality of results. Hence they are preferable in practice. Finally, our method ranks among the state of the art in the KITTI benchmark [GLU12].

In summary, the contributions of this chapter are:

- A new smoothness term that leverages both left and right images to distinguish between image edges due to depth discontinuities, and edges due to surface texture.
- A novel consistency term to obtain a joint left-and-right disparity estimation formulation.
- A temporal smoothness term to achieve temporally coherent disparity maps over stereo video sequences.
- A novel technique for sequential mean-field update with a linear complexity in the number of variables, with extensive comparisons of different CRF optimization techniques.

The rest of this chapter is organized as follows: we discuss the background and previous work in Section 2.1. We introduce our energy formulation that includes a novel consistency term and the temporal extension in Section 2.3. Next, in Section 2.4 we discuss energy minimization via the mean-field approximation and using an iterative algorithm with parallel updates. Parallel updates are not guaranteed to converge, however, and we develop an efficient sequential approach in Section 2.5 that does not suffer from this problem. Finally, we evaluate our approach using standard datasets in Section 2.6.

2.1 Background and Related Work

In this section we describe the background and related contributions for different aspects of the disparity map estimation task. Most methods assume that the stereo inputs are rectified such that the disparity

is only in the horizontal direction, but similar to our technique, they are not limited to this setup.

Cost volume construction. Disparity map estimation is commonly defined as a discrete labeling problem. The estimation process usually starts by computing a cost for each pixel and each disparity hypothesis. These costs are then stored in a volume (array) of linear size in the number of pixels times the number of disparity hypotheses. Birchfield and Tomasi [BT99] use absolute pixel differences, between pixels in the left and right views, to compute the pixel-disparity cost. This approach is highly sensitive to noise ans illumination differences in the image pixels. Therefore, most methods compute this cost using a small patch centered at each pixel. This approach incorporates a basic assumption that the disparities in each patch are the same (i.e. each patch is parallel to the image plane). In this case, having a larger patch size improves the robustness of the cost. On the other hand, larger patch size can violate the uniform disparity assumption and can produce blurry results at depth discontinuities. For a more robust estimation, Spangenberg et al. [SLR13] propose to use a small neighborhood around the pixel and use Hamming distance (instead of absolute differences) between the patches from the two views to compute the cost. To get more semantically consistent costs, Žbontar and Yann [ZL15] use Convolutional Neural Networks to define a new cost function that leads to significant quality improvements, but incurs a high computational cost. In our work, we use a weighted combination of absolute differences and Hamming distance to get a robust estimation of the cost volume.

In practice, the estimated cost volume in these methods are still very noisy. And most often, in uniform regions, the cost values are ambiguous (i.e. there are no unique minimum cost). Therefore, it is necessary to refine this volume using post-processing techniques such as aggregation or optimization.

Cost aggregation. A simple approach to removing the noise in the cost volume is averaging and aggregating its values. This is simply

done by sharing the cost of each assignment with neighboring pixels to reduce noise. Rhemann et al. [RHB+11] use edge-aware distances between pixels to define contribution weights of each neighboring pixels' cost. By assuming that image edges correspond to depth discontinuities, using an edge-aware filter reduces the risk of sharing costs between regions with different disparities. Donatsch et al. [DBRZ14] use the same technique to aggregate the cost, but they consider geodesic distances to compute the contribution weights. For a better representation of the depth discontinuities, we use a different approach which incorporates depth information to compute the contribution weights.

Although efficient, aggregation technique does not evaluate the resulting costs after a single filtering step (i.e. whether or not the noise and ambiguity have been removed). Therefore, naive aggregation is unable to reason about complex assignment configurations. In our method, we use an optimization technique that incorporates more complex assignments by iteratively refining the global energy of the disparity map until the best configuration is found.

Disparity optimization. More robust disparity map estimation methods perform a global optimization to reduce ambiguities as well as noise. These optimization-based methods try to find the best disparity assignments by minimizing an energy function. A very common optimization scheme is to define smoothness connections between immediate neighboring pixels (4- or 8-connected neighborhoods), and incorporate high energy to penalize the assignments where neighboring pixels are assigned to different disparity values. Earlier methods use the Graph Cuts [BVZ01] algorithm to optimize this objective by minimizing the smoothness energy, which is very inefficient in time and memory. For an efficient alternative, Hirschmuller [Hir08] propose the Semi-Global Matching (SGM) optimization. This method approximates the original energy function, of the two dimensional disparity map, using a set of one dimensional scan-lines in different directions. They iteratively minimize the energy function of each scanline using dynamic programming, which is very efficient in practice. To further improve the efficiency of SGM, Herman and Klette [HK12]

propose to use the resulting disparity maps of each iteration to reduce the number of hypotheses (the search space) for the next iteration. Spangenberg et al. [SLR13] generalized SGM by including a weighting function between these directional scan-lines. This generalization improves the quality of SGM by adopting the weights for different structures in the scene (e.g. slanted planes).

While SGM is able to find a semi-global arrangement of disparity labels, it is unable to capture the fine details of local structures due to the simple energy formulation. Using image information and constraining further pixels, similar to aggregation-based methods, can help to capture more complex structures. Filter-based mean-field approximation [KK11] is a very attractive approach that enforces full-connectivity of disparity map pixels. Yu and Gallup [YG14] use this model, in which all disparity pixels are connected to all other pixels. Similar to their work, we use a fully-connected model that enforces a very powerful smoothness constraint, and we use the filter-based mean-field approximation for a fast and efficient optimization.

Vineet et al. [VWT14] further extend the optimization to include higher order terms that incorporate information about objects to be used in the disparity estimation problem. In our method, we incorporate very higher order terms to connect the disparity maps of the two views, and we jointly optimize for a pair of consistent disparity maps.

Initialization and convergence. Many methods use a multi-scale approach to increase their robustness to ambiguous regions in the disparity maps. For example, Zhang et al. [ZFM+14] use multiple scales and aggregate the cost between different scales such that the assignment is consistent in all scales. In a similar fashion, optimization-based methods such as Vineet et al. [VWST12] run the optimization on coarser scales to initialize finer ones. In our approach, we use the SGM method to initialize our optimization, which further incorporates other complex terms.

Filter-based mean-field approximation [KK11] performs efficiently by using a parallel scheme to update each disparity variable. However, this scheme is not guaranteed to converge to a stationary point, and

can lead to oscillations in the disparity assignments during the iterations. On the other hand, other optimization techniques that have convergence guarantees, such as SGM [Hir08] and Graph cuts [BVZ01], cannot handle full-connectivity in practice. We propose an alternative optimization scheme for filter-based mean-field approximation that uses sequential updates instead of conventional parallel updates, while still performing efficiently. We use this technique to analyze and validate the convergence of the parallel scheme for our energy function. And we show that, by using our initialization, the parallel scheme can achieve similar or better convergence properties than the sequential case.

Video sequences and flicker artifacts. Temporal coherence is a crucial factor in many applications of disparity maps. An intuitive approach for this challenge is to use several stereo frames (stereo sequence) and attempt to ensure temporal coherence between them. Slanted plane StereoFlow [YMU14] uses two consecutive frames to improve results. This method computes an initial disparity map using SGM and then jointly optimizes for planar surfaces and local segments. This approach is tailored for applications such as autonomous vehicles with an ego-motion assumption, which cannot be generalized to other and more general scenes. Vogel et al. [VSR15] use consistency factors between the views that are defined as a data term in their optimization. Using a piece-wise rigid model their method includes consistencies in the temporal dimension that incorporates neighboring views. Unlike these methods we do not enforce segmentation nor local planarity on our disparity maps. In addition, our method has linear complexity with respect to the number of frames, which allows us to compute the disparity maps of the whole sequence in a single optimization.

Disparity flicker artifacts have been previously studied [ROD+10, MLD12]. Richardt et al. [ROD+10] assumed that the pixel's disparity persist in time and aggregated the costs between temporally consecutive pixels. Min et al. [MLD12] filtered noisy disparity maps between different frames. Similar to their work we use a precomputed flow

field and enforce temporal coherence along its vectors. In addition to end-to-end disparity error, we propose a quantitative measure to better evaluate the flicker artifacts in disparity sequences and compare with previous works.

2.2 Problem Formulation

Let us denote X as the unknown disparity map. Given left and right images L and R, our goal is to estimate the corresponding disparity maps between them. We follow the literature [KF09a] to define an intuitive formulation of the disparity map estimation problem. We first define the influential variables in our estimation problem as random variables (random fields). Second, we represent the distribution of our variables using a graphical model.

Conditional random fields. We consider a stereo image random field (L, R) for left and right images. Similarly, we define a random field (X^L, X^R) for left and right disparity maps. Consequently, we can form their joint random field (X^L, X^R, L, R) indicating all possible left and right image pairs (L and R), and their corresponding disparity maps (X^L and X^R). When the left and right images are known, we can form the conditional random field $(X^L, X^R|L, R)$. Our goal in the disparity map estimation task is to find the most likely disparity maps $(X^L, X^R)^*$ for a given a pair of stereo images. Specifically, we would like to maximize the probability of the conditional posteriori by finding

$$(X^L, X^R)^* = \operatorname{argmax} \operatorname{Prob}(X^L, X^R|L, R).$$

In practice, we cannot measure this probability exactly even though, in theory, the above conditional distribution and its probability exist. One approach to approximate this probability is to use large disparity map datasets as done in data-driven methods. In practice, the available datasets are very limited in size and application since capturing and calibrating the left and right images and their

corresponding ground truth disparity maps is very challenging. To overcome this limitation, we avoid using any dataset by incorporating common rules and observations to design a novel graphical model to parametrize the disparity maps probability distribution.

Graphical model. We represent the probability distribution of our joint random field using a graphical structure [KF09a]. Specifically, we represent our disparity random field X using a graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$, with $\mathcal{V}$ denoting its set of nodes and $\mathcal{E}$ denoting its set of edges. The nodes of this graph represent the disparity map pixels, that is $\mathcal{V} = \{X_1, X_2, ..., X_n\}$, where X_i denotes the random variable at i-th pixel of the disparity map and n denotes the total number of pixels. In our graph formulation, we include an edge between any pair of pixels in the disparity map. In other words, for any i and j, we have an undirected edge $X_i \rightleftharpoons X_j \in \mathcal{E}$. We further extend our graphical representation to include disparity variables from both views by including nodes from both views, i.e. $\mathcal{V} = \{X_1^L, X_2^L, ..., X_n^L, X_1^R, X_2^R, ..., X_n^R\}$.

In the graphical formulation, any subset of the nodes in which all its nodes are connected with an edge is called a clique. Therefore in a fully-connected graph, any subset of the nodes can be used to form a clique. We use the notion of cliques in our graph to parametrize the distribution of our disparity maps. We characterize our conditional random field by a Gibbs distribution [KF09a] with graph $\mathcal{G}$ on (X^L, X^R):

$$\text{Prob}(X^L, X^R | L, R) = \frac{1}{Z} \exp\left\{ - \sum_{c \in \mathcal{C}_\mathcal{G}} \phi_c \Big((X^L, X^R)_c | L, R \Big) \right\},$$

where Z is a normalizing constant (partition function), and c denotes a clique from a set of all cliques $\mathcal{C}_\mathcal{G}$ in our graph, that contribute an energy therm ϕ_c to this characterization.

Intuition. In the formulation above we use a notation of the cliques in the graphs to define and parametrize our disparity map distribution.

We use cliques in our graph to define an acceptance measure for each local structure in the disparity map. Intuitively, by setting the energies ϕ of the graph cliques, we express the likelihood of structures. We set the clique energies to be high for unlikely structures and low otherwise.

We use this characterization to formulate and represent our disparity map estimation in a declarative manner. In Section 2.3 we define the clique structures of our disparity graph and design their energy functional ϕ. We propose an efficient optimization technique in Section 2.4 to maximize the conditional posteriori, and we compare to different optimization techniques using the same graphical model.

2.3 Energy Terms

In this section we describe our energy terms that characterize the spatio-temporal disparity estimation problem. We define variable assignments $X_i^L = x_i^L$ for the disparity values of pixels i in the disparity field X^L of the left image, and similarly x_i^R in X^R for the right image. In the rest of this work, we omit the left and right superscripts unless necessary. Our joint energy function over X^L and X^R includes unary (per-pixel), smoothness, and consistency terms.

2.3.1 Unary Term

We denote the cost of assigning disparity d to pixel i in the left image L by the unary term $\phi_u^L(x_i = d)$. We compute this term using a standard approach, which is based on edge differences and Census transform distances similar to Yamaguchi et al. [YMU14]. Specifically,

$$\phi_u^L(x_i = d) = \frac{1}{|N(i)|} \sum_{j \in N(i)} \left\{ |S_j^L - S_{j+d}^R| + \lambda_{cen} |H(T_j^L, T_{j+d}^R)| \right\},$$

where $\phi_u^L(x_i = d)$ is the unary cost of assigning disparity d to pixel i in the left image, S^L and S^R denote the response to the horizontal Sobel operator, H is the Hamming distance of the center-symmetric Census

transforms T^L and T^R introduced by Spangenberg et al. [SLR13], and $\lambda_{cen} = \frac{1}{3}$ is a constant that controls the relative weight of the two terms. The cost for pixel i is averaged over its 8-connected neighbors $j \in N(i)$. We compute the Census transform in a 7×7 window on the blurred image using a 3×3 box filter. This will increase robustness against artifacts such as noise and aliasing. The Census transform is a feature that represents the local arrangement of pixels in a neighborhood robust to brightness changes and noise by capturing if the brightness of a pixel is larger than the center pixel of that neighborhood. Since this transformation looses some textural information, adding the edge difference measure helps to better identify the matching pixels in the other view.

2.3.2 Disparity-Dependent Smoothness Term

The goal of the smoothness term is to encourage pairs of pixels that are close in some sense (defined more precisely below), to get similar disparity assignments. We define the smoothness term $\phi_s^L(x_i = d_i, x_j = d_j)$ for a pair of assignments $x_i = d_i$ and $x_j = d_j$ in the left image as a function of both the pixel locations i, j and the disparity assignments d_i, d_j (similarly for the right image). We express this term as a sum of weights $W^L(P)$ over all paths P that connect the points $\langle i, d_i \rangle$ and $\langle j, d_j \rangle$ in the joint pixel-disparity space,

$$\phi_s^L(x_i = d_i, x_j = d_j) = -\left(\sum_{P \in \mathcal{P}(i,d_i,j,d_j)} W^L(P) \right),$$

where $\mathcal{P}(i, d_i, j, d_j)$ is the set of all paths between $\langle i, d_i \rangle$ and $\langle j, d_j \rangle$ in the joint space of pixel locations and disparity hypotheses, and each path $P = \{\langle k, d \rangle\}$ is a sequence of (4-connected) pixels k paired with a disparity hypothesis d.

We define the weight kernel W based on three length functions of the path, its length $l_s(P)$ in the image, its length $l_d(P)$ in the disparity label space, and a length δ^L (discussed below) that takes into account potential disparity discontinuities along the path. Specifically, the

weight kernel is

$$W^L(P) = \exp\left\{-\left|\left|\frac{\delta^L(P)}{\sigma_r} + \frac{l_s(P)}{\sigma_s} + \frac{l_d(P)}{\sigma_d}\right|\right|_2^2\right\}, \tag{2.1}$$

where σ_r, σ_s, and σ_d control the kernel support for the three length terms. Applying a Gaussian weight to the sum of the three distances ensures that $W^L(P)$ decreases when the two pixels are separated by a large distance, and it increases when they are close. Because we sum the negative weights $W^L(P)$ over all paths, the smoothness energy (cost) decreases by the weight of each path, and each short path further reduces the energy. In contrast, Hosni et al. [HBGR09] use only the path with the minimum distance. A single path, however, is more sensitive to noise. Summing up the weights from all paths not only includes the weight from the shortest path, but also increases robustness to noise. Additionally, including all paths favors arrangements where assignments are connected by many long paths in contrast to assignments with few short paths. This choice of weight will later allow us to efficiently compute the smoothness energy.

The key ingredient in the definition of $W^L(P)$ is the length $\delta^L(P)$, which we design to become large when the path crosses depth discontinuities. Since depth discontinuities are not known a priori, either boundaries in superpixel segmentation [VSR15, YMU14] or image edges (pixel-wise differences) [ZL15, ZLL09, RHB+11, DBRZ14, ZFM+14, MSZ+11, KK11, VWST12] are conventionally used in their place. Many image edges, however, represent surface texture, not depth discontinuities, hence these approaches may lead to ineffective smoothness energies. Crucially, we consider color information from both (left and right) views to compute the path length $\delta^L(P)$ such that it depends on the disparities along the path P. For each disparity on the path, we compute a pixel-wise difference of the two views where one is shifted by that disparity. At pixels where the disparity happens to be the correct one, this will cancel image edges due to surface textures, indicating that these edges are not disparity discontinuities. If the disparity is wrong, image edges typically do not cancel. We use this intuition to define a disparity discontinuity indicator for pixel

k and disparity d as $\min(|L_k - R_{k+d}|, |L_k - L_{k-1}|)$, where L and R denote the left and right color images, and pixels $k-1$ and $k+d$ are horizontally offset from pixel k. Taking the minimum makes sure we do not introduce any spurious discontinuities. The path length $\delta^L(P)$ is now simply the sum of these disparity discontinuity indicators along the path,

$$\delta^L(P) = \sum_{\langle k,d \rangle \in P} \min(|L_k - R_{k+d}|, |L_k - L_{k-1}|).$$

This distance will be small if the pixel colors along the path have correspondences in the other image under their disparities, even if the image itself has large color dissimilarities along that path.

We visualize our approach in Figure 2.2. We show slices of the joint disparity-pixel space (d, i), where disparities d are along the horizontal axis, and the vertical axis corresponds to one vertical column of pixels i. The data is from a continuous, slanted surface patch that is highly textured (ground region in Figure 2.8, top left). Figure 2.2a shows conventional disparity discontinuity indicators given by pixel differences $|L_i - L_{i-1}|$, and Figure 2.2d are our proposed indicators $\min(|L_i - L_{i-1}|, |L_i - R_{i+d}|)$. Figure 2.2(a,d) show the ground truth disparities in red, and some estimated disparities consisting of fronto-parallel segments in green. In Figures 2.2(b,c,e,f) we visualize the smoothness energy for the red and green disparity assignments using the conventional and our approach. That is, each point (d, i) in these figures shows the sum $\sum_j \phi_s^L(x_i = d, x_j = \Delta_j)$ where the Δ contain either the ground truth (red) or estimated (green) disparities. We also indicate the total smoothness energy $\sum_{i,j} \phi_s^L(x_i = \Delta_i, x_j = \Delta_j)$. This shows that in the conventional approach some pixels have high smoothness energies even with the ground truth disparity assignment, and the total smoothness energy of the piecewise fronto-parallel disparities (green, Figure 2.2(c)) is actually lower than the ground truth (red, Figure 2.2(b)) here. With our approach, we obtain low smoothness energies at all pixels, and the ground truth (red, Figure 2.2(e)) has lower energy than the piecewise fronto-parallel assignments (green, Figure 2.2(f)).

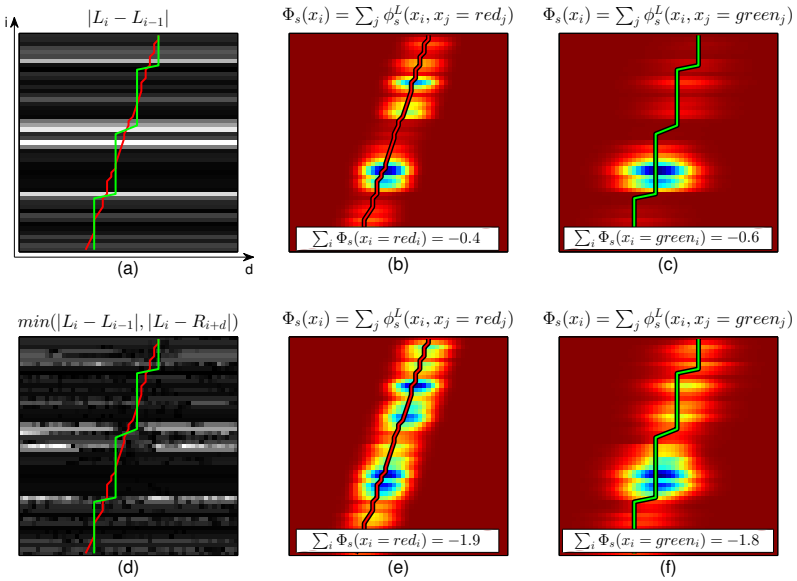


Figure 2.2: Visualization of the smoothness energy in the joint pixel-disparity space (pixels i on vertical axis, disparities d on horizontal axis). The top row shows the conventional approach, and the bottom row is our technique, where (a) is the conventional disparity discontinuity indicator, and (d) our proposed one. The red line in the left and middle column indicates the ground truth disparities, and the green line in the left and right column is a piecewise fronto-parallel disparity assignment. In the conventional approach, the piecewise fronto-parallel disparities incorrectly have a lower smoothness energy (-0.6 in (c)) than the ground truth (-0.4 in (b)). Our technique correctly leads to a lower energy for the ground truth (-1.9 in (e)) compared to the fronto parallel disparities (-1.8 in (f)).

2.3.3 Higher Order Local Consistency Term

Each disparity assignment indicates that the corresponding pixel appears with a shift (disparity) in the other image, therefore we expect that the disparity in the other view would agree with this assignment. We design the consistency energy to be low if the disparity assignments in two corresponding pixels in the left and right image agree. As a key idea, we compute this term over pixel neighborhoods, instead of individual pixels, to be more robust to per-pixel errors. We first introduce a binary consistency factor $\nu = [|x_j^L - x^R_{j+x_j^L}| \leq 1]$, which is one when two corresponding pixels x_j^L and $x^R_{j+x_j^L}$ (according to the disparity assignment in the left image) agree on their disparities up to a threshold of one disparity level, and zero otherwise. We allow for a difference of one disparity level to compensate for sub-pixel disparities and self occlusions. We now define the consistency energy as

$$\phi_c^L(x_i^L = d_i, x_j^L = d_j) = -\left(\sum_{P \in \mathcal{P}(i,d_i,j,d_j)} W^L(P)\right)\nu,$$

where we sum over all paths between joint pixel-disparity assignments x_i^L and x_j^L and use the same path weight $W^L(P)$ as for the smoothness term. Note that although this term is defined over pairs of disparity variables in one view, it implicitly involves a third disparity variable from the other view via the disparity compatibility function ν. Intuitively, given an assignment x_i^L, our consistency energy is low if many assignments x_j^L that are close to x_i^L in the left image, have consistent assignments $x^R_{j+x_j^L}$ in the right image. Since we cannot confirm consistency in the case of occlusions, we ignore them here and treat them later when finalizing the disparity map.

2.3.4 Temporal Extension

A main advantage of our filter-based CRF optimization (Section 2.4) is that we can easily extend it to the temporal domain, and simultaneously optimize disparity assignments over all frames of a stereo video

sequence. By extending the smoothness and consistency terms to the temporal dimension, we will obtain temporally coherent disparity maps that reduce flickering artifacts. We define the smoothness and consistency energies (ϕ_c , ϕ_s) as before, but now with weight kernels W over paths in the joint spatio-temporal and disparity domain,

$$W^L(P) = \exp\left\{ - \left|\left| \frac{\delta^L(P)}{\sigma_r} + \frac{l_s(P)}{\sigma_s} + \frac{l_t(P)}{\sigma_t} + \frac{l_d(P)}{\sigma_d} \right|\right|_2^2 \right\},$$

where $l_t(P)$ is the length of the path in time, and σ_t determines the kernel width along time. Our assumption here is that the disparities persist over a short time defined by σ_t. As a key idea, we define the temporal dimension by following flow vectors of a precomputed flow field over the video sequence. Specifically we use the flow by Lang et al. [LWA+12], and refer the reader to their paper for more details.

