Statistical Multiresolution Estimators in Linear Inverse Problems - Foundations and Algorithmic Aspects

Dissertation

zur Erlangung des Doktorgrades
der Mathematisch-Naturwissenschaftlichen Fakultäten
der Georg-August-Universität zu Göttingen

vorgelegt von

Philipp Marnitz

aus Hamburg

Göttingen, 2010

D7
Referent: Prof. Dr. Axel Munk
Koreferent: Prof. Dr. Russell Luke
Tag der mündlichen Prüfung:

Abstract

Applications of *statistical multiresolution techniques* in regression problems have attracted a lot of attention recently. The main reason for this is that the resulting *statistical multiresolution (SMR) estimators* are *locally and multiscale adaptive*, meaning that they automatically adjust to the smoothness of the true object on different scales and in different locations. In this dissertation, we introduce a novel algorithmic framework to compute SMR-estimators in practice.

On a theoretical level, we take a rigorous and general approach to SMR-estimators by defining them as the solution of a *constrained optimization problem*. We present a derivation of this approach and show a consistency result. The actual computation is carried out via an *Augmented Lagrangian method* by means of which the problem is decomposed into an unconstrained minimization problem and a *large-scale projection problem*. The latter is tackled by *Dykstra's algorithm*, a method which computes the projection onto the intersection of closed and convex sets by successively projecting onto single sets. These individual projections can be stated explicitly in our context which turns Dykstra's algorithm into a particularly fast and hence appealing solution method.

As a result, our methodology allows for treatment of comparatively large datasets. Especially *two-dimensional datasets* can be processed while most publications on the subject so far were restricted to a one-dimensional setting. When applied to regression problems, our method gives better results than state of the art methods in the field of SMR-estimation. Furthermore, our algorithm is the first that allows for computation of SMR-estimators for (possibly ill-posed) *inverse problems*. It can also be combined with a variety of *penalty functions*.

We demonstrate the performance of SMR-estimators computed by our algorithmic framework by presenting numerical examples. Apart from processing synthetic test objects to assess the quality of the estimators in different settings, we also give a practical application from biophotonic imaging in which a large-scale deconvolution problem needs to be solved.

Acknowledgements

I wish to thank my thesis advisor Prof. Dr. Axel Munk for proposing the subject and giving me the opportunity to carry out this research. I also thank my coadvisor Prof. Dr. Russell Luke for helpful discussions and proofreading the dissertation.

I am grateful to Dr. Klaus Frick for a fantastic collaboration during my time as a Ph.D. student and repeated proofreading. I thank Johannes Schmidt-Hieber for three pleasant years of sharing an office. Thanks to everyone else at the Institute for Mathematical Stochastics in Göttingen for creating a nice working atmosphere.

Needless to say that I am deeply indebted to my family - above all my parents Annegret and Jochen and my sister Laura - for providing constant support throughout my studies. Last but certainly not least, I express my deepest gratitude to my girlfriend Christiane Lenk. Without her, this dissertation would never have been finished.

Contents

1 Introduction

1.0.1 Data model

In this thesis, we are concerned with the solution of (possibly ill-posed) linear operator equations. For a known linear operator $K : H_1 \to H_2$ acting between Hilbert spaces H_1 and H_2, an unknown object u^* is to be reconstructed (or estimated in statistical terms) given its image $Ku^* = g$ under K. In many applications, g cannot be observed directly; only a perturbed observation Y of the form

$$Y = Ku^* + \sigma\varepsilon \tag{1.1}$$

is available. Here, $\varepsilon : H_2 \to L^2(\Omega, \mathcal{A}, \mathbb{P})$ is a *white noise process*, i.e. $(\Omega, \mathcal{A}, \mathbb{P})$ is a probability space, ε is linear and for $v, w \in H_2$ one has

$$\varepsilon(v) \sim \mathcal{N}(0, \|v\|^2) \quad \text{and} \quad \mathbf{Cov}\left(\varepsilon(v), \varepsilon(w)\right) = \langle v, w \rangle . \tag{1.2}$$

This problem formulation is very common in the theory of statistical inverse problems (see e.g. [8; 79; 87]) and covers numerous models arising in many applications. We illustrate the rather abstract formulation by the following example which will be of central importance for the algorithmic aspects treated in this thesis.