2.4 Energy Minimization

Here we describe our fast spatio-temporal energy minimization based on the mean-field approximation and filter-based parallel updates. We discuss our initialization and post processing steps, followed by a description of our GPU implementation.

2.4.1 Mean-Field Approximation

We define the global energy function E as a sum of the unary, smoothness, and consistency terms, all evaluated on both left and right images,

$$\begin{aligned} E(X^L, X^R | L, R) = & \sum_i \left\{ \phi_u^L(x_i) + \phi_u^R(x_i) \right\} \\ & + \lambda \sum_{i,j} \left\{ \phi_s^L(x_i, x_j) + \phi_s^R(x_i, x_j) \right\} \\ & + \gamma \sum_{i,j} \left\{ \phi_c^L(x_i, x_j) + \phi_c^R(x_i, x_j) \right\}, \end{aligned}$$

with parameters λ and γ to control the influence of the smoothness and consistency terms relative to the unary term.

As described in Section 2.2, we can relate this energy to the probability distribution of the disparity maps, which takes the form of a conditional random field (CRF),

$$\mathrm{Prob}(X^L, X^R|L, R) = \frac{1}{Z(L,R)} \exp(-E(X^L, X^R|L, R)), \qquad (2.2)$$

where $Z(L,R) = \sum_{X^L,X^R} \exp\left(-E(X^L, X^R|L, R)\right.$ is a partition function that normalizes the probabilities to add to one.

We minimize the energy function by following the mean-field approach [KF09a]. This approach approximates the distribution $\mathrm{Prob}(X^L, X^R)$ with a much simpler distribution $Q(X^L, X^R)$ in which the variables are marginally independent, that is $Q(X^L, X^R) = \prod_i Q_i^L(X_i^L)Q_i^R(X_i^R)$, where $Q_i^L(X_i^L)$ and $Q_i^R(X_i^R)$ are the marginal distributions of all variables (pixels) in the left and right images. Specifically, the original distribution Prob is approximated with the new distribution Q by minimizing their relative entropy (also known as the Kullback-Leibler divergence) defined as

$$\mathbf{D}(Q\|\,\mathrm{Prob}) = \mathbf{E}_Q\Big[\ln\frac{Q}{\mathrm{Prob}}\Big] = \mathbf{E}_Q\big[\ln Q\big] - \mathbf{E}_Q\big[\ln \mathrm{Prob}\big]. \qquad (2.3)$$

For simplicity, we formulate this approximation without the left and right index of the disparity maps and write the optimization as

$$\begin{array}{lrcl} \textbf{Find} & \{Q(X)\} & & \\ \textbf{Minimizing} & \mathbf{D}(Q\|\,\mathrm{Prob}) & & \\ \textbf{Subject to} & Q(X) & = & \prod_i Q_i(X_i) \\ & \sum_{x_i} Q_i(x_i) & = & 1 \quad \forall i \in L, R \end{array} \qquad (2.4)$$

where we force the probabilities in Q to add to one. We can use the dual form of the objective to get rid of the constraints by adding a Lagrange multiplier ν. We keep only the terms that involve Q_i and

write the Lagrangian at variable i as

$$\mathbf{L}_i[Q] = \mathbf{E}_Q\big[\ln Q(X_i)\big] - \mathbf{E}_Q\big[\ln \mathrm{Prob}(X_i)\big] + \nu\left(\sum_{x_i} Q(x_i) - 1\right). \tag{2.5}$$

Taking the derivative of the Lagrangian with respect to the assignment probability $Q_i(X_i = x_i)$ and setting it to zero we get [KF09a]

$$\ln Q_i(X_i = x_i) = \mathbf{E}_Q\big[\ln \mathrm{Prob}(X_i)|X_i = x_i\big] + \nu - 1. \tag{2.6}$$

Finally we take exponents of both sides and normalize,

$$Q_i(X_i = x_i) = \frac{1}{Z_i} \exp\Big\{\mathbf{E}_Q\big[\ln \mathrm{Prob}(X_i)|X_i = x_i\big]\Big\}, \tag{2.7}$$

where the Lagrangian parameter ν drops out due to the normalization. The approximate distribution Q is a stationary point of this equation at all assignments $X_i = x_i$. We can use this equation to iteratively update the probability of each variable assignment independently by computing the expected value of the energy conditioned to that assignment. We refer the reader to the work of Koller and Friedman [KF09a] for a more detailed derivation of this approximation. Using the formulation of our distribution in Equation 2.7 into the Equation 2.2, the stationary equation for our disparity distribution in the left image is derived as

$$\begin{aligned} Q_i^L(d) = \frac{1}{Z_i} \exp\Bigg\{ &- \phi_u^L(x_i) - \\ &\sum_j \Big(\lambda \mathbf{E}[\phi_s^L(x_i, x_j)|x_i = d] + \gamma \mathbf{E}[\phi_c^L(x_i, x_j)|x_i = d]\Big)\Bigg\}, \end{aligned} \tag{2.8}$$

where Z_i is again the partition function that is used to normalize the distribution over the variable x_i. The summation over j accumulates the expected values $\mathbf{E}$, conditioned to $x_i = d$, over all energy terms that include the variable x_i. The expected value for each smoothness

term, conditioned to $x_i = d$, is

$$\mathbf{E}[\phi_s^L(x_i, x_j)|x_i = d] = \sum_l \phi_s^L(x_i = d, x_j = l) Q_j^L(l), \tag{2.9}$$

and for the consistency term it is

$$\mathbf{E}[\phi_c^L(x_i, x_j)|x_i = d] = \sum_l \sum_{k=l-1}^{l+1} \phi_c^L(x_i = d, x_j = l) Q_j^L(l) Q_{j+l}^R(k). \tag{2.10}$$

Here, the sum over $k \in \{l-1, l, l+1\}$ corresponds to the compatibility function ν in Section 2.3.3. Although the consistency term ϕ_c is defined over three independent random variables, the expected value here is conditioned on the assignment of disparity d to pixel i, hence the conditional expected energy only depends on the probabilities of the two remaining variables Q_j^L and Q_{j+l}^R.

2.4.2 Filter-based Parallel Update Iteration

Algorithm 2.1 minimizes our energy by iteratively updating the mean-field distributions by computing Equation 2.8. The first iteration of the algorithm updates the disparity distribution of the left image (Q^L). In subsequent iterations we switch between updating the disparity maps of left and right images (line 5) to avoid oscillations between them. The notation implies that the operations are applied to all variables i and values d in parallel. The first two lines in the loop compute the expected values (Equations 2.9 and 2.10) and the summation over all pixels j in Equation 2.8. First (line 1), we compute intermediate values $\tilde{Q}_i$ that store the contributions that each pixel will make to the conditional expected energies of the smoothness and consistency terms of all other pixels. Next (line 2), at each pixel we simultaneously compute the expected values (summation over l) and accumulate the contributions from all the other pixels (summation over j) using a single, fast filtering operation over the intermediate values $\tilde{Q}_i$. We

Algorithm 2.1 Filter-based parallel update iteration to compute the mean-field approximation. We switch between updating the variables of the left and right image.

initialize Q^L, Q^R with SGM
loop $\#iterations$
1. $\tilde{Q}_i(d) \leftarrow \lambda Q_i^L(d) + \gamma \sum_{k,d-1\leq k\leq d+1} Q_i^L(d) Q_{i+d}^R(k)$
2. $\hat{Q}_i(d) \leftarrow \sum_{j,l}[-\sum_{P\in\mathcal{P}_{(i,d_i,j,d_j)}} W^L(P)\tilde{Q}_j(l)]$
3. $Q_i^L(d) \leftarrow \exp\left\{-\phi_u^L(x_i = d) - \hat{Q}_i(d)\right\}$
4. $Q_i^L(d) \leftarrow Q_i^L(d) / \sum_l Q_i^L(l)$
5. switch L and R

end loop

provide some more details about the filter implementation below. A single filtering step is possible since we have the same weights W defined in ϕ_s and ϕ_c. In line 3 the disparity potential is computed by adding the unary term, exponentiating, and normalizing to a distribution in line 4, which completes computation of Equation 2.8. Finally the iteration ends by switching the target distribution (line 5).

A key element of our algorithm is that we compute the path weights W efficiently using the Domain Transform Filter [GO11], which allows us to evaluate each filtering operation (line 2 of Algorithm 2.1) in constant time. We use interpolated convolution by iteratively applying a moving sum (box filter) in the transformed domain. The joint image and disparity space leads to 3D filtering (summing over j and l), and our temporal extension to 4D filtering over two spatial, the temporal, and the disparity dimensions. In the temporal dimension we filter along the precomputed flow vectors similar as Lang et al. [LWA+12]. We obtained our best results by iterating over passes along spatio-temporal directions and filter in the disparity domain at the end. We refer to the original publication [GO11] for more details about the Domain Transform Filter.

Initialization For initializing Algorithm 2.1 we leverage semi-global matching (SGM) [Hir08] with penalties $P_1 = 4$, $P_2 = 64$ in four directions. Instead of the MAP results of SGM, we rather use the obtained (min-marginal) energies to initialize our distribution $Q_i(d)$. For a better initialization, we run the first two iterations of the optimization using a large kernel support ($\sigma_s = 7$, $\sigma_r = 100$, $\sigma_d = 2$).

2.4.3 Final Disparity Map

We compute final disparities by finding the one with the minimum energy $-\log(Q_i(d))$ from Algorithm 2.1. For accuracy below the level of the disparity discretization we fit a quadratic to the three disparity costs centered at the minimum. We remove spikes by applying a 5×5 median filter. We fill occluded regions by checking for left-right consistency to find pixels with disparity differences higher than a threshold, and replacing disparities marked as occluded with the last non-occluded disparity in the left direction for the left view (similarly for the right view).

2.4.4 Implementation

The CPU version of the proposed pipeline supports 256 or more disparity hypotheses. We also implemented a GPU version for the whole pipeline that takes advantage of parallelism in the optimization at the pixel level. We ran our experiments on an Nvidia Titan Black graphics card with 6GB memory on board. We allocate memory for a batch of left and right images, including the disparity hypothesis layers requiring $2 \times Width \times Height \times Frames \times Disparities$ floating point values. Because of the limited GPU memory we are currently restricted to batches of 14 frames at a resolution of 960×540 and 32 disparity layers. Note that we evaluate the unary term at a finer discretization of disparity steps, typically at one pixel steps. We then store the minimum for each of the 32 layers. At the end of the optimization the disparity is computed and finalized as described above, and by fitting the quadratic to the 32 layers we achieve finer levels of disparity. After

the disparities of a batch of frames are computed, we move forward by seven frames and compute the disparities for the next batch. We finally interpolate the disparity values of the overlapping frames in consecutive batches for smoother transitions.

2.5 Convergence Analysis

Our proposed Algorithm 2.1 in Section 2.3 and other filter-based mean-field approximation methods [KK11, VWT14] update the random variables in the mean-field in parallel. While parallel updates lead to very fast implementations, they are not guaranteed to converge at all. The goal of this section is to answer two questions: First, how good are results obtained using parallel updates of the mean-field compared to sequential updates, which are guaranteed to converge to a fixed point? Second, how well can the mean-field approximate our energy functional compared to methods that do not make the same assumption? To answer the first question, we develop an efficient method that applies mean-field inference with guaranteed convergence using sequential updates, and we compare its results with the parallel implementation's. Second, we compare our approach with the minimized energy of Graph Cuts [BVZ01], which does not rely on the mean-field approximation.

To explain the sequential algorithm more clearly, we assume a simpler labeling problem on a single image with unary and smoothness energies, but without the consistency term between left and right images. The update equation of this problem (compare to Equation 2.8) simplifies to

$$Q_i(x_i = d) = \\ \frac{1}{Z_i} \exp\left\{ -\phi_u(x_i = d) - \sum_j \lambda \mathbf{E}[\phi_s(x_i, x_j)|x_i = d] \right\}. \tag{2.11}$$

Keep in mind, however, that this is only for explanatory purposes. We also implemented the sequential approach for the same energy

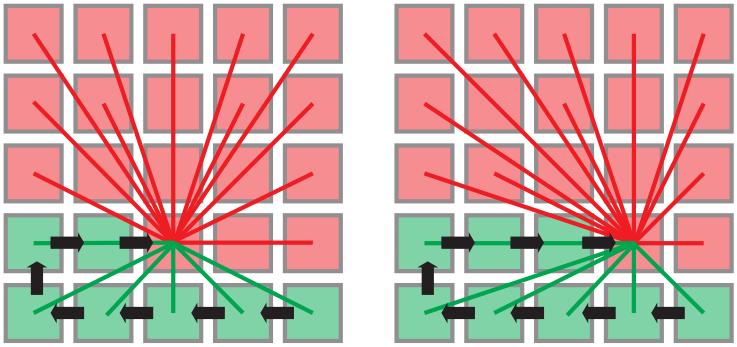


Figure 2.3: Sequential updates using a naive approach, proceeding from bottom right to top left. At each pixel i, we collect the contributions of all pixels that have not been updated yet (red) and already updated pixels (green) to implement the summation in Equation 2.11. Because we have a smoothness term between all pairs of pixels, this requires $O(N^2)$.

and update equations as in Section 2.3 for our evaluation. The main challenge is now to compute the summation over all variables j in Equation 2.11 efficiently, but sequentially over the pixels i. This is what we focus on next.

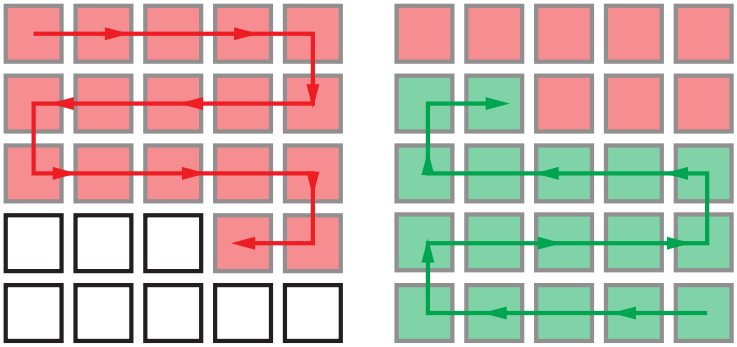
2.5.1 Sequential Updates for Mean-field Approximation

To optimize the mean-field approximation, each variable update needs to reduce the relative entropy (KL-Divergence) between the estimated and the true distribution [KF09a]. In the parallel scheme, while each variable tries to reduce its dependent energy in each update, all other variables change their distribution too, which invalidates the update in each variable. This could lead to oscillations in the distribution as well as being more prone to local minima in the energy functional.

Kolmogorov [Kol06] addressed a similar problem in max-product message passing optimization. Similar to this work, we develop a sequential iteration that updates a single variable in each step and therefore does not suffer from the same problems as the parallel scheme. We visualize the naive implementation of the summation over all pixels j in Equation 2.11 (similar as the one by Kolmogorov [Kol06]) in Figure 2.3. The black arrows indicate the sequence of variable updates, proceeding from bottom right to top left. Green variables are already updated, red ones have not been processed yet. Each update computes the expected energy of the current pixel (that is, variable) by summing up the contributions of all other variables. This is indicated by green and red lines in the figure, distinguishing contributions from previously updated variables (green) and not-yet-updated variables (red). Because we have a smoothness term between each pair of pixels, each variable update has linear complexity in the number of pixels. Updating all variables once has quadratic complexity, which makes this scheme computationally unattractive.

Leveraging constant time filtering. To make the sequential update practical, our key contribution is to leverage the constant time filtering technique by Gastal and Oliveira [GO15]. This approach allows us to accumulate the contributions of all pixels to the expected energy of each individual pixel (the summation over j for each i in Equation 2.11, illustrated by the red and green lines in Figure 2.3) in constant instead of linear time. Note that we compute the summation over all labels (Equation 2.9), which is required to complete the computation of the expected values, in an inner loop of our algorithm as explained later. We proceed using a two pass approach as shown in Figure 2.4, which involves first a *collection* and then an *update* pass:

- The *collection pass* (Figure 2.4(a)) traverses the pixels in the inverse order of the update sequence (compare to Figure 2.3). At each pixel, it collects the contributions from all variables that come later in the update sequence and stores them in a temporary buffer, shown in red. The key point is that we compute each step (each new red pixel) in this pass in constant time using the



(a) Collection pass (b) Update pass

Figure 2.4: Sequential update passes.

technique by Gastal and Oliveira [GO15], instead of linear time as illustrated in Figure 2.4(a).

- The *update pass* (Figure 2.4(b)) traverses the pixels in the update sequence (as in Figure 2.3). In each step, it accumulates the contributions to the current pixel from all previous pixels that have already been updated (green), again in constant time. In addition, we add the contribution from all pixels that have not been updated to the current pixel (that is, the value of the corresponding red pixel from Figure 2.4(a)) to complete the update of the current pixel.

We first give a brief explanation of the constant time filtering process for accumulating the contributions to the expected energy, and then show how the filter is employed in our two pass algorithm. Gastal and Oliveira [GO15] showed that processing signals with infinite impulse response (IIR) filters can be performed using a summation of first-order recursive operations. In other words, a K-th order IIR filter that needs K feedback operations per pixel, can be replaced with a summation of K first-order filters that need one feedback operation

per pixel. For a two dimensional signal f, two orthogonal 1D filters G in the horizontal direction and H in the vertical direction are used such that $H * G * f$ corresponds to a 2D filtering of signal f.

First, the horizontal filtered result $g_f = G * f$ at pixel (y, x) is defined using a set of K first-order recursive operations,

$$g^+_{f,s}(y, x, k) = a_k f(y, x) + b_k g^+_{f,s}(y, x - s, k), \tag{2.12}$$

$$g^-_{f,s}(y, x, k) = a_k b_k f(y, x + s) + b_k g^-_{f,s}(y, x + s, k), \tag{2.13}$$

where $k = 1 \dots K$, $g^+_{f,s}(y, x, k)$ and $g^-_{f,s}(y, x, k)$ are the causal and anti-causal responses of the k-th first-order filter of signal f with complex coefficients a_k and b_k at pixel (y, x). Then,

$$g_f(y, x) = \sum_{k=1}^{K} REAL\left[g^+_{f,s}(y, x, k) + g^-_{f,s}(y, x, k)\right] \tag{2.14}$$

is the response of the desired K-th order filter of signal f, which is computed by taking the real part of the summation of causal and anti-causal filter responses. The parameter $s \in \{1, -1\}$ indicates the direction of first-order filters, where $s = 1$ corresponds to a recursive operation from left-to-right in g^+ and right-to-left for g^-. Note that the choice of s does not influence the final filtered result g. Similar to the horizontal filtering we define $h_f, h^+_{f,r}, h^-_{f,r}$ for vertical filtering, where the direction $r \in \{1, -1\}$ manipulates the vertical index y. The 2D filtering of the signal f is then defined as

$$\begin{aligned} H * G * f &= h_g(y, x) \\ &= \sum_{k=1}^{K} REAL\left[h^+_{g,r}(y, x, k) + h^-_{g,r}(y, x, k)\right], \end{aligned} \tag{2.15}$$

which is the convolution of the two vertical and horizontal filters h and g. The reader is referred to Gastal and Oliveira [GO15] for more details about the filtering operations.

Next we show that the 2D filter formulation in Equation 2.15 can be computed recursively using the two-pass scheme as illustrated

in Figure 2.4(a, b). By expanding Equation 2.12 and 2.13, one can immediately see the relation between the causal and anti-causal filters, that is

$$g^{+}_{f,s}(y,x,k) = a_k f(y,x) + g^{-}_{f,-s}(y,x,k). \tag{2.16}$$

Using Equation 2.16 once for h and once for g in Equation 2.15, it is easily verified that the convolution result can be expressed by

$$h_g(y,x) = \tag{2.17}$$
$$C^{-}_{f,r,s}(y,x) + \left(\sum_{k=1}^{K} REAL[a_k]\right)^2 f(y,x) + C^{-}_{f,-r,-s}(y,x),$$

where

$$C^{-}_{f,r,s}(y,x) = \tag{2.18}$$
$$\sum_{k=1}^{K} REAL\left[h^{-}_{g,r}(y,x,k) + a_k \sum_{k=1}^{K} REAL\left[g^{-}_{f,s}(y,x,k)\right]\right].$$

The crucial insight from Equations 2.17 and 2.18 is that the 2D filtered output signal at pixel (y,x) is expressed as a sum of two contributions, $C^{-}_{f,r,s}(y,x)$ and $C^{-}_{f,-r,-s}(y,x)$, which represent the contributions from all pixels before (y,x) and all pixels after (y,x) in the update sequence. We compute $C^{-}_{f,-r,-s}(y,x)$ in the collection pass, and $C^{-}_{f,r,s}(y,x)$ in the update pass (Figure 2.4(a) and (b)). Note that the smoothness term between a pixel and itself is zero, hence the expected smoothness energy for a variable is a sum over all *other* variables. Therefore the middle term in Equation 2.17 is zero.

All values in Equation 2.17 can be computed with $O(K)$ operations, therefore the complexity to compute the expected energy is constant in the number of pixels and linear in the order K of the kernel function. Using this scheme, a Gaussian filter can be approximated perfectly ($MSE < 2.5 \times 10^{-8}$) by using two recursive filters, that is $K = 2$.

Efficient sequential update algorithm. Algorithm 2.2 shows the proposed sequential iteration of the mean-field approximation in a 2D fully connected grid with distribution Q using the update sequence from bottom right to top left (Figure 2.4(b)). First, the collection pass operates in reverse order (top left to bottom right) to compute and store the contributions to the expected energy from pixels in the sequence that have not been updated (Figure 2.4(a)).

In line 1, we compute the contribution to the expected energy from all previous variables on the current scanline using Equation 2.13, illustrated in yellow in Figure 2.5(a). In line 2 we compute Equation 2.18, also illustrated in Figure 2.5(a). We sum all contributions from previous variables in the horizontal (g^-, shown in yellow) and vertical directions (h^-, blue). This completes the collection step for the current pixel, and we store the result in a temporary buffer $\hat{Q}$. Next, lines 3–5 are needed to prepare for the next scanline. First we compute g^+ using Equation 2.12 in line 3, which we need to complete the horizontal filter g in line 4 (Equation 2.14). In line 5, we accumulate the horizontal contributions g in the vertical direction (h^-) to be used in the next scanline. This is visualized in Figure 2.5(b), where we apply the vertical anti-causal filter h^- to the horizontally filtered contributions g.

Second, in the update pass we now proceed in the update sequence order as in Figure 2.4(b), with analogous computations to the previous pass. Here we update the buffer $\hat{Q}$ by adding the contributions to the expected energies from the green (previously updated) half of the variables (line 7).

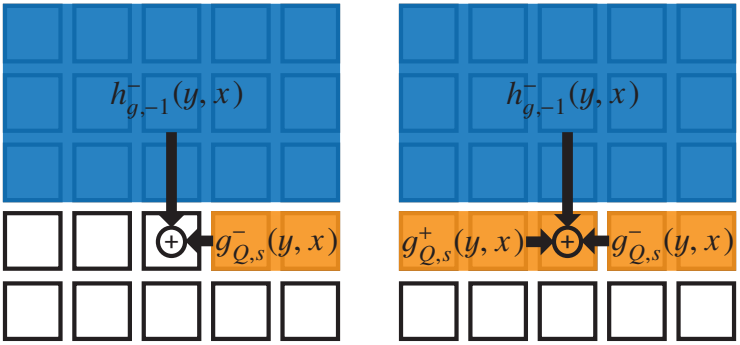
Note that in our algorithm we perform the update steps described so far for all hypotheses separately, but we omitted this in the notation for simplicity. To obtain the final expected energy of a pixel we now need to perform the summation over all hypotheses (Equation 2.9) in an inner loop (line 8). We also take into account the unary term here. The compatibility function of the hypotheses $w(d,l) = \exp(-|d - l|^2/\sigma_d^2)$ corresponds to the third factor in Equation 2.1. We then use the expected energy to update the distribution (line 9).

The proposed sequential update does not change the linear com-

Algorithm 2.2 The proposed sequential update for mean-field approximation. We explain each step in more details in Section 2.5.1.

$s \leftarrow -1$ // Collection pass Figure 2.4(a)
For $y : 1$ to $Height$
  For $x \in [1, Width]$ ordered by s // Process current scanline, Figure 2.5(a)
    1. $\forall k$, compute $g^{-}_{Q,s}(y, x, k)$ // Equation 2.13
    2. $\hat{Q}_{y,x} \leftarrow C^{-}_{Q,-1,s}(y, x)$ // Equation 2.18
  $s \leftarrow -s$
  For $x \in [1, Width]$ ordered by s // Prepare for next scanline, Figure 2.5(b)
    3. $\forall k$, compute $g^{+}_{Q,s}(y, x, k)$ // Equation 2.12
    4. compute $g_Q(y, x)$ // Equation 2.14
    5. $\forall k$, compute $h^{-}_{g,-1}(y + 1, x, k)$ // Figure 2.5(b)

$s \leftarrow 1$ // Update pass Figure 2.4(b)
For $y : Height$ to 1
  For $x \in [1, Width]$ ordered by s
    6. $\forall k$, compute $g^{-}_{Q,s}(y, x, k)$ // Equation 2.13
    7. $\hat{Q}_{y,x} \leftarrow \hat{Q}_{y,x} + C^{-}_{Q,1,s}(y, x)$ // Equation 2.17, 2.18
    8. $Q_{y,x}(d) \leftarrow \exp\left\{-\phi_u(\mathbf{x}_{y,x} = d) - \sum_l w(d, l)\hat{Q}_{y,x}(l)\right\}$ // Update
    9. $Q_{y,x}(d) \leftarrow Q_{y,x}(d)/\sum_l Q_{y,x}(l)$ // Normalize to distribution
  $s \leftarrow -s$
  For $x \in [1, Width]$ ordered by s
    10. $\forall k$, compute $g^{+}_{Q,s}(y, x, k)$ // Equation 2.12
    11. compute $g_Q(y, x)$ // Equation 2.14
    12. $\forall k$, compute $h^{-}_{g,1}(y - 1, x, k)$ // Figure 2.5(b)



(a) Constructing $\hat{Q}_{y,x}$ (lines 1 and 2 in Algorithm 2.2)

(b) Computing $h^-_{g,-1}(y+1,x)$ (lines 3-5 in Algorithm 2.2)

Figure 2.5: Visualization of the sequential update operations.

plexity of the algorithm in the number of pixels, however, it includes additional complex exponentials and multiplications for the IIR filtering ($O(NMK)$ for N pixels, M hypothesis and K-th order smoothness kernel). Although the sequential iteration is guaranteed to converge and minimize the KL-divergence, its result is biased with respect to the chosen update sequence due to the nature of the mean-field approximation (i.e. the result depends on the order in which variables are updated). To reduce this bias, in each iteration, we estimate the distribution over four sequences (top-to-bottom, bottom-to-up, left-to-right and right-to-left) and update with the mixture of these distributions. Methods such as Jaakkola et al. [JJ98] use the KL-Divergence to optimally mix mean-field distributions, however, we found that simply averaging them is enough in our case.

2.5.2 Convergence Results

We set up a toy experiment with synthetic data to compare the results of the parallel and sequential mean-field iteration. To have a

Method	Smoothness measure	Complexity (per iteration)	Convergence guarantee
Graph cut	direct	quadratic	yes
Naive MF	direct	quadratic	yes
Parallel MF w/o init.	direct/geodesic	linear	no
Proposed sequential MF	geodesic	linear	yes

Table 2.1: Properties of different optimization methods for a fully-connected graph. We compare the distance metric that is supported in each method for the smoothness term between each pair of variables. Additionally we compare the convergence guarantees and the running time complexity of one iteration in the number of random variables and hypothesis.

baseline for our comparison we also computed energies from Graph Cuts [BVZ01] with alpha-beta swaps. Further, in our experiment we include the parallel update algorithm initialized with our SGM approach, as described in Section 2.4.2, to check the effect of our initialization. The other methods in this comparison do not include this initialization step. In Table 2.1 we compare these methods in terms of smoothness measure, their time complexity and convergence guarantees. This comparison shows that the proposed sequential update is an attractive method that can perform efficiently while guaranteeing convergence.

To see the behavior of the methods in practice, we perform additional empirical comparison between them. Similar to Kolmogorov [Kol06], we compare the results from 50 randomly generated instances of unary data. Variables were distributed on a 39×29 grid with 16 hypotheses. The energy was defined over a fully connected graph with a smoothness term with Gaussian weights ($\sigma = 3$) between them. We used a single CPU core (3.5 Hz) for all methods.

Figure 2.6 shows the average energy (left) and KL-divergence (right) over the 50 random data instances for parallel, sequential, and SGM-initialized-parallel mean-field approximation implementations, in addition to Graph Cuts, as a function of computation time.

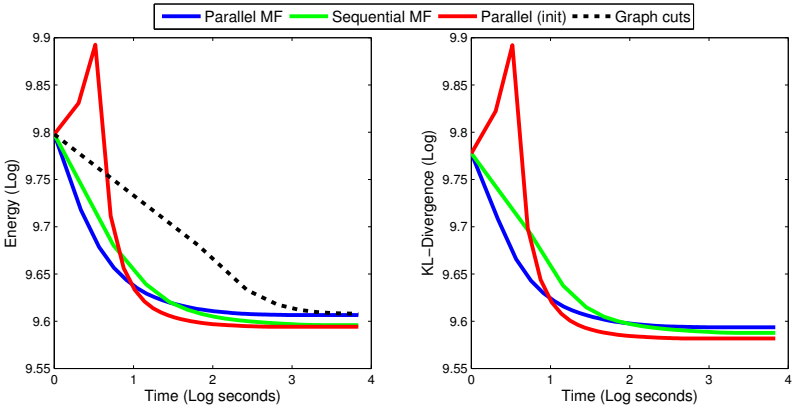


Figure 2.6: Convergence comparison between parallel, sequential, and SGM-initialized parallel mean-field updates. We also include Graph Cuts minimization of our energy, which does not rely on the mean-field approximation.

The minimized energy indicates that with Gaussian weights on a fully connected graph, the mean-field approximation performs well compared to Graph Cuts. Both sequential and parallel mean-field approximations have linear time complexity in the number of variables and hypotheses, hence they converge faster in contrast to Graph Cuts.

Without initialization, we observe that parallel updates (blue) converge to a higher energy and KL-divergence than the sequential approach (green). This confirms that sequential updates are more robust to local minima in the energy functional compared to the parallel approach. Initializing the distribution before parallel updates (red) using SGM (Section 2.4.2) leads to convergence to a lower energy and KL-Divergence, closing the gap to the sequential approach. This is because SGM (as the first iteration of Tree-Reweighted Message Passing [DHAH14]) can find the global establishment of the variables to some extent. After the initialization, the parallel updates can refine the local configuration of the variables more independently. Note

that at the beginning, the initialization increases the energy and KL-Divergences sharply, because it tries to minimize a much simpler energy functional that does not necessarily have the same solution as our desired energy.

It is interesting to see that SGM-initialized parallel updates perform better than the sequential approach in terms of the KL-Divergence (Figure 2.6(right)). This could be explained by the fact that, in contrast to the sequential approach, parallel updates do not suffer from directional bias. In practice, parallel updates can be implemented much more efficiently, for example using GPU devices, since operations can be done for each pixel separately. Therefore they are more attractive in practice. In the absence of a good initialization, however, the sequential update can be expected to obtain better results.

Finally, we compare sequential and parallel updates by computing end-to-end errors of the disparity maps from a subset of nine stereo images in the KITTI training dataset. In this comparison we used the full pipeline proposed in Section 2.3 using a single core CPU implementation of the IIR filter. We also include results from an SGM-initialized version of the sequential method to see if initialization has a similar influence as in the parallel case. Figure 2.7 shows the percentage of pixels that have disparities differing by more than three pixels from their ground truth. These results agree with our previous experiments on energy and KL-Divergence. Without initialization, sequential updates lead to better results than parallel ones. The SGM initialization improves both methods, and it closes the gap between them. The sequential update is about four times slower, however, since the distribution is computed with a mixture of four update sequences as described in the previous section.

2.6 Experiments and Results

As seen above, initialized parallel updates lead to best results in practice. Hence, in this section we are reporting results and evaluations

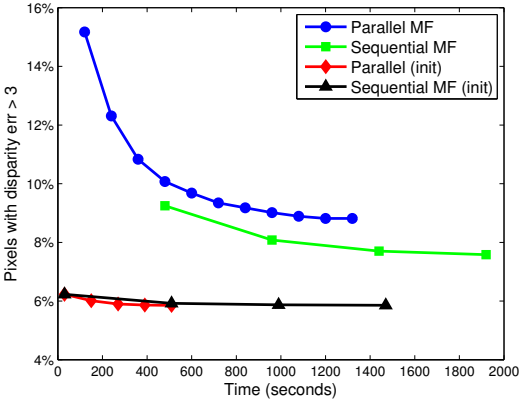


Figure 2.7: Comparing end-to-end results between sequential and parallel mean-field updates for a subset of KITTI stereo images.

of this technique as described in Section 2.3 in more detail.

2.6.1 KITTI Stereo Evaluation

We first tested our CPU implementation without the temporal extension on the entire KITTI [GLU12] dataset. We fixed parameters $\sigma_s = 4$, $\sigma_r = 6$, $\sigma_d = 4$, $\lambda = 10^9$, $\gamma = 50\lambda$, which we found by exhaustive search, and observed convergence after four iterations. Figure 2.8 illustrates our qualitative results from two scenes of the KITTI training dataset, where the first row shows the left input image, the middle row our final disparity map, and the last row the errors clamped to 5. In Table 2.2 we show the performance of each step of the proposed method in the KITTI training dataset. SGM initialization improves the quality about 30%. The proposed consistency term does not increase the computation time and further decrease the error by 10%. Table 2.3 summarizes the quantitative performance of our method on the KITTI test dataset. Our method obtains an average error of 3.32% for error threshold 3 and we rank number 8 on the list (in the time of submis-

Figure 2.8: Example results from the KITTI dataset. Top to bottom: left image, disparity map, and clamped disparity errors.

Included terms	$\% > 3$px	Time
ϕ_u	22.30	16 s
ϕ_u, ϕ_s	6.88	25 s
E_{SGM}	4.52	35 s
(init.) ϕ_u, ϕ_s	4.02	60 s
(init.) ϕ_u, ϕ_s, ϕ_c	3.67	60 s

Table 2.2: Performance gain in each step of the proposed method.

sion). Unlike other state of the art methods, the proposed method does not have simplifying assumptions about the scene geometry such as piece-wise planarity and does not assume prior knowledge on the data. Our CPU implementation compares to the rest in simplicity and scalability, and still obtains state of the art results.

2.6.2 Stereo Sequences

To measure the temporal coherence we compared the flicker index (IESNA standard [DHMS00]) of the final disparity maps. This index is computed in a temporal window of five frames as the ratio

Method	% >3px	% >4px	% >5px	Time
Displets [GG15]	2.47	1.94	1.67	265 s
MC-CNN [ZL15]	2.61	2.04	1.75	100 s
PRSM [VSR15] *	2.78	2.15	1.74	300 s
SPS-StFl [YMU14] *	2.83	2.24	1.90	35 s
VC-SF [VRS14] *	3.05	2.35	1.92	300 s
OSF [MG15] *	3.28	2.59	2.16	50 min
CoR [CXGZ15]	3.30	2.59	2.16	6 s
Ours	**3.32**	**2.45**	**1.96**	**60** s
SPS-St [YMU14]	3.39	2.72	2.33	2 s
PCBP-SS [YMU13]	3.40	2.62	2.18	5 min

Prior knowledge Planarity *: Flow

Table 2.3: The top 10 methods in KITTI benchmark by the time of submission. For an update list of methods visit evaluation webpage (http://www.cvlibs.net/datasets/kitti)

of the time-averaged disparities and the disparities above that average, which indicates how much disparities deviate from their average value in a temporal window (Figure 2.9). In Figure 2.1 we compare the average flicker index of our GPU implementation with Richardt et al. [ROD+10] and Vogel et al. [VSR15]. The plot on the right shows that we can significantly reduce the flicker index by enlarging the temporal smoothness kernel σ_t. In Table 2.4 we report the average computation times and flicker index over five video sequences with resolutions from 417×360 to 960×540. Our GPU implementation requires less than three seconds per frame, and with $\sigma_t = 5$ it produces significantly less temporal artifacts. Video results are available online for visual comparison.[1]

[1] http://www.cgg.unibe.ch/publications/temporally-consistent-disparity-maps

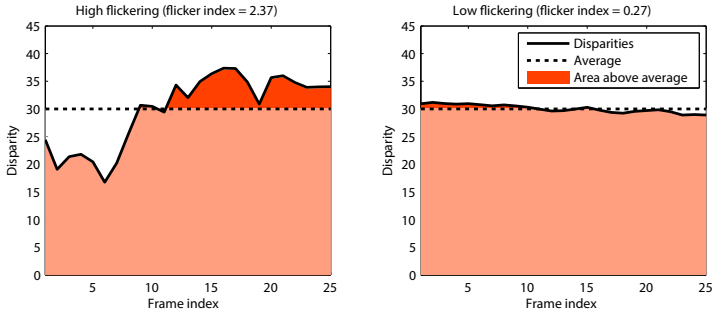


Figure 2.9: Visualization of the flicker index for tow disparity sequences. We compute the flicker index by taking the ratio of between the time-averaged disparities and the disparities above this average. This index is high in case of rapid changes of disparities in a time window (left), and is low for smooth transition of disparities (right).

2.7 Discussion

We have presented a robust method to compute disparity maps of stereo sequences in a single optimization. The optimization is solved efficiently using 4D filtering in pixel-disparity space. The proposed method ranks amongst the state of the art in challenging tests (KITTI) and produces less flicker artifacts in stereo videos.

We have developed a new and efficient filter-based optimization

Method	Time (sec)	Flicker
SGM	1.89	39.48
SPSS-St [YMU14]	1.62	47.95
PRSM [VSR15]	130.24	45.98
TDCBG [ROD+10]	0.06	35.21
Ours	**2.57**	**25.44**

Table 2.4: Comparison of flicker index and computation time.

algorithm that performs sequential variable update in the mean-field approximation. This algorithm guarantees convergence along with a decrease of the KL-Divergence in each iteration that is not available in previous filter-based mean-field approximation methods with parallel variable updates. In addition, our experiments showed that the new algorithm can perform well in comparison to Graph Cuts, a very well established optimization method. We showed that with an intuitive initialization, the parallel scheme can perform as well as the sequential method. However, the right initialization might not be available all the time, in which case, the proposed sequential algorithm can be used instead.

Chapter 3

Natural Priors for Image Restoration

Image restoration is one of the main topics in the field of computer vision and graphics. Almost all digital image capturing systems require restoration and refinement steps to produce meaningful and pleasant images. Image demosiacing, denoising, super-resolution are the primary applications of the restoration techniques used in these systems. More advanced techniques such as image deblurring and inpainting are generalizations of these primary applications.

Image restoration tasks are generally ill-posed problems, whose solution requires effective image priors (prior knowledge). These priors plays the main role on regularizing under-determined restoration problems. Therefore the prior is the core of each restoration technique and, one way or another, all methods that implement an image restoration technique assume some prior about the natural image.

Deep learning has been successful recently at advancing the state of the art in various low-level image restoration problems including image super-resolution, demosaicing, and denoising. The common approach to solve these problems is to train a network end-to-end for a specific task, that is, different networks need to be trained for

each noise level in denoising, or each magnification factor in super-resolution. This makes it hard to apply these techniques to related problems such as non-blind deconvolution, where training a network for each blur kernel would be impractical. The main reason for this limitation is that in neural networks the prior is manifested in the restoration operations and steps (i.e. the routine code and algorithm). Therefore, using a single pass of a trained neural network is not effective to remove other degradations. An alternative strategy to approach image restoration problems is to design or learn priors separately from the operations and steps, that can successfully constrain these under-determined problems

We propose a generic formulation that benefits from a learning stage for prior construction. After this, the proposed priors can be used, off the shelf, for many image restoration tasks. In summary, the contribution of this chapter is to propose a generic class of priors based on denoising autoencoders (DAEs) that can be used for various restoration tasks. The rest of this chapter is organized as follows: In Section 3.2 we describe the standard and commonly used image degradation model. We discuss the related work in Section 3.1. At the end of this chapter, in Section 3.3, we describe our prior derivation.

3.1 Problem Formulation

In this section we describe the standard image degradation model [JZSK09], and explain different restoration strategies. We model degradation including blur, noise, and downsampling as

$$y = D(k * \xi) + n, \quad n \sim \mathcal{N}(0, \sigma_n^2), \tag{3.1}$$

where ξ is the unknown image, k is the blur kernel, D is a downsampling operator using point sampling, n is zero-mean Gaussian noise with variance σ_n^2, and y is the observed degraded image (Figure 3.1). Like many other models, this degradation forms an a posteriori probability distribution $p(x|y)$, which indicates the probability of a solution x, given the observed, degraded image y.

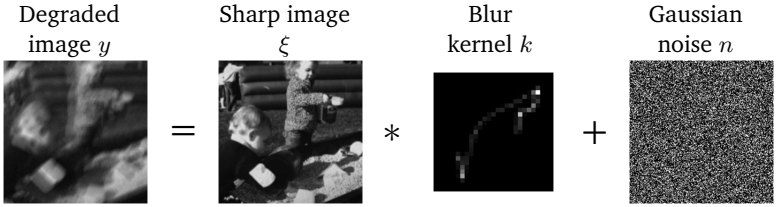


Figure 3.1: Our framework is built to restore degraded image y that is modeled as an unknown image ξ, blurred by convolution kernel k and corrupted with additive Gaussian noise n with variance σ_n^2.

In light of the ambiguity in image restoration, there are two major estimation strategies that are used for this problem. **Minimum mean squared error (MMSE)** estimators address this ambiguity by providing the estimate that is the closest to all possible solutions, weighted by their probabilities, that is

$$x_{\text{MMSE}} = \underset{x}{\operatorname{argmax}} \int_{\bar{x}} \|x - \bar{x}\|^2 p(\bar{x}|y) d\bar{x}. \tag{3.2}$$

The intuition behind this objective is to incorporate each image $\bar{x}$ as a candidate solution of the degradation. The optimal estimator of this equation is the average of these candidates weighted by the posterior probability of the degradation model $p(\bar{x}|y)$. An issue regarding these estimators is that the final estimation is, most often, not a natural image due to the averaging.

An alternative strategy is to use a **Maximum a Posteriori (MAP)** estimator in the restoration. This estimator intuitively returns the most probable solution of the posteriori distribution, that is

$$x_{\text{MAP}} = \underset{x}{\operatorname{argmax}}\, p(x|y) = \underset{x}{\operatorname{argmax}}\, p(y|x)p(x), \tag{3.3}$$

where $p(y|x)$ includes the data dependency to observed image y and $p(x)$ encodes the prior knowledge about the distribution of natural images (independent of the degradation model). The MAP estimator

is often defined in the logarithmic domain as a minimization problem

$$x_{\text{MAP}} = \underset{x}{\operatorname{argmin}} - \log p(y|x)p(x) = \underset{x}{\operatorname{argmin}} - \log p(y|x) - \log p(x), \tag{3.4}$$

that is often simpler and more practical to optimize. Unlike the MMSE, the solution of MAP is guaranteed to be a probable natural image.

The choice of an estimator highly depends on its application (e.g. entertainment, medical science), and sometimes it is more practical to define a new estimator to increase performance and robustness. For example in neural networks, the absolute error is often used in place of the squared distance of the MMSE estimators to produce sharper results. Another example is the use of the relaxation techniques in the MAP estimators that increase the robustness of the optimization with respect to the bad local minima. Most methods, irrespective of the type of the estimator, focus on providing a good approximation for prior $p(x)$, while assuming that the data dependency is known. Others focus on problems that the degradation model is unknown and try to approximate its parameters (e.g. blur kernel and noise variance), while restoring the sharp image. In the next section, we describe these methods and discuss their relevance and differences to our approach.

3.2 Related Work

Image restoration, like many other inference techniques, can be categorized into two classes of algorithms. **Procedural** methods work in an end-to-end fashion by incorporating the knowledge (prior) inside their routine code. These approaches are usually implemented with neural networks, using training examples to minimize the MMSE objective. One limitation of these methods is that they cannot incorporate their knowledge outside of the scope of their application. Alternatively, **declarative** methods separate their reasoning from their prior knowledge. In these techniques, the algorithm makes queries

to a knowledge bank (prior) about the state and quality of their estimation. These methods usually estimate the MAP solution and can be adopted more easily to other applications in the same domain (image restoration). In these methods, the knowledge is required to be stored in a suitable format to be accessed by the algorithm. This leads to additional processing overhead for making queries to the prior at every step. In the remaining of this section, we discuss relevant work in these two classes and explain their techniques to attain and use their prior.

3.2.1 Procedural and End-to-End

Deep learning has been very successful recently at advancing the state of the art in various low-level image restoration problems including image super-resolution, deblurring, and denoising. Solving image restoration problems using neural networks seems attractive because they allow for straightforward end-to-end learning. This has led to remarkable success for example for single image super-resolution [DLHT14, GZX+15, DLHT16, LWW+16, KKLML16] and denoising [BSH12, MSY16]. Common approaches to solve these problems train a network end-to-end for a specific task, that is, different networks need to be trained for each noise level in denoising, or each magnification factor in super-resolution. A disadvantage of the end-to-end learning is that, in principle, it requires training a different network for each restoration task (e.g., each different noise level or magnification factor). This makes it hard to apply these techniques to related problems such as non-blind deconvolution, where training a network for each possible blur kernel would be impractical. While a single network can be effective for denoising different noise levels [MSY16], and similarly a single network can perform well for different super-resolution factors [KKLML16], it seems unlikely that in non-blind deblurring, the same network would work well for arbitrary blur kernels. Additionally, experiments by Zhang et al. [ZZC+16] show that training a network for multiple tasks reduces performance compared to training each task on a separate network. Previous re-

search addressing non-blind deconvolution using deep networks includes the work by Schuler et al. [SCBHS13] and more recently Xu et al. [XRLJ14], but they require end-to-end training for each blur kernel.

In the area of neural networks, some of the focus has been on the architectural improvements. Kim et al. [KKLML16] employs the common residual learning scheme by adding the input image of the network to its final output layer. This scheme allows the use of deeper networks by providing speed and stability during training. Tai et al. [TYLX17] propose a deeper networks architecture that effectively uses less or equal number of parameters than conventional methods. They use dense skip connections in a form of long-term memory that are adaptively weighted using short-term memory blocks. Apart from the efficiency and performance, network architecture plays an implicit role on the regularization effect. Experiments by Ulyanov et al. [UVL17] show that, without the need for training, a well designed network can be used effectively to regularize various restoration tasks. When designing a network, one implicitly make an assumption that the underlying image distribution can be expressed by the current architecture. Therefore, any image structure that is not representable by the current architecture, will be regularized out during the process.

Another challenge in neural networks is to define a suitable loss function to be used during training. Conventionally, the Euclidean norm is used to define the distance between the network's output and the true solution. Johnson et al. [JAFF16] use the so-called perceptual loss for image super-resolution. This loss incorporates the Euclidean norm in the feature space of the image rather than the pixels directly, and usually leads to a sharper solution. Another common approach is the use of adversarial loss [GPAM+14] that enforces the solution of the network to be a sample from the image distribution. This approach, when trained well, usually produce very sharp results compared to other techniques. For a more stable training and better results, Ledig et al. [LTH+16] propose a combination of losses that includes pixel-wise Euclidean norm, perceptual loss, and the adversarial loss. Although these methods produce visually more appealing

results, they are still limited in their application similar to other neural network approaches.

Apart from neural networks, Yang et al. [YWHM10] proposes a dictionary based representation of image patches for image super-resolution. Their method consists of learning two dictionaries for low- and high-resolution image patches and a mapping between them to compute the high-resolution results. Schmidt et al. [SJN+16a] use a cascade of regression trees with applications in image denoising and deblurring. These methods also requires an end-to-end training with corresponding pairs of degraded and sharp images, which limits their prior to be used generally for other restoration tasks.

3.2.2 Declarative and Generic

As mentioned before, a key idea of declarative models is to separate the knowledge from the algorithm. For image restoration problems, this knowledge is usually referred to by the natural image prior or the prior for short. In the last decades, several natural image priors have been proposed. Earlier methods employ hand-crafted priors based on low-level image statistics, by making assumptions about the underlying image structure (e.g. edge sparsity). Although simplistic, these methods are fast and they can achieve good quality results in practice. On the other hand, data-dependent priors learn a representation of the natural image distribution, and usually achieve better performance in terms of quality. Most of these methods make assumptions about the image distribution (e.g. Gaussian mixture models) to be able to efficiently encode and employ their prior.

Hand-crafted priors. These priors are very attractive due to their simplicity and efficiency, and we give a brief description on some of the successful and recent hand-crafted priors. The main idea in these priors comes from a common observation that the natural images have sparse gradients (edges). Using this observation, these priors try to measure the likelihood of an image by counting the number of its edges. The Total Variation (TV) regularizing method of Rudin et

al. [ROF92] is a very popular example of such hand-crafted priors. In this method, the prior is simply the sum of absolute gradients in the image, which corresponds to the L_1-norm over the image gradients. Using the larger class of Hyper-Laplacian regularizers, Krishnan and Fergus [KF09b] propose an optimization scheme that incorporates $L_{\frac{1}{2}}$- and $L_{\frac{2}{3}}$-norms for image gradients. Perrone and Favaro [PF16] extend the TV prior by using an even sparser representation of image gradients. They use the image gradients norm in the logarithmic domain and they show that, in the limit, their approach approximates the L_0-norm. In contrast to L_1, L_0-norm does not penalize edge magnitudes, but only penalizes the number of non-zero edges.

Our work has an interesting connection to the work of Romano et al. [REM16], where they designed a prior model that is implemented by a denoiser function. Interestingly, the gradient of their regularization term boils down to the residual of the denoiser, that is, the difference between its input and output, which is the same as in our approach. However, their framework does not establish the connection between the prior and the natural image probability distribution, as in data-dependent approaches.

Data-dependent priors. Earlier methods for learning priors focused on low level image statistics such as gradients. Tappen et al. [TRF03] use Guassians to model the distribution of directional gradients in images, and image samples to fit the model parameters. Portilla et al. [PSWS03] took a similar approach to model the distribution of wavelet coefficients (in contrast to simple gradients). To get a more general approximation, Fergus et al. [FSH+06] use a mixture of Gaussians to model the distribution of image gradients. On the other hand, Fattal [Fat07], proposes graphical modeling to capture a representation for image gradients.

While some of these techniques are tailored for small degradations, patch-based priors achieve superior results for more challenging applications, such as deblurring with large kernels. Aharon et al. [AEB06] use k-SVD to learn a dictionary of patches for denoising. The key idea in their work is to provide a collection of patch representations (i.e.

a dictionary). They constrain the dictionary such that any natural patch can be very well represented by only a few combinations of its elements. For a more compact representation, Roth and Black [RB05] use Markov random fields to model images using local patch filters. Zoran and Weiss [ZW11] use a simpler, more tractable, approach by modeling image patches using mixtures of Gaussian distributions (GMMs). This approach, called EPLL, is very effective in practice and is a common baseline for performance comparisons.

Instead of parametrizing the patch distribution (e.g. using GMMs), Levin and Nadler [LN11] took a non-parametric approach for image denoising, which can be easily generalized to other restoration tasks. They use a very large dataset and evaluate the patch likelihood at runtime. This is not a practical approach for restoration since it requires comparisons of patches in the estimated image to all patches in the dataset. However, subsequent work [LNDF12] use this approach to provide effective bounds of optimality and complexity of denoisers with respect to the patch size.

Plug-and-play denoisers. Our approach is most related to techniques that leverage Alternating Directions Method of Multipliers (ADMM) to regularize the inverse restoration problem. These techniques build on the observation by Venkatakrishnan et al. [VBW13] that many algorithms that solve image restoration via MAP estimation only need the proximal operator of the regularization term, which can be interpreted as a MAP denoiser [MMHC17]. Venkatakrishnan et al. [VBW13] build on the ADMM algorithm and propose to replace the proximal operator of the regularizer with a denoiser such as BM3D [DFKE06] or NLM [BCM05]. Unsurprisingly, this inspired several researchers to learn the proximal operator using CNNs [CLP+17, ZZGZ17, XHH+17, MMHC17]. Meinhardt et al. [MMHC17] consider various proximal algorithms including the proximal gradient method, ADMM, and the primal-dual hybrid gradient method, where in each case the proximal operator for the regularizer can be replaced by a neural network. They show that no single method will produce systematically better results than the others.

In the proximal techniques the relation between the proximal operator of the regularizer and the natural image probability distribution remains unclear. While their use of a denoiser is a consequence of ADMM, our work shines a light on how a trained denoiser is directly related to the underlying data density (the distribution of natural images). We explicitly use the Gaussian-smoothed natural image distribution as a prior, and we show that we can learn the gradient of its logarithm using a denoising autoencoder. Our approach also leads to a different, simpler gradient descent optimization that does not rely on ADMM approximation.

A key idea of our work is to train a neural denoising autoencoder, that we use as a prior for image restoration. Autoencoders are typically used for unsupervised representation learning [VLL+10]. The focus of these techniques lies on the descriptive strength of the learned representation, which can be used to address classification problems for example. In addition, generative models such as generative adversarial networks [GPAM+14] or variational autoencoders [KW14] also facilitate sampling the representation to generate new data. Their network architectures usually consist of an encoder followed by a decoder, with a bottleneck that is interpreted as the data representation in the middle. The ability of autoencoders and generative models to create images from abstract representations makes them attractive for restoration problems. Notably, the encoder-decoder architecture in Mao et al.'s image restoration work [MSY16] is highly reminiscent of autoencoder architectures, although they train their network in a supervised manner.

Internal priors. The methods we discussed mainly bring knowledge about the image distribution form an external set of images. Handcrafted or data-driven, these methods are based on seperate observations, that are different from the input (degraded) image in process. Non-local means [BCM05] is one of the most popular methods that implements an internal prior for image denoising. They key idea in their method is that structures in image patches have, many, similar correspondences in the same image. By finding the corresponding set of

similar patches, they compute their average to remove the noise from the image. Intuitively, the more corresponding patches are found, the better the quality of the noise-free estimate will be. Another successful approach, called BM3D [DFKE06], extends this idea by using a more elaborate technique to remove the noise form the set of corresponding image patches.

The simple underlying assumption of these techniques is, in practice, very effective. Extensive experiments by Plotz and Roth [PR17] reveal that for real camera noise removal, BM3D outperforms other external-based-prior methods. This can be explained by the fact that, unlike restoration techniques using external priors, BM3D makes very few assumptions about the underlying data and its degradation noise distributions. However, these methods are still ineffective for restoration tasks other than denoising. An example failure case is image deblurring, where all patches in the image are similarly blurred. In this case, finding and incorporating similar patches has no meaningful regularizing influence.

3.2.3 Noise- and Kernel-Blind Deconvolution

It is also worth mentioning methods that try to estimate the degradation model parameters such as the degradation noise variance and blur kernel. Amongst these methods, Kernel-blind deconvolution has seen the most effort recently. Perrone and Favaro [PF14] used a constrained optimization regularized by gradient sparsity. After obtaining an estimate of the blur kernel, they use a separate deconvolution technique to restore the sharp image. In a similar fashion, noise-blind deblurring is usually performed by first estimating the noise variance and then applying the restoration using the estimated variance. Zhang et al. [ZW13, ZY14] explored a spatially-adaptive sparse prior and scale-space formulation to handle noise- or kernel-blind deconvolution. Jin et al. [JRF17] proposed a Bayes risk formulation that can perform deblurring by adaptively balancing the regularization weight. The blind methods, however, are mostly tailored for the specific task of image deconvolution. And they can usually handle one case of

either noise- or kernel-blind. Our work extends the work of Jin et al. [JRF17] by providing a Bayesian framework that can incorporate a generic prior, and consequently can be used for various tasks such as the joint noise- and kernel-blind deconvolution.

3.2.4 Summary of Priors

We summarize some of the most popular and recent image restoration techniques and their characteristics in Table 3.1. We compare the **domain** of knowledge for these methods and see that most successful methods employ external datasets to learn their priors. We also compare these methods by the **feature** that each prior regularizes and see that recent methods tend to measure the global image likelihood in the place of small patches and gradients. Rather than operating on individual patches, BM3D [DFKE06] operates on a set of patches in a form of a patch blocks, which implicitly extends the regularization area of this method.

By comparing the type of the **prior** in these methods, we can see that most popular methods use a representation of the density of their features as opposed to sparsity to regularize their reconstruction. This is mainly because sparse representations assume a specific distribution of the image features. As example of this occurs in the Total Variation (TV) norm and its variants, where the priors force Laplace-type distributions on the image gradients. Therefore, most successful methods use a more flexible representation of the true density and tune their parameters by a set of data samples. Although these methods try to propose a good representation of the true density, often they use approximations for tractability, efficiency, and implementation reasons. The kernel density approximation method, is one of the most effective approaches to learn an intuitive representation of the distribution using a dataset with discrete set of image examples. Markov Random Fields (MRFs) are less effective to represent the distribution since their parametrization using Graphical models are too simplistic.

The **parametrization** is an effective property of these methods that could influence the prior representation. For example, the

Method	Domain	Feature	Prior	Parametrization	Application
TV [ROF92]	internal	gradient	Sparsity	L_2-norm	generic
NLM [BCM05]	internal	patch	MRFs	Gaussian	generic
BM3D [DFKE06]	internal	patch blocks	Sparsity	DCTs	generic
FoE [RB05]	external	patch	MRFs	t-distribution	generic
KSVD [AEB06]	external	patch	Sparsity	Dictionaries	generic
EPLL [ZW11]	external	patch	True density	GMMs	generic
VDSR [KKLML16]	external	image*	True density	CNNs	upsampling
DnCNN [ZZC+16]	external	image*	True density	CNNs	denoising
DJDD [GCPD16]	external	image*	True density	CNNs	demosaicing
GradNet [JRF17]	external	image*	Kernel density	Gumbel, CNNs	deblurring
IRCNN [ZZGZ17]	external	image*	True density	Cascade of CNNs	generic
DAEP (ours)	external	image*	Kernel density	CNNs	generic
DMSP (ours)	external	image*	Kernel density	CNNs	generic

Table 3.1: A summary of the characteristics of some of the most popular and recent methods for image restoration. *In practice, these methods use a very large region of the images, which is considerably bigger than the usual patch size.

EPLL [ZW11] method parametrizes the true distribution using a mixture of Gaussians, which leads to a very simplifying representation of the complex patch distribution in practice. Recent methods benefit from the efficiency and the power of the convolutional neural networks (CNNs) compared to the more classical approaches of GMMs and MRF potentials. By comparing the **application** of these methods, however, we see that most CNN-based methods are not generic since they are trained for a specific task.

In light of the advances in constructing priors, Shaham and Michaeli [SM16] propose an interesting approach for understanding image priors. They provide a prior visualization technique to help understand what type of structures are preferred by which priors. We show a visual comparison for some of the popular priors using this technique in Figure 3.2. Basic priors (such as TV, KSVD, and NLM) that make simplifying assumptions about the images fail to capture the detail structures of the image. Other priors tend to prefer simple image structures such as straight lines and sharp corners over the complex and curved edges. We refer the reader to the original work [SM16] for a more comprehensive review and visualization of previous image priors.

3.3 Denoising Autoencoder as Natural Image Prior

In this section, we describe our approach for learning priors for the natural image distribution. Using the advances in deep learning, we benefit from fast and accurate neural networks to learn representations of image distribution. We build our idea based on the underlying theory of denoising autoencoders and how their solution relates to the natural image distribution.

A denoising autoencoder [VLBM08] is an autoencoder trained to reconstruct data that was corrupted with noise. Previously, Alain and Bengio [AB14] and Nguyen et al. [NYB+16] used DAEs to construct generative models. We build on the key observation by Alain and

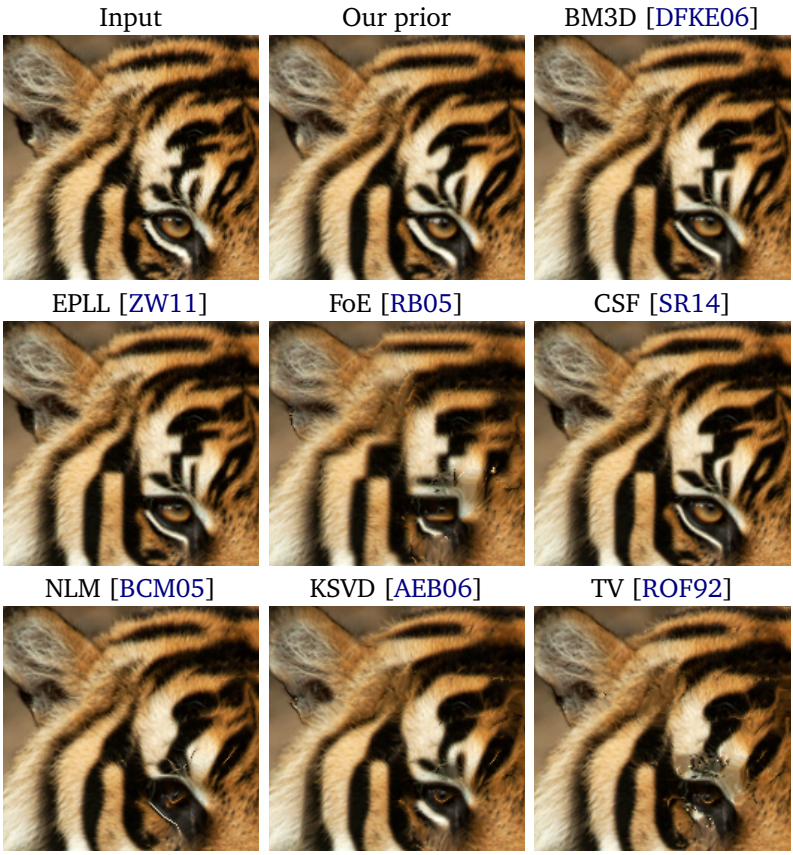


Figure 3.2: Visualization of image priors using the method by Shaham et al. [SM16]: Our deep mean-shift prior learns complex structures with different curvatures. Other priors prefer simpler structures like lines with small curvature or sharp corners.

Bengio [AB14] that for each input, the output of an optimal denoising autoencoder is a local mean of the true natural image density. The weight function that defines the local mean is equivalent to the noise distribution used to train the DAE. Our insight is that the autoencoder error, which is the difference between the output and input of the trained autoencoder, is a mean shift vector [CM02], and the noise distribution represents a mean shift kernel. It is worth mentioning that prior to the work of Alain and Bengio [AB14], others such as Miyasawa [Miy61], Raphan and Simoncelli [RS11], and Levin et al. [LN11], made similar observations about optimal denoisers. Specifically, the derivations below were, to the best of our knowledge, first discovered by Miyasawa [Miy61].

We visualize the intuition behind DAEs in Figure 3.3. Let us denote a DAE as r_σ. Given an input image x, its output is an image $r_\sigma(x)$. A DAE r_σ is trained to minimize [VLBM08]

$$\mathcal{L}_{\text{DAE}} = \mathbb{E}_{\eta,x}\left[\|x - r_\sigma(x+\eta)\|^2\right], \tag{3.5}$$

where the expectation is over all images x and Gaussian noise η with variance σ^2, and r_σ indicates that the DAE was trained with noise variance σ^2. It is important to note that the noise variance σ^2 here is not related to the degradation noise and its variance σ_n^2, and it is not a parameter to be learned. Instead, it is a user specified parameter whose role becomes clear with the following proposition. Let us denote the true data density of natural images as $p(x)$. Alain et al. [AB14] show that the output $r_\sigma(x)$ of the optimal DAE (assuming unlimited capacity) is related to the true data density $p(x)$ as

$$\begin{aligned} r_\sigma(x) &= \frac{\mathbb{E}_\eta\left[p(x-\eta)(x-\eta)\right]}{\mathbb{E}_\eta\left[p(x-\eta)\right]} \\ &= \frac{\int g_{\sigma^2}(\eta)p(x-\eta)(x-\eta)d\eta}{\int g_{\sigma^2}(\eta)p(x-\eta)d\eta}. \end{aligned} \tag{3.6}$$

This reveals an interesting connection to the mean shift algorithm [CM02]:

Proposition 3.1. *The autoencoder error, that is the difference between the output and the input of the autoencoder $r_\sigma(x) - x$ is an exact*

mean shift vector. More precisely, the mean shift vector (Comaniciu and Meer [CM02], Equation 17) is a Monte Carlo estimate of Equation 3.6 using random samples $\xi_i \sim p, i = 1 \dots n$.

Proof. By substituting $\xi = \eta$ in Equation 3.6, and Monte Carlo estimation of the integrals with a sum over n random samples $\xi_i \sim p, i = 1 \dots n$,

$$r_\sigma(x) = x - \frac{\sum_{i=1}^{n} g(x - \xi_i)(\xi_i)}{\sum_{i=1}^{n} g(x - \xi_i)},$$

we directly arrive at the original mean shift formulation (Comaniciu and Meer [CM02], Equation 17). □

The autoencoder output can be interpreted as a local mean or a weighted average of images in the neighborhood of x. The weights are given by the true density $p(x)$ multiplied by the noise distribution that was used during training, which is a local Gaussian kernel $g_\sigma(\eta)$ centered at x with variance σ^2. Hence the parameter σ^2 of the autoencoder determines the size of the region around x that contributes to the local mean. The key of our approach is the following theorem:

Theorem 3.1. *When the training noise* η *has a Gaussian distribution, the autoencoder error is proportional to the gradient of the log likelihood of the data density* p *smoothed by the Gaussian kernel* $g_{\sigma^2}(\eta)$*,*

$$r_\sigma(x) - x = \sigma^2 \nabla \log \left[g_\sigma * p \right](x), \tag{3.7}$$

where $*$ *means convolution.*

Proof. A proof of this theorem was derived earlier by Miyasawa [Miy61] and was generalized by Raphan and Simoncelli [RS11] for other types of noise. Here we describe an alternative proof that drives directly from the definition of the optimal DAE. We first rewrite the original equation for the optimal DAE (Alain and Bengio [AB14]

and our Equation 3.6) as

$$\begin{aligned} r_\sigma(x) &= \frac{\mathbb{E}_\eta\left[p(x-\eta)(x-\eta)\right]}{\mathbb{E}_\eta\left[p(x-\eta)\right]} \\ &= x - \frac{\mathbb{E}_\eta\left[p(x-\eta)\eta\right]}{\mathbb{E}_\eta\left[p(x-\eta)\right]}, \eta \sim \mathcal{N}(0,\sigma^2). \end{aligned}$$

By expanding the numerator in the quotient we get

$$\begin{aligned} \mathbb{E}_\eta\left[p(x-\eta)\eta\right] &= \int g_{\sigma^2}(\eta)p(x-\eta)\eta d\eta \\ &= -\sigma^2 \int \nabla g_{\sigma^2}(\eta)p(x-\eta)d\eta, \end{aligned}$$

where we used the definition of the derivative of the Gaussian to remove η inside the integral. Now we can use the Leibniz rule to interchange the ∇ operator with the integral and we get

$$\mathbb{E}_\eta\left[p(x-\eta)\eta\right] = -\sigma^2 \nabla \mathbb{E}_\eta\left[p(x-\eta)\right].$$

Plugging this back into our equation for the DAE we get

$$r_\sigma(x) = x + \sigma^2 \frac{\nabla \mathbb{E}_\eta\left[p(x-\eta)\right]}{\mathbb{E}_\eta\left[p(x-\eta)\right]},$$

and using the derivative of the logarithm we see that this is

$$\begin{aligned} r_\sigma(x) &= x + \sigma^2 \nabla \log \mathbb{E}_\eta\left[p(x-\eta)\right] \\ &= x + \sigma^2 \nabla \log[g_{\sigma^2} * p](x), \end{aligned}$$

as in Equation 3.7. □

With this alternative formulation of the DAEs we have removed the normalization term in the denominator of the DAE definition. This result shows that the autoencoder error (that is, the mean shift vector) corresponds to the gradient of the log-likelihood of the distribution blurred with a Gaussian kernel with variance σ^2. Hence we observe that the autoencoder error vanishes at stationary points, including local extrema, of the true density smoothed by the Gaussian kernel.

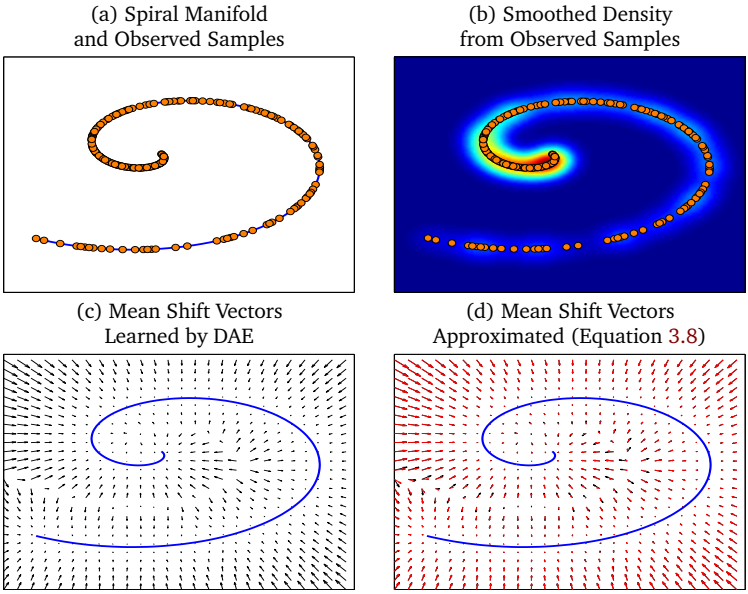


Figure 3.3: Visualization of a denoising autoencoder using a 2D spiral density. Given input samples of a true density (a), the autoencoder is trained to pull each sample corrupted by noise back to its original location. Adding noise to the input samples smooths the density represented by the samples (b). Assuming an infinite number of input samples and an autoencoder with unlimited capacity, for each input, the output of the optimal trained autoencoder is the local mean of the true density. The local weighting function corresponds to the noise distribution that was used during training, and it represents a mean shift kernel [CM02]. The difference between the output and the input of the autoencoder is a mean shift vector (c), which vanishes at local extrema of the true density smoothed by the mean shift kernel. Due to practical limitations, we approximate the mean shift vectors (d, red) using Equation 3.8. The difference between the true mean shift vectors (d, black) and our approximate vectors (d, red) vanishes as we get closer to the manifold.

Overcoming training limitations. The theory above assumes unlimited data and time to train an unlimited capacity autoencoder. In particular, to learn the true mean shift mapping, for each natural image the training data needs to include noise patterns that lead to other natural images. In practice, however, such patterns virtually never occur because of the high dimensionality. Since the DAE never observed natural images during training (produced by adding noise to other images), it overfits to noisy images. This is problematic during the gradient descent optimization, when the input to the DAE does not have noise.

As a workaround, we obtained better results by adding noise to the image before feeding it to the trained DAE during optimization. We further justify this by showing that with this workaround, we can still approximate a DAE that was trained with a desired noise variance σ^2 using the lowerbound

$$r_\sigma(x) - x \geq 2\Big(\mathbb{E}_\epsilon\left[r_{\frac{\sigma}{\sqrt{2}}}(x-\epsilon)\right] - x\Big), \tag{3.8}$$

where $\epsilon \sim \mathcal{N}(0, \frac{\sigma}{\sqrt{2}}^2)$, and $r_{\frac{\sigma}{\sqrt{2}}}$ is a DAE trained with noise standard deviation $\frac{\sigma}{\sqrt{2}}$. This is visualized in Figure 3.3(d). The red vectors indicate the approximated mean shift vectors using Equation 3.8 and the black vectors indicate the exact mean shift vectors. The approximation error decreases as we approach the true manifold.

Chapter 4

Autoencoding Priors

In this chapter we propose to leverage denoising autoencoder networks as priors to address image restoration problems. We build on the key observation that the output of an optimal denoising autoencoder is a local mean of the true data density, and the autoencoder error (the difference between the output and input of the trained autoencoder) is a mean shift vector. We use the magnitude of this mean shift vector, that is, the distance to the local mean, as the negative log likelihood of our natural image prior. For image restoration, we maximize the likelihood using gradient descent by backpropagating the autoencoder error. A key advantage of our approach is that we do not need to train separate networks for different image restoration tasks, such as non-blind deconvolution with different kernels, or super-resolution at different magnification factors. We demonstrate state of the art results for non-blind deconvolution and super-resolution using the same autoencoding prior.

We build on the key observations in Section 3.3 that for each input, the output of an optimal denoising autoencoder is a local mean of the true natural image density. The weight function that defines the local mean is equivalent to the noise distribution used to train the DAE. Our insight in Section 3.3 is that the autoencoder error, which is the

difference between the output and input of the trained autoencoder, is a mean shift vector [CM02], and the noise distribution represents a mean shift kernel.

Hence, we leverage neural DAEs in an elegant manner to define powerful image priors: Given the trained autoencoder, our natural image prior is based on the magnitude of the mean shift vector. For each image, the mean shift is proportional to the gradient of the true data distribution smoothed by the mean shift kernel, and its magnitude is the distance to the local mean in the distribution of natural images. With an optimal DAE, the energy of our prior vanishes exactly at the stationary points of the true data distribution smoothed by the mean shift kernel. This makes our prior attractive for maximum a posteriori (MAP) estimation.

For image restoration, we include a data term based on the known image degradation model. For each degraded input image, we maximize the likelihood of our solution using gradient descent by backpropagating the autoencoder error and computing the gradient of the data term. Intuitively, this means that our approach iteratively moves our solution closer to its local mean in the natural image density, while satisfying the data term. This is illustrated in Figure 4.1.

A key advantage of our approach is that we do not need to train separate networks for different image restoration tasks, such as non-blind deconvolution with different kernels, or super-resolution at different magnification factors. Even though our autoencoding prior is trained on a denoising problem, it is highly effective at removing these different degradations. We demonstrate state of the art results for non-blind deconvolution and super-resolution using the same autoencoding prior. In summary, the main contributions of this chapter are:

- An image restoration formulation based on maximum a posteriori (MAP) estimator that makes use of the connection between DAEs and mean shift, and the relation of an optimal DAE to the underlying data distribution.

- An implementation of the prior based on denoising autoen-

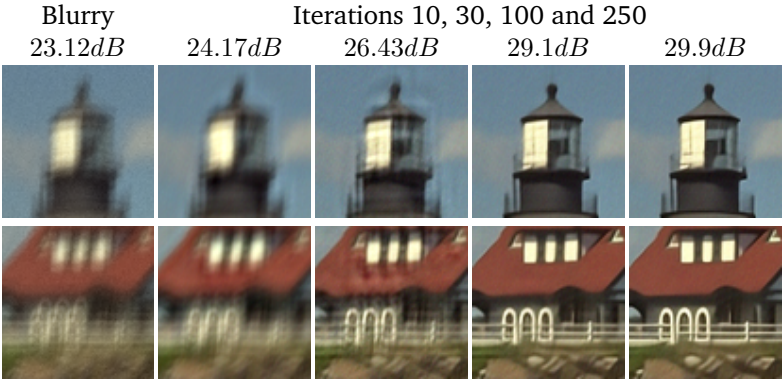


Figure 4.1: We propose a natural image prior based on a denoising autoencoder, and apply it to image restoration problems like non-blind deblurring. The output of an optimal denoising autoencoder is a local mean of the true natural image density, and the autoencoder error is a mean shift vector. We use the magnitude of the mean shift vector as the negative log likelihood of our prior. To restore an image from a known degradation, we use gradient descent to iteratively minimize the mean shift magnitude while respecting a data term. Hence, step-by-step we shift our solution closer to its local mean in the natural image distribution.

coders (DAEs) parametrized by a convolutional neural network.[1]

- Experiments to show the effectiveness of our prior for different restoration problems, including deblurring with arbitrary kernels and super-resolution with different magnification factors.

[1]The source code of the proposed method is available at https://github.com/siavashbigdeli/DAEP.

Figure 4.2: Local minimum of our natural image prior. Starting with a noisy image (left), we minimize the prior via gradient descent (middle: intermediate step) to reach the local minimum (right).

4.1 Prior Formulation

Following the problem formulation in Section 3.1, we propose a MAP estimator for image restoration. We follow the observations in Section 3.3 to use the squared magnitude of the mean shift vector as the energy (the negative log likelihood) of our prior, $L(x) = \|r_\sigma(x) - x\|^2$. This energy is very powerful because it tells us how close an image x is to its local mean $r_\sigma(x)$ in the true data density, and it vanishes at local extrema of the true density smoothed by the mean shift kernel. Figure 3.3(c), illustrates how small values of $L(x) = \|r_\sigma(x) - x\|^2$ occur close to the data manifold, as desired. Figure 4.2 visualizes a local minimum of our prior on natural images, which we find by iteratively minimizing the prior via gradient descent starting from a noisy input, without any help from a data term.

We use the logarithmic objective as in Equation 3.4, that leads to an energy minimization algorithm. Including the data term, we recover latent images as

$$\operatorname*{argmin}_x \|y - D(k * x)\|^2/\sigma_n^2 + \gamma\|r_\sigma(x) - x\|^2. \tag{4.1}$$

Our energy has two parameters that we will adjust based on the restoration problem. First, this is the mean shift kernel size σ, and

Algorithm 4.1 Proposed gradient descent. We express convolution as a matrix-vector product.

loop $\#iterations$

- Compute data term gradients $\nabla_x L(y|x)$:

$$K^T D^T (DKx - y)/\sigma_n^2$$

- Compute prior gradients $\nabla_x L(x)$:

$$\nabla_x r_\sigma(x)^T \Big(r_\sigma(x) - x \Big) + x - r_\sigma(x)$$

- Update x by descending

$$\nabla_x L(y|x) + \gamma \nabla_x L(x)$$

end loop

second we introduce a parameter γ to weight the relative influence of the data term and the prior.

4.1.1 Optimization

Given a trained autoencoder, we minimize our loss function in Equation 4.1 by applying gradient descent and computing the gradient of the prior using backpropagation through the autoencoder. Algorithm 4.1 shows the steps to minimize Equation 4.1. In the first step of each iteration, we compute the gradient of the data term with respect to image x. The second step is to find the gradients for our prior. The gradient of the mean shift vector $\|r_\sigma(x) - x\|^2$ requires the gradient of the autoencoder $r_\sigma(x)$, which we compute by backpropagation through the network. Finally, the image x is updated using the weighted sum of the two gradient terms.

4.1.2 Overcoming Training Limitations

Following our observations in Section 3.3, we obtained better results by adding noise to the image before feeding it to the trained DAE during optimization. Specifically, we can approximate DAE r_σ, by

another DAE $r_{\frac{\sigma}{\sqrt{2}}}$, that is

$$r_\sigma(x) - x \approx 2\Big(\mathbb{E}_\epsilon \left[r_{\frac{\sigma}{\sqrt{2}}}(x-\epsilon)\right] - x\Big), \tag{4.2}$$

where, again, $\epsilon \sim \mathcal{N}(0, \frac{\sigma^2}{2})$, and $r_{\frac{\sigma}{\sqrt{2}}}$ is a DAE trained with $\frac{\sigma}{\sqrt{2}}$ standard deviation noise.

To derive the above approximation, we start by using the alternative equation of the DAE from Equation 3.7 for r_{σ_τ}, where $\sigma_\tau \leq \sigma$, and write

$$r_{\sigma_\tau}(x) - x = \sigma_\tau^2 \nabla \log \mathbb{E}_\tau \left[p(x-\tau)\right],$$

and we take expectations of both sides over noise variable $\epsilon \sim \mathcal{N}(0, \sigma_\epsilon^2)$, where $\sigma_\epsilon^2 = \sigma^2 - \sigma_\tau^2$, that is

$$\mathbb{E}_\epsilon \left[r_{\sigma_\tau}(x-\epsilon)\right] - x = \sigma_\tau^2 \nabla \mathbb{E}_\epsilon \left[\log \mathbb{E}_\tau \left[p(x-\tau-\epsilon)\right]\right],$$

where we used the Leibniz rule to interchange the ∇ operator with the expectation. Now we would like to move the expectation over ϵ inside the $\log$. For this we perform a first order Taylor approximation of the $\log$ around $\mathbb{E}_\epsilon \left[\mathbb{E}_\tau \left[p(x-\tau-\epsilon)\right]\right]$ and replace the equality sign with approximation, which gives us

$$\mathbb{E}_\epsilon \left[r_{\sigma_\tau}(x-\epsilon)\right] - x \approx \sigma_\tau^2 \nabla \log \mathbb{E}_\epsilon \left[\mathbb{E}_\tau \left[p(x-\tau-\epsilon)\right]\right].$$

Now we use the fact that consecutive convolution of the density by Gaussian kernels with bandwidths σ_ϵ^2 and σ_τ^2 is identical to a single convolution by a Gaussian kernel with bandwidth $\sigma^2 = \sigma_\epsilon^2 + \sigma_\tau^2$, that is

$$\mathbb{E}_\epsilon \left[r_{\sigma_\tau}(x-\epsilon)\right] - x \approx \sigma_\tau^2 \nabla \log \mathbb{E}_\eta \left[p(x-\eta)\right],$$

where $\eta \sim \mathcal{N}(0, \sigma^2)$. We now use Equation 3.7 to rewrite this as

$$\mathbb{E}_\epsilon \left[r_{\sigma_\tau}(x-\epsilon)\right] - x \approx \frac{\sigma_\tau^2}{\sigma_\eta^2}\Big(r_\sigma(x) - x\Big),$$

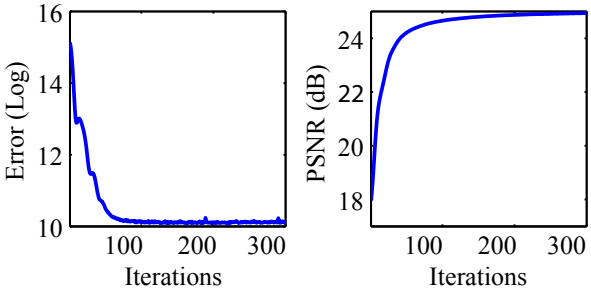


Figure 4.3: Convergence results of our stochastic objective error (left) and reconstruction PSNR (right) during the iterations.

which is the result we wanted. We use the specific case where $\sigma_\tau^2 = \sigma_\epsilon^2 = \frac{1}{2}\sigma^2$, which leads to Equation 4.2.

During optimization, we approximate the expected value in Equation 4.2 by stochastically sampling over ϵ. We use momentum of 0.9 and step size 0.1 in all experiments and we found that using one noise sample per iteration performs well enough to compute meaningful gradients. This approach resulted in a PSNR gain of around $1.7dB$ for the super-resolution task (Section 4.2.1), compared to evaluating the left hand side of Equation 4.2 directly.

Bad Local Minima and Convergence. The mean shift vector field learned by the DAE could vanish in low density regions [AB14], which corresponds to undesired local minima for our prior. In practice, however, we have not observed such degenerate solutions because our data term pulls the solution towards natural images. In all our experiments the optimization converges smoothly (Figure 4.1, intermediate steps), although we cannot give a theoretical guarantee. We show the convergence of our algorithm for a single image deblurring example in Figure 4.3. By using a momentum in our stochastic gradient descent, we are able to avoid oscillations and our reconstruction converges smoothly to the solution.

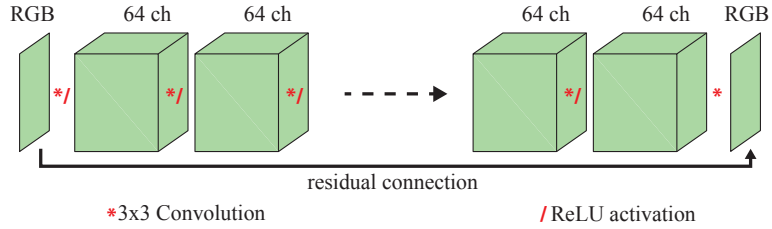


Figure 4.4: Our neural network consists of 20 convolution layers with 3×3 filters and ReLU activations in between.

4.1.3 Autoencoder Architecture and Training

Our network architecture is inspired by Zhang et al. [ZZC+16]. We visualize this architecture in Figure 4.4, where the network consists of 20 convolutional layers with batch normalization in between except for the first and last layers, and we use ReLU activations except for the last convolutional layer. The convolution kernels are of size 3×3 and the number of channels are 3 (RGB) for input and output and 64 for the rest of the layers. Unlike typical neural autoencoders, our network does not have a bottleneck. An explicit latent space implemented as a bottleneck is not required in principle for DAE training, and we do not need it for our application. We use a fully-convolutional network that allows us to compute the gradients with respect to the image more efficiently since the neuron activations are shared between many pixels. Our network is trained on color images of the ImageNet dataset [DDS+09] by adding Gaussian noise with standard deviation $\sigma_\epsilon = 25$ (around 10%). We perform residual learning by minimizing the L_2 distance of the output layer to the ground truth noise. We used the *Caffe* package [JSD+14] and employed an Adam solver [KB14] with $\beta_1 = 0.9$, $\beta_2 = 0.999$ and learning rate of 0.001, which we reduced during the iterations.

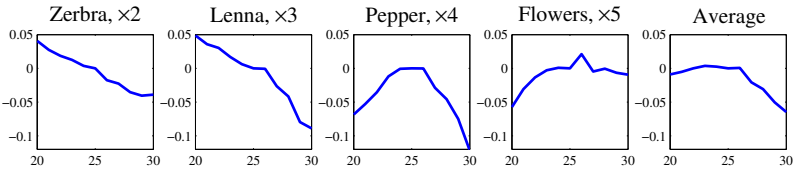


Figure 4.5: Performance gain for different DAE noise standard deviations $\sigma_\epsilon \in [20, 30]$ for supper-resolving different images from 'Set14' dataset [ZEP10]. The vertical axis shows the PSNR gain with respect to the PSNR score of the standard deviation $\sigma_\epsilon = 25$.

4.2 Experiments and Results

We compare our approach, Denoising Autoencoder Prior (DAEP), to state of the art methods in super-resolution and non-blind deconvolution problems. For all our experiments, we trained the autoencoder with $\sigma_\epsilon = 25$ ($\sigma = 25\sqrt{2}$), and the parameter of our energy (Equation 4.1) were set to $\gamma = 6.875/\sigma^2$. We always perform 300 gradient descent iteration steps during image restoration.

Optimal DAE noise variance. We visualize the influence of our DAE noise standard deviation parameter in Figure 4.5 for the super-resolution task. This experiment shows that the DAEs with bigger standard deviation perform consistently better for larger degradations. However in average, most small variations of the DAE standard deviation form $\sigma_\epsilon = 25$ result into negligible differences and do not change the overall performance (< 0.05 PSNR).

4.2.1 Super-Resolution

The super-resolution problem is usually defined in absence of noise ($\sigma_n = 0$), therefore we weight the prior by the inverse square root of the iteration number. This policy starts with a rough regularization and reduces the prior weight in each iteration, leading to solutions

that satisfy $\sigma_n = 0$. We compare our method to recent techniques by Kim et al. [KKLML16] (SRCNN), Dong et al. [DLHT16] (VDSR), Zhang et al. [ZZC+16] (DnCNN-3), Chen and Pock [CP16] (TNRD), and IRCNN by Zhang et al. [ZZGZ17]. SRCNN, VDSR and DnCNN-3 train an end-to-end network by minimizing the L_2 loss between the output of the network and the high-resolution ground truth, and TNRD uses a learned reaction diffusion model. While SRCNN and TNRD were trained separately for each scale, the VDSR and DnCNN-3 models were trained jointly on $\times 2, 3$ and 4 (DnCNN-3 training included also denoising and JPEG artifact removal tasks). For $\times 5$ super-resolution we used SRCNN and TNRD models that were trained on $\times 4$, and we used VDSR and DnCNN-3 models trained jointly on $\times 2, 3$ and 4. Table 5.4 compares the average PSNR of the super-resolved images from 'Set5' and 'Set14' datasets [BRGA12, ZEP10] for scale factors $\times 2, 3, 4,$ and 5, were we denote our method as DAEP. We compute PSNR values over cropped RGB images (where the crop size in pixels corresponds to the scale factor) for all methods. For SRCNN, however, we used a boundary of 13 pixels to provide full support for their network. While SRCNN, VDSR and DnCNN-3 solve directly for MMSE, our method solves for the MAP solution, which is not guaranteed to have better PSNR. Still, we achieve better results in average. For scale factor $\times 5$ our method performs significantly better since our prior does not need to be trained for a specific scale. Figure 4.6 shows visual comparisons to the super-resolution results from SRCNN [DLHT16], TNRD [CP16], and DnCNN-3 [ZZC+16] on three example images. We exclude results of VDSR due to limited space and visual similarity with DnCNN-3. Our natural image prior provides clean and sharp edges over all magnification factors.

4.2.2 Non-Blind Deconvolution

To evaluate and compare our method for non-blind deconvolution we used the dataset from Levin et al. [LFDF07] with four grayscale images and eight blur kernels in different sizes from 13×13 to 27×27. We compare our results to Levin et al. [LFDF07] (Levin), Zoran and

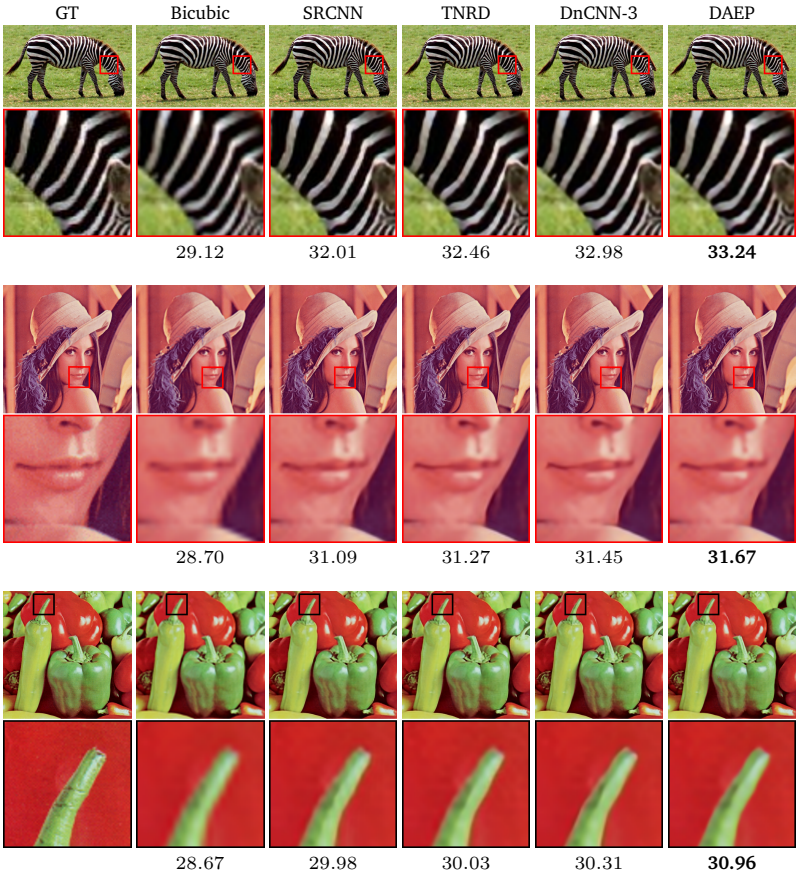


Figure 4.6: Comparison of super-resolution for scale factor 2 (top row), scale factor 3 (middle row), and scale factor 4 (bottom row) with the corresponding PSNR (dB) scores.

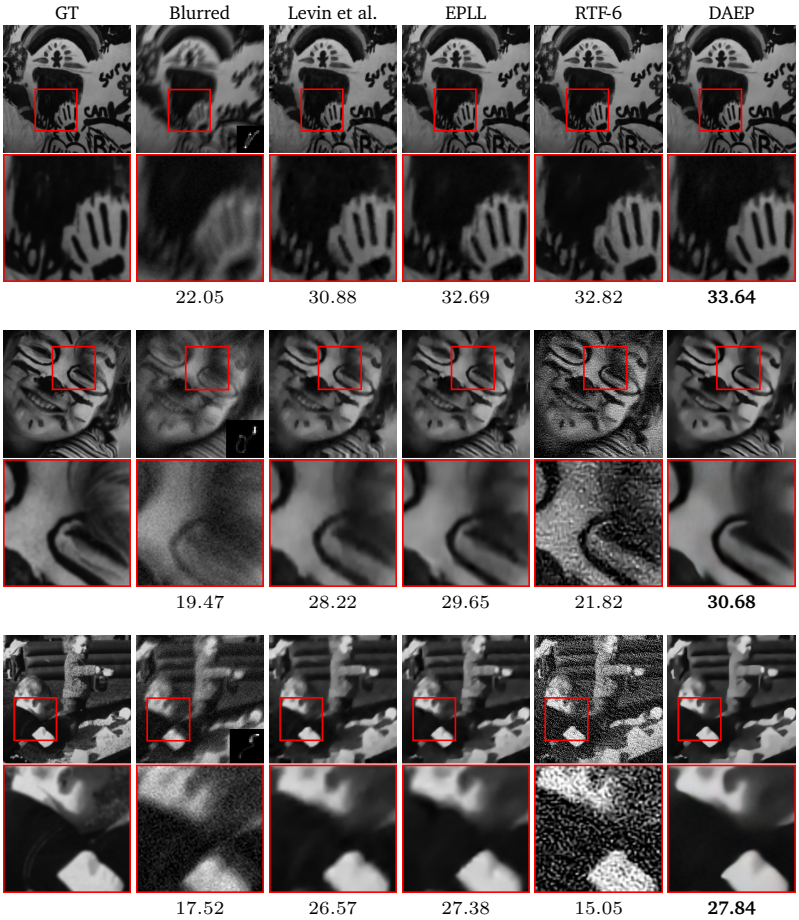


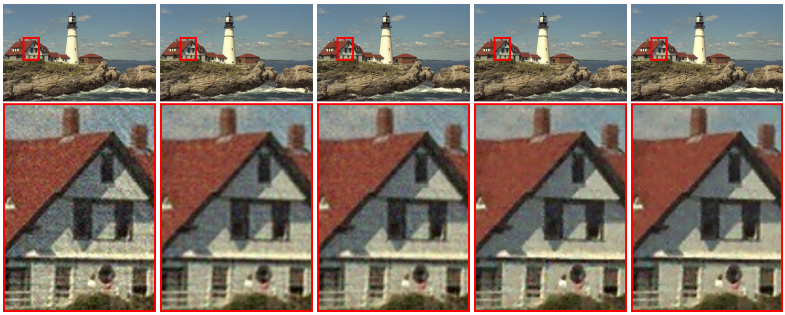
Figure 4.7: Comparison of non-blind deconvolution with additive noise standard deviation $\sigma = 2.55$ (top row), $\sigma = 7.65$ (middle row), and $\sigma = 12.75$ (bottom row) with the corresponding PSNR (dB) scores. The kernel is visualized in the bottom right of the blurred image.

Method	$\sigma \rightarrow$	2.55	7.65	12.75	time (sec)
Levin [LFDF07]		31.09	27.40	25.36	3.09
CSF [SR14]		29.35	27.05	25.50	0.72
EPLL [ZW11]		32.51	28.42	26.13	16.49
RTF-6 [SJN+16b]		32.51	21.44	16.03	9.82
IRCNN [ZZGZ17]		30.78	28.77	**27.41**	2.47
DAEP (Ours)		**32.69**	**28.95**	26.87	11.19

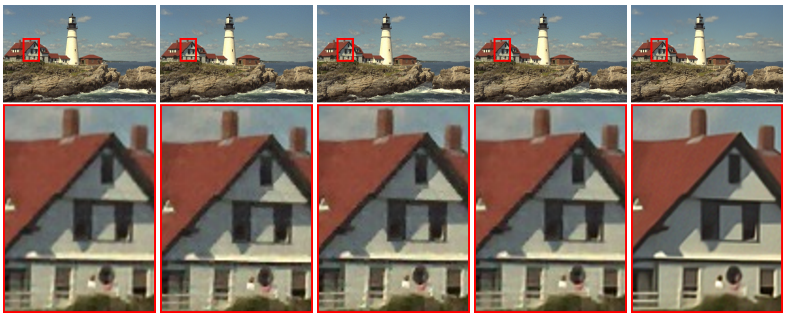
Table 4.1: Average PSNR (dB) for non-blind deconvolution on Levin et al.'s [LFDF07] dataset for different noise levels.

Weiss [ZW11] (EPLL), Schmidt et al. [SJN+16b] (RTF-6), and IRCNN by Zhang et al. [ZZGZ17] in Table 4.1, where we show the average PSNR of the deconvolution for three levels of additive noise ($\sigma \in \{2.55, 7.65, 12.75\}$). Note that RTF-6 [SJN+16b] is only trained for noise level $\sigma = 2.55$, therefore it does not perform well for other noise levels. Figure 4.7 provides visual comparisons for two deconvolution result images. Our natural image prior achieves higher PSNR and produces sharper edges and less visual artifacts compared to Levin et al. [LFDF07], Zoran and Weiss [ZW11], and Schmidt et al. [SJN+16b]. We report runtimes for different methods in Table 4.1 for image size of 128x128 on an Nvidia Titan X GPU. Our runtime is on par with popular methods such as EPLL [ZW11].

We performed an additional comparison on color images similar to Fortunato and Oliveira [FO14] using 24 color images from the Kodak Lossless True Color Image Suite from PhotoCD PCD0992 [Kod]. The images are blurred with a 19×19 blur kernel from Krishnan and Fergus [KF09b] and 1% noise is added. Figure 4.8 shows visual comparisons and average PSNRs over the whole dataset. Our method produces much sharper results and achieves a higher PSNR in average over this dataset. We refer to Section 5.3.1 for a more extensive evaluation of our method, denoted as DAEP.



(Lucy Richardson) 24.38/24.47 | [ZN09] 27.38/27.68 | [LFDF07] (L_2) 27.04/27.37 | [WYYZ08] (L_1) 27.68/28.23 | [WYYZ08] (TV) 28.63/29.25



[LFDF07] (IRLS) 28.96/30.15 | [SJA08] 28.97/30.01 | [KF09b] 29.15/30.18 | [FO14] 29.25/30.34 | DAEP (Ours) **29.92**/**31.07**

Figure 4.8: Comparison of non-blind deconvolution methods on the 21st image from the Kodak image set [Kod]. For each method, we report the PSNR (dB) of the visualized image (left) and the average PSNR on the whole set (right). The results of other methods were reproduced from Fortunato and Oliveira [FO14] for ease of comparison.

4.3 Discussion

We introduced a natural image prior based on denoising autoencoders (DAEs). Our prior minimizes the distances of restored images to their local means (the length of their mean shift vectors). This is powerful since mean shift vectors vanish at local extrema of the true density smoothed by the mean shift kernel. Our results demonstrate that a single DAE prior achieves state of the art results for non-blind image deblurring with arbitrary blur kernels and image super-resolution at different magnification factors.

A disadvantage of our approach is that it requires the solution of an optimization problem to restore each image. In contrast, end-to-end trained networks perform image restoration in a single feed-forward pass. For the increase in runtime computation, however, we gain much flexibility. With a single autoencoding prior, we obtain not only state of the art results for non-blind deblurring with arbitrary blur kernels and super-resolution with different magnification factors, but also successfully restore images corrupted by noise or holes as shown in Figure 4.9.

Our approach requires some user defined parameters (mean shift kernel size σ for DAE training and restoration, weight of the prior γ). While we use the same parameters for all experiments reported here, other applications may require to adjust these parameters. For example, we have experimented with image denoising (Figure 4.9), but so far we have not achieved state of the art results. We believe that this may require an adaptive kernel width for the DAE, and further fine-tuning of our parameters.

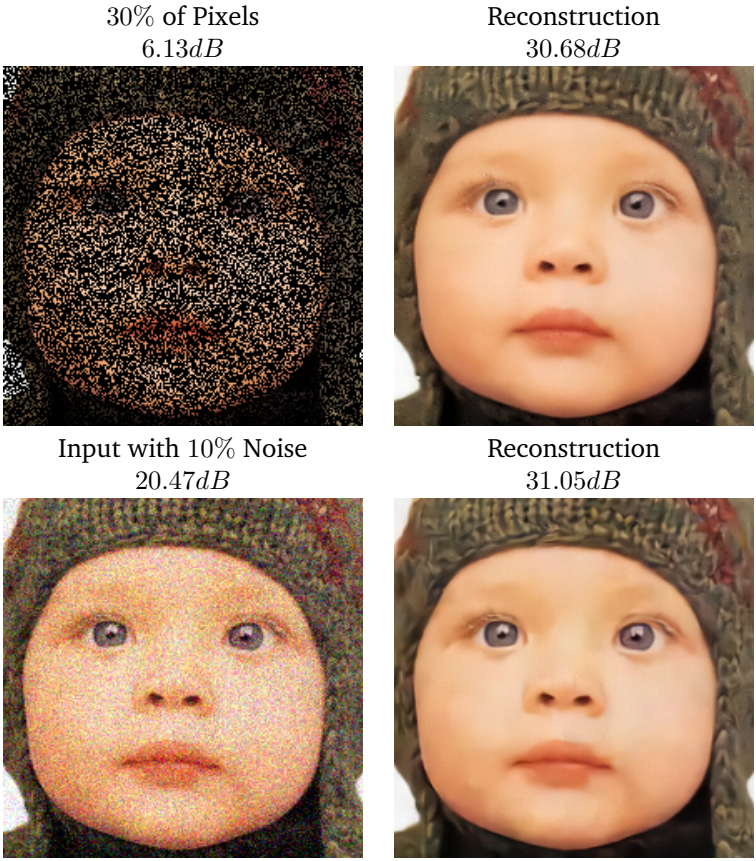


Figure 4.9: Restoration of images corrupted by noise and holes using the same autoencoding prior as in our other experiments.

Chapter 5

Deep Mean-Shift Priors

In Chapter 4 we described a generic prior that could be used in a MAP estimation of various restoration tasks. This approach requires the degradation model to be known when estimating the sharp image, which is not always the case. Additionally, the proposed autoencoding prior uses the mean-shift magnitude over the natural image distribution, which requires an expensive backpropagation step at each iteration of the optimization. In this chapter we introduce a natural image prior that directly represents a Gaussian-smoothed version of the natural image distribution. We include our prior in a formulation of image restoration as a Bayes estimator that also allows us to solve noise-blind image restoration problems. We show that the gradient of our prior corresponds to the mean-shift vector on the natural image distribution. In addition, we learn the mean-shift vector field using denoising autoencoders, and use it in a gradient descent approach to perform Bayes risk minimization. We demonstrate competitive results for noise-blind deblurring, super-resolution, and demosaicing. We propose an image prior that is directly based on an estimate of the natural image probability distribution. Although this seems like the most intuitive and straightforward idea to formulate a prior, only few previous techniques have taken this route [LN11]. Instead, most

priors are built on intuition or statistics of natural images (e.g., sparse gradients). Most previous deep learning priors are derived in the context of specific algorithms to solve the restoration problem, but it is not clear how these priors relate to the probability distribution of natural images. In contrast, our prior directly represents the natural image distribution smoothed with a Gaussian kernel, an approximation similar to using a Gaussian kernel density estimate. Note that we cannot hope to use the true image probability distribution itself as our prior, since we only have a finite set of samples from this distribution. We show a visual comparison in Figure 3.2, where our prior is able to capture the structure of the underlying image, but others tend to simplify the texture to straight lines and sharp edges.

We formulate image restoration as a Bayes estimator, and define a utility function that includes the smoothed natural image distribution. We approximate the estimator with a bound, and show that the gradient of the bound includes the gradient of the logarithm of our prior, that is, the Gaussian smoothed density. In addition, the gradient of the logarithm of the smoothed density is proportional to the mean-shift vector [CM02], and it has recently been shown that denoising autoencoders (DAEs) learn such a mean-shift vector field for a given set of data samples [AB14, BZ17]. Hence we call our prior a *deep mean-shift prior*, and our framework is an example of Bayesian inference using deep learning.

We demonstrate image restoration using our prior for noise-blind deblurring, super-resolution, and image demosaicing, where we solve Bayes estimation using a gradient descent approach. We achieve performance that is competitive with the state of the art for these applications. In summary, the main contributions of this chapter are:

- A formulation of image restoration as a Bayes estimator that leverages the Gaussian smoothed density of natural images as its prior. In addition, the formulation allows us to solve noise-blind restoration problems.

- An implementation of the prior, which we call *deep mean-shift prior*, that builds on denoising autoencoders (DAEs). We rely on

the observation that DAEs learn a mean-shift vector field, which is proportional to the gradient of the logarithm of the prior.

- Image restoration techniques based on gradient-descent risk minimization with competitive results for noise-blind image deblurring, super-resolution, and demosaicing.[1]

5.1 Bayesian Formulation

We assume a standard model for image degradation,

$$y = k * \xi + n, \quad n \sim \mathcal{N}(0, \sigma_n^2), \tag{5.1}$$

where ξ is the unknown image, k is the blur kernel, n is zero-mean Gaussian noise with variance σ_n^2, and y is the observed degraded image. For brevity, this model simplifies the degradation model introduced in Section 3.1 by removing the downsampling operator D. We restore an estimate x of the unknown image by defining and maximizing an objective consisting of a data term and an image likelihood,

$$\underset{x}{\operatorname{argmax}}\, \Phi(x) = \operatorname{data}(x) + \operatorname{prior}(x). \tag{5.2}$$

Our core contribution is to construct a prior that corresponds to the logarithm of the Gaussian-smoothed probability distribution of natural images. We will optimize the objective using gradient descent, and leverage the fact that we can learn the gradient of the prior using a denoising autoencoder (DAE). We next describe how we define our objective by formulating a Bayes estimator in Section 5.1.1, then explain how we leverage DAEs to obtain the gradient of our prior in Section 5.1.2, describe our gradient descent approach in Section 5.1.3, and finally our image restoration applications in Section 5.2.

[1] The source code of the proposed method is available at https://github.com/siavashbigdeli/DMSP.

5.1.1 Defining the Objective via a Bayes Estimator

A typical approach to solve the restoration problem is via a maximum a posteriori (MAP) estimate, where one considers the posterior distribution of the restored image $p(x|y) \propto p(y|x)p(x)$, derives an objective consisting of a sum of data and prior terms by taking the logarithm of the posterior, and maximizes it (minimizes the negative log-posterior, respectively). Instead, we will compute a Bayes estimator x for the restoration problem by maximizing the posterior expectation of a utility function,

$$E_{\tilde{x}}[G(\tilde{x}, x)] = \int G(\tilde{x}, x)p(y|\tilde{x})p(\tilde{x})d\tilde{x}, \tag{5.3}$$

where G denotes the utility function (e.g., a Gaussian), which encourages its two arguments to be similar. This is a generalization of MAP, where the utility is a Dirac impulse.

Ideally, we would like to use the true data distribution as the prior $p(\tilde{x})$. But we only have data samples, hence we cannot learn this exactly. Therefore, we introduce a smoothed data distribution

$$p'(x) = E_{\eta}[p(x + \eta)] = \int g_{\sigma}(\eta)p(x + \eta)d\eta, \tag{5.4}$$

where η has a Gaussian distribution with zero-mean and variance σ^2, which is represented by the smoothing kernel g_{σ}. The key idea here is that it is possible to estimate the smoothed distribution $p'(x)$ or its gradient from sample data. In particular, we will need the gradient of its logarithm, which we will learn using denoising autoencoders (DAEs). We now define our utility function as

$$G(\tilde{x}, x) = g_{\sigma}(\tilde{x} - x)\frac{p'(x)}{p(\tilde{x})}, \tag{5.5}$$

where we use the same Gaussian function g_{σ} with standard deviation σ as introduced for the smoothed distribution p'. This penalizes the estimate x if the latent parameter $\tilde{x}$ is far from it. In addition, the term $p'(x)/p(\tilde{x})$ penalizes the estimate if its smoothed density is lower

than the true density of the latent parameter. Unlike the utility in Jin et al. [JRF17], this approach will allow us to express the prior directly using the smoothed distribution p'.

By inserting our utility function into the posterior expected utility in Equation 5.3 we obtain

$$E_{\tilde{x}}[G(\tilde{x}, x)] = \int g_\sigma(\epsilon) p(y|x+\epsilon) \int g_\sigma(\eta) p(x+\eta) d\eta d\epsilon, \tag{5.6}$$

where the true density $p(\tilde{x})$ canceled out, as desired, and we introduced the variable substitution $\epsilon = \tilde{x} - x$.

We finally formulate our objective by taking the logarithm of the expected utility in Equation 5.6, and introducing a lower bound that will allow us to split Equation 5.6 into a data term and an image likelihood. By exploiting the concavity of the $\log$ function, we apply Jensen's inequality and get our objective $\Phi(x)$ as

$$\begin{aligned}
\log E_{\tilde{x}}[G(\tilde{x}, x)] &= \log \int g_\sigma(\epsilon) p(y|x+\epsilon) \int g_\sigma(\eta) p(x+\eta) d\eta d\epsilon \\
&\geq \int g_\sigma(\epsilon) \log \left[p(y|x+\epsilon) \int g_\sigma(\eta) p(x+\eta) d\eta \right] d\epsilon \\
&= \underbrace{\int g_\sigma(\epsilon) \log p(y|x+\epsilon) d\epsilon}_{\text{Data term } \mathrm{data}(x)} + \underbrace{\log \int g_\sigma(\eta) p(x+\eta) d\eta}_{\text{Image likelihood } \mathrm{prior}(x)}.
\end{aligned} \tag{5.7}$$

Image Likelihood. We denote the image likelihood as

$$\mathrm{prior}(x) = \log \int g_\sigma(\eta) p(x+\eta) d\eta. \tag{5.8}$$

The key observation here is that our prior expresses the image likelihood as the logarithm of the Gaussian-smoothed true natural image distribution $p(x)$, which is similar to a kernel density estimate.

Data Term. Given that the degradation noise is Gaussian, we see that [JRF17]

$$\begin{aligned}\text{data}(x) &= \int g_\sigma(\epsilon) \log p(y|x+\epsilon) d\epsilon \\ &= -\frac{|y - k * x|^2}{2\sigma_n^2} - M\frac{\sigma^2}{2\sigma_n^2}|k|^2 - N \log \sigma_n + \text{const}, \end{aligned} \tag{5.9}$$

where M and N denote the number of pixels in x and y respectively. This will allow us to address noise-blind problems as we will describe in detail in Section 5.2.

5.1.2 Gradient of the Prior via Denoising Autoencoders

A key insight of our approach is that we can effectively learn the gradients of our prior in Equation 5.8 using denoising autoencoders (DAEs). From our analysis in Section 3.3, specifically Equation 3.7, we can now see that the DAE error, that is, the difference $r_\sigma(x) - x$ between the output of the DAE and its input, is the gradient of the image likelihood in Equation 5.8. Hence, a main result of our approach is that we can write the gradient of our prior using the DAE error,

$$\nabla \text{prior}(x) = \nabla \log \int g_\sigma(\eta) p(x+\eta) d\eta = \frac{1}{\sigma^2}\Big(r_\sigma(x) - x\Big). \tag{5.10}$$

5.1.3 Stochastic Gradient Descent

We consider the optimization as minimization of the negative of our objective $\Phi(x)$ and refer to it as gradient descent. Similar to our previous observations in Section 4.1 and Section 3.3, we observed that the trained DAE is overfitted to noisy images. Because of the large gap in dimensionality between the embedding space and the natural image manifold, the vast majority of training inputs (noisy images) for the DAE lie at a distance very close to σ from the natural image manifold. Hence, the DAE cannot effectively learn mean-shift vectors

for locations that are closer than σ to the natural image manifold. In other words, our DAE does not produce meaningful results for input images that do not exhibit noise close to the DAE training σ.

To address this issue, we reformulate our prior to perform stochastic gradient descent steps that include noise sampling. We rewrite our prior from Equation 5.8 as

$$\begin{aligned}\text{prior}(x) &= \log \int g_\sigma(\eta) p(x+\eta) d\eta \\ &= \log \int g_{\sigma_2}(\eta_2) \int g_{\sigma_1}(\eta_1) p(x+\eta_1+\eta_2) d\eta_1 d\eta_2 \\ &\geq \int g_{\sigma_2}(\eta_2) \log \left[\int g_{\sigma_1}(\eta_1) p(x+\eta_1+\eta_2) d\eta_1 \right] d\eta_2 \\ &= \text{prior}_L(x), \end{aligned} \tag{5.11}$$

where $\sigma_1^2 + \sigma_2^2 = \sigma^2$, we used the fact that two Gaussian convolutions are equivalent to a single convolution with a Gaussian whose variance is the sum of the two, and we applied Jensen's inequality again. This leads to a new lower bound for the prior, which we call $\text{prior}_L(x)$. Note that the bound proposed by Jin et al. [JRF17] corresponds to the special case where $\sigma_1 = 0$ and $\sigma_2 = \sigma$.

We address our DAE overfitting issue by using the new lower bound $\text{prior}_L(x)$ with $\sigma_1 = \sigma_2 = \frac{\sigma}{\sqrt{2}}$. Its gradient is

$$\nabla \text{prior}_L(x) = \frac{2}{\sigma^2} \int g_{\frac{\sigma}{\sqrt{2}}}(\eta_2) \left(r_{\frac{\sigma}{\sqrt{2}}}(x+\eta_2) - (x+\eta_2) \right) d\eta_2. \tag{5.12}$$

In practice, computing the integral over η_2 is not possible at runtime. Instead, we approximate the integral with a single noise sample, which leads to the stochastic evaluation of the gradient of the prior as

$$\nabla \text{prior}_L^s(x) = \frac{2}{\sigma^2} \left(r_{\frac{\sigma}{\sqrt{2}}}(x+\eta_2) - x \right), \tag{5.13}$$

where $\eta_2 \sim \mathcal{N}(0, \sigma_2^2)$. This addresses the overfitting issue, since it means we add noise each time before we evaluate the DAE. Given

the stochastically sampled gradient of the prior, we apply a gradient descent approach with momentum that consists of the following steps:

$$\boxed{\begin{aligned} &\textbf{1. } u^t = -\nabla \,\mathrm{data}(x^{t-1}) - \nabla \,\mathrm{prior}_L^s(x^{t-1}) \\ &\textbf{2. } \bar{u} = \mu\bar{u} - \alpha u^t \\ &\textbf{3. } x^t = x^{t-1} + \bar{u} \end{aligned}} \tag{5.14}$$

where u^t is the update step for x at iteration t, $\bar{u}$ is the running step, and μ and α are the momentum and step-size.

5.2 Image Restoration using the Deep Mean-Shift Prior

We next describe the detailed gradient descent steps, including the derivatives of the data term, for different image restoration tasks. We provide a summary in Algorithm 5.1. For brevity, we omit the role of downsampling (required for super-resolution) and masking.

Non-Blind Deblurring (NB). The gradient descent steps for non-blind deblurring with a known kernel and degradation noise variance are given in Algorithm 5.1, top row (NB). Here K denotes the Toeplitz matrix of the blur kernel k.

Noise-Adaptive Deblurring (NA). When the degradation noise variance σ_n^2 is unknown, we can solve Equation 5.9 for the optimal σ_n^2 (since it is independent of the prior), which gives

$$\sigma_n^2 = \frac{1}{N}\left[|y - k * x|^2 + M\sigma^2|k|^2\right]. \tag{5.15}$$

Algorithm 5.1 Gradient descent steps for non-blind (NB), noise-blind (NA), and kernel-blind (KE) image deblurring. Kernel-blind deblurring involves the steps for (NA) and (KE) to update image and kernel.

Non-blind (NB):

1. $u^t = \frac{1}{\sigma_n^2} K^T (K x^{t-1} - y) - \nabla \text{prior}_L^s(x^{t-1})$

2. $\bar{u} = \mu \bar{u} - \alpha u^t$

3. $x^t = x^{t-1} + \bar{u}$

Noise-blind (NA):

1. $u^t = \lambda^t K^T (K x^{t-1} - y) - \nabla \text{prior}_L^s(x^{t-1})$

2. $\bar{u} = \mu \bar{u} - \alpha u^t$

3. $x^t = x^{t-1} + \bar{u}$

Kernel-blind (KE):

4. $v^t = \lambda^t \left[x^T (K^{t-1} x^{t-1} - y) + M \sigma^2 k^{t-1} \right]$

5. $\bar{v} = \mu_k \bar{v} - \alpha_k v^t$

6. $k^t = k^{t-1} + \bar{v}$

By plugging this back into the equation, we get the following data term

$$\text{data}(x) = -\frac{N}{2} \log \left[|y - k * x|^2 + M \sigma^2 |k|^2 \right], \tag{5.16}$$

which is independent of the degradation noise variance σ_n^2. We show the gradient descent steps in Algorithm 5.1, second row (NA), where $\lambda^t = N \left(|y - K x^{t-1}|^2 + M \sigma^2 |k|^2 \right)^{-1}$ adaptively scales the data term with respect to the prior.

Noise- and Kernel-Blind Deblurring (NA+KE). Gradient descent in noise-blind optimization includes an intuitive regularization for the kernel. We can use the objective in Equation 5.16 to jointly optimize for the unknown image and the unknown kernel. The gradient descent steps to update the image remain as in Algorithm 5.1, second

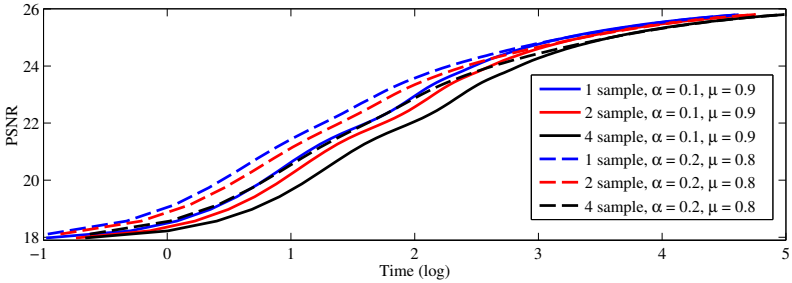


Figure 5.1: Effect of parameters in convergence of our method using the example image shown in Figure 5.2. We show the PSNR results for optimization using 1, 2, and 4 samples per iteration. For each optimization, we also compare the effect of increasing the stepsize.

row (NA), and we take additional steps to update the kernel estimate, as in Algorithm 5.1, third row (KE). Additionally, we project the kernel by applying $k^t = \max(k^t, 0)$ and $k^t = \frac{k^t}{|k^t|_1}$ after each step.

5.3 Experiments and Results

We use the same network and training procedure as in Section 4.1.3. The runtime of our method is linear in the number of pixels, and our implementation takes about 0.2 seconds per iteration for one megapixel on an Nvidia Titan X (Pascal). As mentioned in Section 5.1.3, we use a single noise sample to approximate our prior's gradient. Figure 5.1 compares cases where we use 1, 2, and 4 samples in each iteration to compete the gradients. These results indicate that increasing the number of samples only increases the runtime and does not improve the quality. Since including more samples better approximates the gradients, we extend the experiment by comparing the gradient descent stepsize. Taking larger stepsize leads to a faster convergence at the begging of the optimization, but leads to a slightly lower performance at the time or convergence. Therefore, for image

σ_n	DAE σ_1:	5	7.7	11	15
2.55 (NB)		25.71	**25.89**	25.69	25.17
5.10 (NB)		24.09	24.35	**24.45**	24.26
7.65 (NB)		23.16	23.42	**23.60**	23.59
10.2 (NB)		22.52	22.79	22.99	**23.06**
Avg. (NB)		23.87	24.11	**24.18**	24.02
2.55 (NA)		25.64	25.93	**26.00**	25.99
5.10 (NA)		24.03	24.34	**24.47**	24.47
7.65 (NA)		23.10	23.40	23.61	**23.63**
10.2 (NA)		22.45	22.77	22.97	**23.07**
Avg. (NA)		23.80	24.11	24.26	**24.29**
Avg.		23.84	24.11	**24.22**	24.15

Table 5.1: Evaluation of different noise standard deviation for DAE training. We train our DAE with different noise levels (scaled about twice in variance) and we show the average PSNR (dB) for non-blind deconvolution on the Berkeley [AMFM11] dataset.

restoration we always take 300 iterations with step length $\alpha = 0.1$ and momentum $\mu = 0.9$.

Optimal DAE noise variance. The main parameter of our framework is the noise level used in the DAE training. We experimented with different noise levels $\sigma_1 \in \{5, 7.7, 11, 15\}$ scaling about twice in variance, and found $\sigma_1 = 11$ to perform well for all our deblurring and super-resolution experiments. Table 5.1 shows the comparison results for non-blind deconvolution on the Berkeley [AMFM11] dataset. This results indicate a correlation between the degradation noise variance and the DAE training noise variance. In the Non-Blind (NB) optimization, the DAE with smaller training variance performed better for smaller amount of degradation noise. The Noise-Blind (NA) optimization scheme, however, is more robust to changes in the degradation noise variance compared to the non-blind case. The training noise

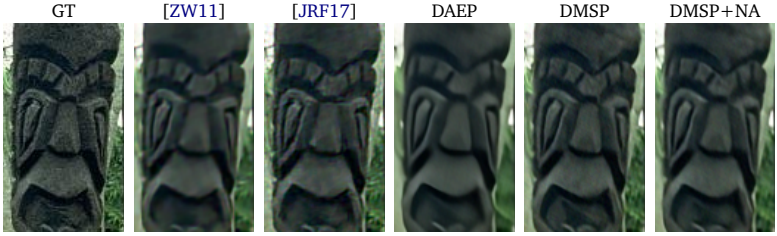


Figure 5.2: Visual comparison of our deconvolution results.

levels of $\{7.7, 11, 15\}$ performed similarly in average over the whole experiment, which means that the proposed method is insensitive to small variations of this parameter.

5.3.1 Deblurring: Non-Blind and Noise-Blind

In this section we evaluate our method for image deblurring using two datasets. Table 5.2 reports the average PSNR for 32 images from the Levin et al. [LFDF07] and 50 images from the Berkeley [AMFM11] segmentation dataset, where 10 images are randomly selected and blurred with 5 kernels as in Jin et al. [JRF17]. We highlight the best performing PSNR in bold and underline the second best value. The upper half of the table includes non-blind methods for deblurring. EPLL [ZW11] + NE uses a noise estimation step followed by non-blind deblurring. Noise-blind experiments are denoted by NA for noise adaptivity. We include our results for non-blind (DMSP) and noise-blind (DMSP + NA). Our noise adaptive approach consistently performs well in all experiments and on average we achieve better results than the state of the art. Figure 5.2 provides a visual comparison of our results. Our prior is able to produce sharp textures while also preserving the natural image structure.

	Levin [LFDF07]				Berkeley [AMFM11]			
Method	σ_n: 2.55	5.10	7.65	10.2	2.55	5.10	7.65	10.2
FD [KF09b]	30.03	28.40	27.32	26.52	24.44	23.24	22.64	22.07
EPLL [ZW11]	32.03	29.79	28.31	27.20	25.38	23.53	22.54	21.91
RTF-6 [SJN+16b]*	32.36	26.34	21.43	17.33	25.70	23.45	19.83	16.94
CSF [SR14]	29.85	28.13	27.28	26.70	24.73	23.61	22.88	22.44
DAEP (ours)	**32.64**	30.07	28.30	27.15	25.42	23.67	22.78	22.21
IRCNN [ZZGZ17]	30.86	29.85	28.83	28.05	25.60	24.24	23.42	22.91
EPLL [ZW11] + NE	31.86	29.77	28.28	27.16	25.36	23.53	22.55	21.90
EPLL [ZW11] + NA	32.16	**30.25**	28.96	27.85	25.57	23.90	22.91	22.27
TV-L2 + NA	31.05	29.14	28.03	27.16	24.61	23.65	22.90	22.34
GradNet 7S [JRF17]	31.43	28.88	27.55	26.96	25.57	24.23	23.46	22.94
DMSP (ours)	29.68	29.45	28.95	**28.29**	25.69	24.45	23.60	**22.99**
DMSP + NA (ours)	32.57	30.21	**29.00**	28.23	**26.00**	**24.47**	**23.61**	22.97

Table 5.2: Average PSNR (dB) for non-blind deconvolution on two datasets (*trained for $\sigma_n = 2.55$).

Method	$\sigma_n \rightarrow$	2.55	5.10	7.65	10.2
FD [KF09b]		30.79	28.90	27.86	27.14
EPLL [ZW11]		<u>32.05</u>	29.60	28.25	27.34
CSF [SR14]		30.88	28.60	27.65	26.97
TNRD [CP17]		30.03	28.79	-	-
IRCNN [ZZGZ17]		31.80	**30.13**	**28.93**	**28.09**
DAEP (ours)		31.76	29.31	28.01	27.16
EPLL [ZW11] + NE		32.02	29.60	28.25	27.34
EPLL [ZW11] + NA		**32.18**	<u>30.08</u>	<u>28.77</u>	27.81
TV-L2 + NA		30.07	28.59	27.60	26.89
GradNet 7S [JRF17]		31.75	29.31	28.04	27.54
DMSP (ours)		29.41	29.04	28.56	<u>27.97</u>
DMSP + NA (ours)		32.01	29.56	28.56	27.93

Table 5.3: Average PSNR (dB) for non-blind deconvolution on the Sun et al.'s [SCWH13] dataset.

5.3.2 Deblurring: Noise- and Kernel-Blind

We performed fully blind deconvolution with our method using Levin et al.'s [LFDF07] dataset. In this test, we performed 1000 gradient descent iterations. We used momentum $\mu = 0.7$ and step size $\alpha = 0.3$ for the unknown image and momentum $\mu_k = 0.995$ and step size $\alpha_k = 0.005$ for the unknown kernel. Figure 5.3 shows visual results of fully blind deblurring and performance comparison to state of the art (last column). We compare the SSD error ratio and the number of images in the dataset that achieves error ratios less than a threshold. Results for other methods are as reported by Perrone and Favaro [PF16]. Our method can reconstruct all the blurry images in the dataset with errors ratios less than 3.5. Note that our optimization performs end-to-end estimation of the final results and we do not use the common two stage blind deconvolution (kernel estimation, followed by non-blind deconvolution). Additionally our method uses a noise adaptive scheme where we do not assume knowledge of the input noise level.

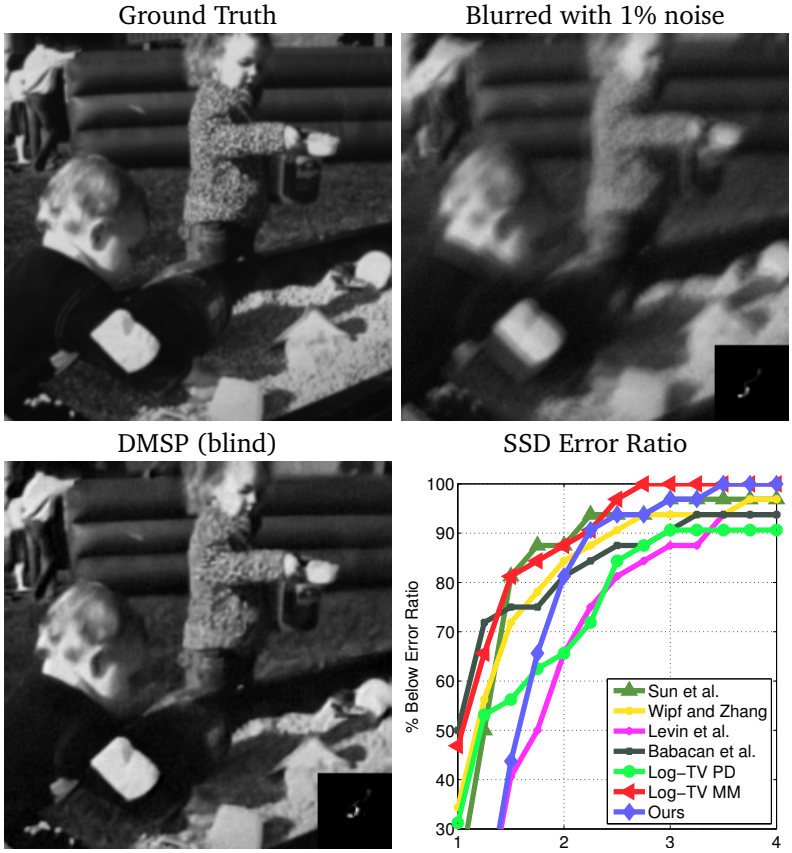


Figure 5.3: Performance of our method for fully (noise- and kernel-) blind deblurring on Levin's set.

Figure 5.4: Visual comparison for restoration from real camera noise and blur.

We also compare visual results for real camera noise and motion blur in Figure 5.4. Our noise- and kernel-blind optimization (NA + KE) is robust to real camera noise and non-uniform motion blur.

5.3.3 Super-resolution

To demonstrate the generality of our prior, we perform additional comparisons for the task of single image super-resolution. We evaluate our method (DMSP) on the two common datasets Set5 [BRGA12] and Set14 [ZEP10] for different upsampling scales. Since these tests do not include degradation noise ($\sigma_n = 0$), similar to Section 4.2.1, we perform our optimization with a rough weight for the prior and decrease it gradually to zero. We compare our method in Table 5.4. The upper half of the table represents methods that are specifically trained for super-resolution. SRCNN [DLHT16] and TNRD [CP17] have separate models trained for $\times 2, 3, 4$ scales, and we used the model for $\times 4$ to produce the $\times 5$ results. VDSR [KKLML16] and DnCNN-3 [ZZC+16] have a single model trained for $\times 2, 3, 4$ scales, which we also used

to produce $\times 5$ results. The lower half of the table represents general priors that are not designed specifically for super-resolution (see Section 4.2.1 for a more detail explanation). Our method performs on par with state of the art methods over all the upsampling scales.

5.3.4 Demosaicing

We finally performed a demosaicing experiment on the dataset introduced by Khashabi et al. [KNJF14]. This dataset is constructed by taking RAW images from a Panasonic camera, where the images are downsampled to construct the ground truth data. Due to the down sampling effect, in this evaluation we train a DAE with $\sigma_1 = 3$ noise standard deviation. The test dataset consists of 100 noisy images captured by a Panasonic camera using a Bayer color filter array (RGGB). We initialize our method with Matlab's demosaic function [MHC04]. To get even better initialization, we perform our initial optimization with a large degradation noise estimate ($\sigma_n = 2.5$) and then perform the optimization with a lower estimate ($\sigma_n = 1$). We summarize the quantitative results in Table 5.5. Our method is again on par with the state of the art. Additionally, our prior is not trained for a specific color filter array and therefore is not limited to a specific sub-pixel order. Figure 5.5 shows a qualitative comparison, where our method produces much smoother results compared to other methods.

5.4 Relationship to MAP

Here we show how our estimator relates to MAP and the formulation by Jin et al. [JRF17]. We start with the logarithm of the maximum a-posteriori (MAP) estimator and see that our proposed formulation is bounded from above by MAP. In addition, we observe that our

	Set5 [BRGA12]				Set14 [ZEP10]			
Method	$\times 2$	$\times 3$	$\times 4$	$\times 5$	$\times 2$	$\times 3$	$\times 4$	$\times 5$
Bicubic	31.80	28.67	26.73	25.32	28.53	25.92	24.44	23.46
SRCNN [DLHT16]	34.50	30.84	28.60	26.12	30.52	27.48	25.76	24.05
TNRD [CP17]	34.62	31.08	28.83	26.88	30.53	27.60	25.92	24.61
VDSR [KKLML16]	34.50	31.39	29.19	25.91	30.72	27.81	26.16	24.01
DnCNN-3 [ZZC+16]	35.20	**31.58**	**29.30**	26.30	30.99	27.93	**26.25**	24.26
IRCNN [ZZGZ17]	35.07	31.26	29.01	27.13	30.79	27.68	25.96	24.73
DAEP (ours)	**35.23**	31.44	29.01	27.19	**31.07**	**27.93**	26.13	24.88
DMSP (ours)	35.16	31.38	29.16	**27.38**	30.99	27.90	26.22	**25.01**

Table 5.4: Average PSNR (dB) for super-resolution on two datasets.

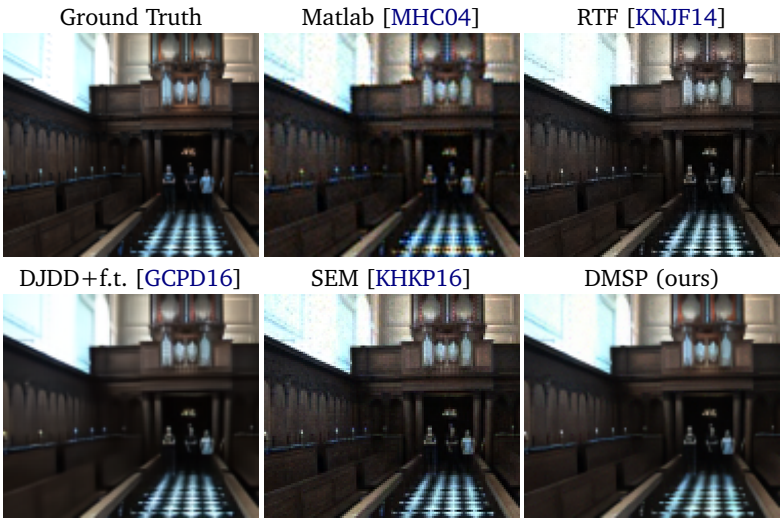


Figure 5.5: Visual comparison for demosaicing noisy images from the Panasonic dataset [KNJF14].

Matlab [MHC04]	RTF [KNJF14]	DJDD [GCPD16]
33.9	37.8	38.4
DJDD+f.t. [GCPD16]	SEM [KHKP16]	DMSP (ours)
38.6	38.8	38.7

Table 5.5: Average PSNR (dB) in linear RGB space for demosaicing on the Panasonic dataset [KNJF14].

formulation is bounded from below by Jin et al. [JRF17],

$$
\begin{aligned}
&\log \max_{\hat{x}} p(y|\hat{x})p(\hat{x}) \quad \text{(MAP)} \\
&= \max_{x} \log \max_{\hat{x}} p(y|\hat{x})p(\hat{x}) \int g_\sigma(x-\bar{x}_1)d\bar{x}_1 \int g_\sigma(x-\bar{x}_2)d\bar{x}_2 \\
&\geq \max_{x} \log \int g_\sigma(x-\bar{x}_1)p(y|\bar{x}_1)d\bar{x}_1 \int g_\sigma(x-\bar{x}_2)p(\bar{x}_2)d\bar{x}_2 \\
&\geq \max_{x} \underbrace{\int g_\sigma(x-\bar{x}_1)\log p(y|\bar{x}_1)d\bar{x}_1 + \log \int g_\sigma(x-\bar{x}_2)p(\bar{x}_2)d\bar{x}_2}_{\text{(Our lower bound, Equation 5.7)}} \\
&\geq \max_{x} \underbrace{\int g_\sigma(x-\bar{x}_1)\log p(y|\bar{x}_1)d\bar{x}_1 + \int g_\sigma(x-\bar{x}_2)\log p(\bar{x}_2)d\bar{x}_2}_{\text{Jin et al. [JRF17]}},
\end{aligned}
$$

where we applied Jensen's inequality several times. The interesting observation here is that our formulation is produced by separately relaxing the posteriori and prior, which later allows us to get a tighter lower bound for MAP compared to Jin et al. [JRF17].

5.5 Ratio between Runtime Noise and Training Noise

We perform an evaluation to find the best ratio between the runtime additive noise during stochastic gradient descent and the noise used during DAE training. Specifically, we set up an experiment for image deconvolution with our framework for fixed $\sigma = 15$ and compare the performance for different ratios between σ_1 and σ_2 (Equation 5.11). First, we train different DAEs with noise levels $\sigma_1 = 1 : 15$. Second, for each DAE we compute the variance of runtime additive noise by setting it to $\sigma_2^2 = \sigma^2 - \sigma_1^2$. And finally, we evaluate the performance of each configuration with our experiment. Figure 5.6 shows the quantitative performance for each ratio $\frac{\sigma_1^2}{\sigma^2}$. The configuration $\sigma_1^2 =$

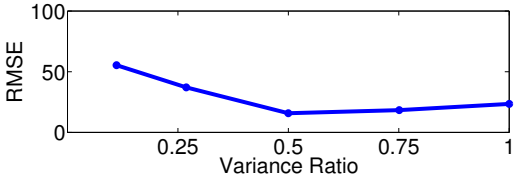


Figure 5.6: Performance comparison for different additive noise variances in our stochastic gradient descent method. We show the average RMSE for deblurring over a set of images for each ratio σ_1^2/σ^2. As expected, the desired variance σ^2 should be evenly split over the trained DAE and additive noise during stochastic gradient descent, that is $\sigma_1^2/\sigma^2 = \sigma_2^2/\sigma^2 = 0.5$.

$\sigma_2^2 = \frac{\sigma^2}{2}$ achieves the best performance. This is expected since our the DAE trained with a noise variance σ_1^2 performs better for that specific noise variance, therefore it is better to use the same variance for runtime additive noise.

5.6 Discussion

We proposed a Bayesian deep learning framework for image restoration with a generic image prior that directly represents the Gaussian smoothed natural image probability distribution. We showed that we can compute the gradient of our prior efficiently using a trained denoising autoencoder (DAE). Our formulation allows us to learn a single prior and use it for many image restoration tasks, such as noise-blind deblurring, super-resolution, and image demosaicing. Our results indicate that we achieve performance that is competitive with the state of the art for these applications. In the future, we would like to explore generalizing from Gaussian smoothing of the underlying distribution to other types of kernels. Similarly, one could also investigate other utility functions. We are also considering multi-scale optimization where one would reduce the Bayes utility support grad-

ually to get a tighter bound with respect to maximum a posteriori. Finally, our approach is not limited to image restoration and could be exploited to address other inverse problems.

Chapter 6

Conclusions

In this thesis, we addressed restoration techniques for inverse problems, including disparity map estimation and image restoration. We took a declarative approach to solve these problems by separating our prior knowledge from our reasoning. This approach enabled us to use the developed knowledge in various restoration tasks.

We described a disparity map estimation technique in Chapter 2. Common observations were used to design new prior constraints to estimate disparity maps. These constraints were formulated in a graphical model, which could then be solved efficiently using the filter-based parallel mean-shift approximation method. Since parallel updates are not guaranteed to converge, we developed a novel and efficient technique to perform sequential (as opposed to parallel) update scheme for inference. The time complexity of the proposed approach is still linear in the number of variables, just like in the parallel scheme. In contrast to the parallel scheme, where all variables are updated at the same time, the sequential scheme introduces a risk of ordering bias in the variable updates. In our implementation, we significantly reduce this bias by using a multi-directional optimization technique. We compared the parallel and the sequential update schemes using simulated and real data and we observed that by using an initialization step,

parallel updates can perform more efficiently in terms of quality and speed. We extended our disparity map estimation approach to stereo video sequences by including time-domain constraints. Using this extension, our technique produced disparity maps that are temporally coherent and have significantly less flickering artifacts compared to other state of the art methods.

Our general sequential algorithm and the proposed temporal constraints can be used and extended for many other labeling problems, such as semantic and motion segmentation tasks. When encountering such problems, the lack of a general-purpose dataset prevents us from developing data-driven approaches. Therefore, hand-crafted constraints are more practical and intuitive to use. However, more complex restoration problems such as image restoration cannot be solved efficiently using hand-crafted priors and require data-driven techniques.

In Chapter 3, we described the general image restoration task using a standard degradation model. We proposed an intuitive formulation of the prior distribution using Denoising Autoencoders (DAEs). We showed that an optimal DAE can be used to compute the gradient of a smoothed version of the natural image distribution.

We use this analysis in Chapters 4 and 5 to formulate generic image restoration techniques. In practice, we used neural networks as an efficient parametrization of DAEs in our techniques. Our method in Chapter 4, called *Denoising Autoencoder Prior (DAEP)*, is built on the observation that the gradient of the smoothed density has its minimum length in the local extrema of the natural image distribution. Therefore, we defined an energy minimization objective to minimize the magnitude of this gradient. This led to a generic approach that could be used for various image restoration tasks such as deblurring and super resolution.

In a more generic approach, we propose our *Deep Mean-Shift Prior (DMSP)* method for image restoration problems. We built this framework using a Bayesian formulation that relaxes the conventional maximum a-posteriori estimators. This relaxation led to a simple objective function that incorporates the smoothed version of the natural image

density. We used our previous analysis to compute the gradient of our objective via DAEs efficiently and perform different restoration tasks. We showed that this general framework can be used also when degradation parameters such as noise variance and blur kernel are unknown. Our evaluations implicates that, using this single framework, we can achieve state of the art performance in many tasks such as deblurring and demosaicing.

A main limitation of the proposed methods is the use of fully-convolutional neural networks to parametrize the DAEs. In practice, this parametrization only works for small degradations and cannot handle more global artifacts. For example, our primary results on the image reflection removal task shows that the trained DAEs are unable to regularize the low frequencies in images. An intuitive approach to resolve this issue could be to use a multi-scale regularization, where one would train and use different DAEs for different scales of images. Despite its limitations, the proposed approach of learning natural priors using DAEs can be used for many other types of data, where a large set of samples are available for training such as medical images.

Bibliography

[AB14] Guillaume Alain and Yoshua Bengio. What regularized auto-encoders learn from the data-generating distribution. *Journal of Machine Learning Research*, 15:3743–3773, 2014.

[AEB06] Michal Aharon, Michael Elad, and Alfred Bruckstein. rmk-svd: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Transactions on signal processing*, 54(11):4311–4322, 2006.

[AMFM11] Pablo Arbelaez, Michael Maire, Charless Fowlkes, and Jitendra Malik. Contour detection and hierarchical image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(5):898–916, 2011.

[BBZ16] Siavash Arjomand Bigdeli, Gregor Budweiser, and Matthias Zwicker. Temporally coherent disparity maps using crfs with fast 4d filtering. *IPSJ Transactions on Computer Vision and Applications*, 8(1):10, 2016.

[BCM05] Antoni Buades, Bartomeu Coll, and J-M Morel. A non-local algorithm for image denoising. In *Computer Vision and Pattern Recognition (CVPR), 2005 IEEE Conference on*, volume 2, pages 60–65. IEEE, 2005.

[BRGA12] Marco Bevilacqua, Aline Roumy, Christine Guillemot, and Marie-Line Alberi-Morel. Low-complexity single-image super-resolution based on nonnegative neighbor embedding. In *British Machine Vision Conference, BMVC 2012, Surrey, UK, September 3-7, 2012*, pages 1–10, 2012.

[BSH12] Harold C Burger, Christian J Schuler, and Stefan Harmeling. Image denoising: Can plain neural networks compete with bm3d? In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2392–2399. IEEE, 2012.

[BT99] Stan Birchfield and Carlo Tomasi. Depth discontinuities by pixel-to-pixel stereo. *International Journal of Computer Vision*, 35(3):269–293, 1999.

[BVZ01] Yuri Boykov, Olga Veksler, and Ramin Zabih. Fast approximate energy minimization via graph cuts. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 23(11):1222–1239, 2001.

[BZ17] Siavash Arjomand Bigdeli and Matthias Zwicker. Image restoration using autoencoding priors. *arXiv preprint arXiv:1703.09964*, 2017.

[BZFJ17] Siavash Arjomand Bigdeli, Matthias Zwicker, Paolo Favaro, and Meiguang Jin. Deep mean-shift priors for image restoration. In *Advances in Neural Information Processing Systems*, pages 763–772, 2017.

[CLP+17] JH Chang, Chun-Liang Li, Barnabas Poczos, BVK Kumar, and Aswin C Sankaranarayanan. One network to solve them all—solving linear inverse problems using deep projection models. *arXiv preprint arXiv:1703.09912*, 2017.

[CM02] Dorin Comaniciu and Peter Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(5):603–619, 2002.

[CP16] Yunjin Chen and Thomas Pock. Trainable nonlinear reaction diffusion: A flexible framework for fast and effective image restoration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2016.

[CP17] Yunjin Chen and Thomas Pock. Trainable nonlinear reaction diffusion: A flexible framework for fast and effective image restoration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(6):1256–1272, 2017.

[CXGZ15] Ayan Chakrabarti, Ying Xiong, Steven J Gortler, and Todd Zickler. Low-level vision by consensus in a spatial hierarchy of regions. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4009–4017. IEEE, 2015.

[DBRZ14] Daniel Donatsch, Siavash Arjomand Bigdeli, Philippe Robert, and Matthias Zwicker. Hand-held 3d light field photography and applications. *The Visual Computer*, 30(6-8):897–907, 2014.

[DDS+09] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition (CVPR), 2009 IEEE Conference on*, pages 248–255. IEEE, 2009.

[DFKE06] Kostadin Dabov, Alessandro Foi, Vladimir Katkovnik, and Karen Egiazarian. Image denoising with block-matching and 3d filtering. In *Electronic Imaging 2006*, pages 606414–606414. International Society for Optics and Photonics, 2006.

[DHAH14] Amnon Drory, Carsten Haubold, Shai Avidan, and Fred A Hamprecht. Semi-global matching: a principled derivation in terms of message passing. In *German Conference on Pattern Recognition*, pages 43–53. Springer, 2014.

[DHMS00] David DiLaura, Kevin Houser, Richard Mistrick, and Gary Steffy. *TheÂăIESNAÂăLightingÂăHandbook:ÂăÂăReferenceÂă&ÂăApplication*. Illuminating Engineering Society of North America, New York, 10 edition, 2000.

[DLHT14] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Learning a deep convolutional network for image super-resolution. In *European Conference on Computer Vision*, pages 184–199. Springer, 2014.

[DLHT16] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Image super-resolution using deep convolutional networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(2):295–307, 2016.

[Eco10] Economist. Camera-phones, dotty but dashing. http://www.economist.com/node/15865270, April 2010. Accessed: 11-01-2018.

[Fat07] Raanan Fattal. Image upsampling via imposed edge statistics. *ACM Trans. Graph.*, 26(3), July 2007.

[FO14] Horacio E. Fortunato and Manuel M. Oliveira. Fast high-quality non-blind deconvolution using sparse adaptive priors. *The Visual Computer*, 30(6-8):661–671, 2014.

[FSH+06] Rob Fergus, Barun Singh, Aaron Hertzmann, Sam T Roweis, and William T Freeman. Removing camera shake from a single photograph. In *ACM Transactions on Graphics (TOG)*, volume 25, pages 787–794. ACM, 2006.

[GCPD16] Michaël Gharbi, Gaurav Chaurasia, Sylvain Paris, and Frédo Durand. Deep joint demosaicking and denoising. *ACM Transactions on Graphics (TOG)*, 35(6):191, 2016.

[GG15] Fatma Guney and Andreas Geiger. Displets: Resolving stereo ambiguities using object knowledge. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4165–4175, 2015.

[GLU12] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 3354–3361. IEEE, 2012.

[GO11] Eduardo SL Gastal and Manuel M Oliveira. Domain transform for edge-aware image and video processing. In *ACM Transactions on Graphics (TOG)*, volume 30, page 69. ACM, 2011.

[GO15] Eduardo SL Gastal and Manuel M Oliveira. High-order recursive filtering of non-uniformly sampled signals for image and video processing. In *Computer Graphics Forum*, volume 34, pages 81–93. Wiley Online Library, 2015.

[GPAM+14] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 2672–2680. Curran Associates, Inc., 2014.

[GZX+15] Shuhang Gu, Wangmeng Zuo, Qi Xie, Deyu Meng, Xiangchu Feng, and Lei Zhang. Convolutional sparse coding for image super-resolution. In *Computer Vision and*

Pattern Recognition (CVPR), 2015 IEEE Conference on, pages 1823–1831. IEEE, 2015.

[HBGR09] Asmaa Hosni, Michael Bleyer, Margrit Gelautz, and Christoph Rhemann. Local stereo matching using geodesic support weights. In *2009 16th IEEE International Conference on Image Processing (ICIP)*, pages 2093–2096. IEEE, 2009.

[Hir08] Heiko Hirschmuller. Stereo processing by semiglobal matching and mutual information. *IEEE Trans. PAMI*, 30(2):328–341, 2008.

[HK12] Simon Hermann and Reinhard Klette. Iterative semiglobal matching for robust driver assistance systems. In *Asian Conference on Computer Vision*, pages 465–478. Springer, 2012.

[JAFF16] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European Conference on Computer Vision*, pages 694–711. Springer, 2016.

[JJ98] Tommi S Jaakkola and Michael I Jordan. Improving the mean field approximation via the use of mixture distributions. In *Learning in graphical models*, pages 163–173. Springer, 1998.

[JRF17] M. Jin, S. Roth, and P. Favaro. Noise-blind image deblurring. In *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*. IEEE, 2017.

[JSD+14] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the 22nd ACM international conference on Multimedia*, pages 675–678. ACM, 2014.

[JZSK09] Neel Joshi, C Lawrence Zitnick, Richard Szeliski, and David J Kriegman. Image deblurring and denoising using color priors. In *Computer Vision and Pattern Recognition (CVPR), 2009 IEEE Conference on*, pages 1550–1557. IEEE, 2009.

[KB14] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[KF09a] Daphne Koller and Nir Friedman. *Probabilistic graphical models: principles and techniques*. MIT press, 2009.

[KF09b] Dilip Krishnan and Rob Fergus. Fast image deconvolution using hyper-laplacian priors. In *Advances in Neural Information Processing Systems*, pages 1033–1041, 2009.

[KHKP16] Teresa Klatzer, Kerstin Hammernik, Patrick Knobelreiter, and Thomas Pock. Learning joint demosaicing and denoising based on sequential energy minimization. In *Computational Photography (ICCP), 2016 IEEE International Conference on*, pages 1–11. IEEE, 2016.

[KK11] Philipp Krähenbühl and Vladlen Koltun. Efficient inference in fully connected CRFs with Gaussian edge potentials. In *Proc. NIPS*, pages 109–117, 2011.

[KKLML16] Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee. Accurate image super-resolution using very deep convolutional networks. In *Computer Vision and Pattern Recognition (CVPR), 2016 IEEE Conference on*, pages 1646–1654. IEEE, 2016.

[KNJF14] Daniel Khashabi, Sebastian Nowozin, Jeremy Jancsary, and Andrew W Fitzgibbon. Joint demosaicing and denoising via learned nonparametric random fields. *IEEE Transactions on Image Processing*, 23(12):4968–4981, 2014.

[Kod] Kodak. Kodak lossless true color image suite. http://r0k.us/graphics/kodak/. Accessed: 27-1-2013.

[Kol06] Vladimir Kolmogorov. Convergent tree-reweighted message passing for energy minimization. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 28(10):1568–1583, 2006.

[KW14] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. In *ICLR 2014*, 2014.

[LeC98] Yann LeCun. The mnist database of handwritten digits. *http://yann. lecun. com/exdb/mnist/*, 1998.

[LFDF07] Anat Levin, Rob Fergus, Frédo Durand, and William T Freeman. Image and depth from a conventional camera with a coded aperture. *ACM Transactions on Graphics (TOG)*, 26(3):70, 2007.

[LN11] Anat Levin and Boaz Nadler. Natural image denoising: Optimality and inherent bounds. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 2833–2840. IEEE, 2011.

[LNDF12] Anat Levin, Boaz Nadler, Fredo Durand, and William T Freeman. Patch complexity, finite pixel correlations and optimal denoising. In *European Conference on Computer Vision*, pages 73–86. Springer, 2012.

[LTH+16] Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, et al. Photo-realistic single image super-resolution using a generative adversarial network. *arXiv preprint arXiv:1609.04802*, 2016.

[LWA+12] Manuel Lang, Oliver Wang, Tunc Aydin, Aljoscha Smolic, and Markus H Gross. Practical temporal consistency for

image-based graphics applications. *ACM Trans. Graph.*, 31(4):34, 2012.

[LWW+16] Ding Liu, Zhaowen Wang, Bihan Wen, Jianchao Yang, Wei Han, and Thomas S Huang. Robust single image super-resolution via deep networks with sparse prior. *IEEE Transactions on Image Processing*, 25(7):3194–3207, 2016.

[MG15] Moritz Menze and Andreas Geiger. Object scene flow for autonomous vehicles. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3061–3070, 2015.

[MHC04] Henrique S Malvar, Li-wei He, and Ross Cutler. High-quality linear interpolation for demosaicing of bayer-patterned color images. In *Acoustics, Speech, and Signal Processing, 2004. Proceedings.(ICASSP'04). IEEE International Conference on*, volume 3, pages iii–485. IEEE, 2004.

[Miy61] KOICHI Miyasawa. An empirical bayes estimator of the mean of a normal population. *Bull. Inst. Internat. Statist*, 38(181-188):1–2, 1961.

[MLD12] Dongbo Min, Jiangbo Lu, and Minh N Do. Depth video enhancement based on weighted mode filtering. *IEEE Trans. Imag. Proc.*, 21(3):1176–1190, 2012.

[MMHC17] Tim Meinhardt, Michael Möller, Caner Hazirbas, and Daniel Cremers. Learning proximal operators: Using denoising networks for regularizing inverse imaging problems. *arXiv preprint arXiv:1704.03488*, 2017.

[MSY16] Xiao-Jiao Mao, Chunhua Shen, and Yu-Bin Yang. Image restoration using very deep convolutional encoder-decoder networks with symmetric skip connections. In *Advances in Neural Information Processing Systems 29:*

Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain, pages 2802–2810, 2016.

[MSZ+11] Xing Mei, Xun Sun, Mingcai Zhou, Shaohui Jiao, Haitao Wang, and Xiaopeng Zhang. On building an accurate stereo matching system on graphics hardware. In *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, pages 467–474. IEEE, 2011.

[NYB+16] Anh Nguyen, Jason Yosinski, Yoshua Bengio, Alexey Dosovitskiy, and Jeff Clune. Plug & play generative networks: Conditional iterative generation of images in latent space. *arXiv preprint arXiv:1612.00005*, 2016.

[PF14] Daniele Perrone and Paolo Favaro. Total variation blind deconvolution: The devil is in the details. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 2909–2916. IEEE, 2014.

[PF16] Daniele Perrone and Paolo Favaro. A logarithmic image prior for blind deconvolution. *International Journal of Computer Vision*, 117(2):159–172, 2016.

[PR17] Tobias Plötz and Stefan Roth. Benchmarking denoising algorithms with real photographs. *arXiv preprint arXiv:1707.01313*, 2017.

[PSWS03] J. Portilla, V. Strela, M. J. Wainwright, and E. P. Simoncelli. Image denoising using scale mixtures of gaussians in the wavelet domain. *IEEE Transactions on Image Processing*, 12(11):1338–1351, Nov 2003.

[RB05] Stefan Roth and Michael J Black. Fields of experts: A framework for learning image priors. In *Computer Vision and Pattern Recognition (CVPR), 2005 IEEE Conference on*, volume 2, pages 860–867. IEEE, 2005.

[REM16] Yaniv Romano, Michael Elad, and Peyman Milanfar. The little engine that could: Regularization by denoising (red). *arXiv preprint arXiv:1611.02862*, 2016.

[RHB+11] Christoph Rhemann, Asmaa Hosni, Michael Bleyer, Carsten Rother, and Margrit Gelautz. Fast cost-volume filtering for visual correspondence and beyond. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 3017–3024. IEEE, 2011.

[ROD+10] Christian Richardt, Douglas Orr, Ian Davies, Antonio Criminisi, and Neil A Dodgson. Real-time spatiotemporal stereo matching using the dual-cross-bilateral grid. In *European Conference on Computer Vision*, pages 510–523. Springer, 2010.

[ROF92] Leonid I. Rudin, Stanley Osher, and Emad Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D: Nonlinear Phenomena*, 60(1):259 – 268, 1992.

[RS11] M Raphan and E P Simoncelli. Least squares estimation without priors or supervision. *Neural Computation*, 23(2):374–420, Feb 2011. Published online, Nov 2010.

[SCBHS13] Christian J Schuler, Harold Christopher Burger, Stefan Harmeling, and Bernhard Scholkopf. A machine learning approach for non-blind image deconvolution. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 1067–1074. IEEE, 2013.

[SCWH13] Libin Sun, Sunghyun Cho, Jue Wang, and James Hays. Edge-based blur kernel estimation using patch priors. In *Computational Photography (ICCP), 2013 IEEE International Conference on*, pages 1–8. IEEE, 2013.

[SHK+14] Daniel Scharstein, Heiko Hirschmüller, York Kitajima, Greg Krathwohl, Nera Nešić, Xi Wang, and Porter Westling. High-resolution stereo datasets with subpixel-

accurate ground truth. In *German Conference on Pattern Recognition*, pages 31–42. Springer, 2014.

[SJA08] Qi Shan, Jiaya Jia, and Aseem Agarwala. High-quality motion deblurring from a single image. In *ACM Transactions on Graphics (TOG)*, volume 27, page 73. ACM, 2008.

[SJN+16a] Uwe Schmidt, Jeremy Jancsary, Sebastian Nowozin, Stefan Roth, and Carsten Rother. Cascades of regression tree fields for image restoration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(4):677–689, 2016.

[SJN+16b] Uwe Schmidt, Jeremy Jancsary, Sebastian Nowozin, Stefan Roth, and Carsten Rother. Cascades of regression tree fields for image restoration. *IEEE transactions on pattern analysis and machine intelligence*, 38(4):677–689, 2016.

[SLR13] Robert Spangenberg, Tobias Langner, and Raúl Rojas. Weighted semi-global matching and center-symmetric census transform for robust driver assistance. In *International Conference on Computer Analysis of Images and Patterns*, pages 34–41. Springer, 2013.

[SM16] Tamar Rott Shaham and Tomer Michaeli. Visualizing image priors. In *European Conference on Computer Vision*, pages 136–153. Springer, 2016.

[SR14] Uwe Schmidt and Stefan Roth. Shrinkage fields for effective image restoration. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 2774–2781. IEEE, 2014.

[TL12] Yu-Wing Tai and Stephen Lin. Motion-aware noise filtering for deblurring of noisy and blurry images. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 17–24. IEEE, 2012.

[TM87] Alvin Toffler and John McHale. The future and the functions of art: A conversation between alvin toffler and john mchale. *Leonardo*, 20(4), 1987.

[TRF03] Marshall F. Tappen, Bryan C. Russell, and William T. Freeman. Exploiting the sparse derivative prior for super-resolution and image demosaicing. In *In IEEE Workshop on Statistical and Computational Theories of Vision*, 2003.

[TYLX17] Ying Tai, Jian Yang, Xiaoming Liu, and Chunyan Xu. Memnet: A persistent memory network for image restoration. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4539–4547, 2017.

[UVL17] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Deep image prior. *arXiv preprint arXiv:1711.10925*, 2017.

[VBW13] Singanallur V Venkatakrishnan, Charles A Bouman, and Brendt Wohlberg. Plug-and-play priors for model based reconstruction. In *GlobalSIP*, pages 945–948. IEEE, 2013.

[VLBM08] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th International Conference on Machine Learning*, pages 1096–1103. ACM, 2008.

[VLL+10] Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, and Pierre-Antoine Manzagol. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *J. Mach. Learn. Res.*, 11:3371–3408, December 2010.

[VRS14] Christoph Vogel, Stefan Roth, and Konrad Schindler. View-consistent 3d scene flow estimation over multi-

ple frames. In *European Conference on Computer Vision*, pages 263–278. Springer, 2014.

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

[VWST12] Vibhav Vineet, Jonathan Warrell, Paul Sturgess, and Philip Torr. Improved initialization and gaussian mixture pairwise terms for dense random fields with mean-field inference. In *Proceedings of the British Machine Vision Conference*, pages 73.1–73.11. BMVA Press, 2012.

[VWT14] Vibhav Vineet, Jonathan Warrell, and Philip HS Torr. Filter-based mean-field inference for random fields with higher-order terms and product label-spaces. *International Journal of Computer Vision*, 110(3):290–307, 2014.

[WYYZ08] Yilun Wang, Junfeng Yang, Wotao Yin, and Yin Zhang. A new alternating minimization algorithm for total variation image reconstruction. *SIAM Journal on Imaging Sciences*, 1(3):248–272, 2008.

[XHH+17] Lei Xiao, Felix Heide, Wolfgang Heidrich, Bernhard Schölkopf, and Michael Hirsch. Discriminative transfer learning for general image restoration. *arXiv preprint arXiv:1703.09245*, 2017.

[XRLJ14] Li Xu, Jimmy SJ Ren, Ce Liu, and Jiaya Jia. Deep convolutional neural network for image deconvolution. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 1790–1798. Curran Associates, Inc., 2014.

[YG14] Fisher Yu and David Gallup. 3d reconstruction from accidental motion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3986–3993. IEEE, 2014.

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

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

[YWHM10] Jianchao Yang, John Wright, Thomas S Huang, and Yi Ma. Image super-resolution via sparse representation. *IEEE transactions on image processing*, 19(11):2861–2873, 2010.

[ZCM+13] Lin Zhong, Sunghyun Cho, Dimitris Metaxas, Sylvain Paris, and Jue Wang. Handling noise in single image deblurring using directional filters. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 612–619. IEEE, 2013.

[ZEP10] Roman Zeyde, Michael Elad, and Matan Protter. On single image scale-up using sparse-representations. In *International Conference on Curves and Surfaces*, pages 711–730. Springer, 2010.

[ZFM+14] Kang Zhang, Yuqiang Fang, Dongbo Min, Lifeng Sun, Shiqiang Yang, Shuicheng Yan, and Qi Tian. Cross-scale cost aggregation for stereo matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1590–1597, 2014.

[ZL15] Jure Zbontar and Yann LeCun. Computing the stereo matching cost with a convolutional neural network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1592–1599, 2015.

[ZLL09] Ke Zhang, Jiangbo Lu, and Gauthier Lafruit. Cross-based local stereo matching using orthogonal integral images. *Circuits and Systems for Video Technology, IEEE Transactions on*, 19(7):1073–1079, 2009.

[ZN09] Changyin Zhou and Shree Nayar. What are good apertures for defocus deblurring? In *Computational Photography (ICCP), 2009 IEEE International Conference on*, pages 1–8. IEEE, 2009.

[ZW11] Daniel Zoran and Yair Weiss. From learning models of natural image patches to whole image restoration. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 479–486. IEEE, 2011.

[ZW13] Haichao Zhang and David Wipf. Non-uniform camera shake removal using a spatially-adaptive sparse penalty. In *Advances in Neural Information Processing Systems*, pages 1556–1564, 2013.

[ZY14] Haichao Zhang and Jianchao Yang. Scale adaptive blind deblurring. In *Advances in Neural Information Processing Systems*, pages 3005–3013, 2014.

[ZZC+16] Kai Zhang, Wangmeng Zuo, Yunjin Chen, Deyu Meng, and Lei Zhang. Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. *arXiv preprint arXiv:1608.03981*, 2016.

[ZZGZ17] Kai Zhang, Wangmeng Zuo, Shuhang Gu, and Lei Zhang. Learning deep cnn denoiser prior for image restoration. *arXiv preprint arXiv:1704.03264*, 2017.