Example 1.0.1. Consider the case where $H_1 = H_2 = \mathbb{R}^X$ and $X = \{1, \dots, n\}^2$ for some $n \in \mathbb{N}$ is a finite lattice in $\mathbb{R}^2$. Put simply, we assume a two-dimensional dataset Y of size $n \times n$ to be given. Such datasets can be visualized as grayscale images. According to this interpretation, we will refer to $(i, j) \in X$ as pixels. In view of (1.1), the value of Y in each such pixel is given as

$$Y_{i,j} = (Ku)_{i,j} + \sigma\varepsilon_{i,j} \tag{1.3}$$

where $\varepsilon_{i,j} \overset{i.i.d.}{\sim} \mathcal{N}(0, 1)$. We note that while we assume the lattice X to be square for ease of notation, the analysis and algorithms in this thesis can easily be extended to rectangular lattices.

Remark 1.0.2. We will assume the *noise level* $\sigma > 0$ to be known throughout the thesis. Estimating the variance of a perturbed observation Y is a subject of its own and clearly beyond the scope of this work. For the sake of completeness, we refer to [31] and [85] and the references therein for robust estimators of the noise level in the setting at hand.

1.0.2 Inverse problems and regularization

We will now specify further which operators K in (1.1) are of special interest. The easiest case of a perturbed operator equation is given for $K = \mathrm{Id}$ and leads to so-called *denoising problems*. An example of such a problem with synthetic data in the two-dimensional setting of Example 1.0.1 is given in Figure 1.1.

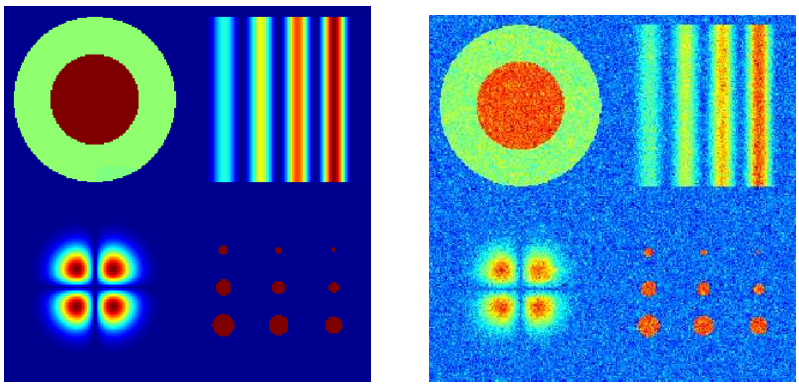

Figure 1.1: Left: synthetic test object "circles and bars" of size 256×256 with values scaled in $[0, 1]$. Right: object perturbed with noise, yielding an observation Y as in (1.3) with $\sigma = 0.1$ and $K = \mathrm{Id}$.

While we also treat denoising in the methodology that will be introduced in this thesis, especially *ill-posed operators* are of central importance. Considering this class of operators leads to the field of *inverse problems*, a mathematical discipline that has attracted a lot of attention in the past decades. We illustrate such a problem in Figure 1.2. There are many journals and numerous monographs ([3; 41; 70; 84; 102] to name but a few) dedicated to the subject, showing the vast amount of research in this field done recently. We give a formal definition of ill-posed operators which traces back to J. Hadamard.

Definition 1.0.3. An operator $K : H_1 \to H_2$ is said to be *well-posed* if

1. a solution $u \in H_1$ of $Ku = g$ exists for every $g \in H_2$;
2. the solution u is unique;
3. the solution u depends continuously on g.

In particular, the inverse K^{-1} of K is well-defined and continuous. An operator that is not well-posed is called *ill-posed*.

Remark 1.0.4. In the discrete setting of Example 1.0.1, no discontinuous operators K^{-1} exist in a strict sense. Calling an invertible discrete operator K ill-posed (by a slight abuse of notation) hence rather refers to its *condition number* κ_K defined as

$$\kappa_K := \limsup_{u \to v} \frac{\|Ku - Kv\|}{\|u - v\|}$$

being large. For a detailed treatment of discretized ill-posed problems, we refer to [56].

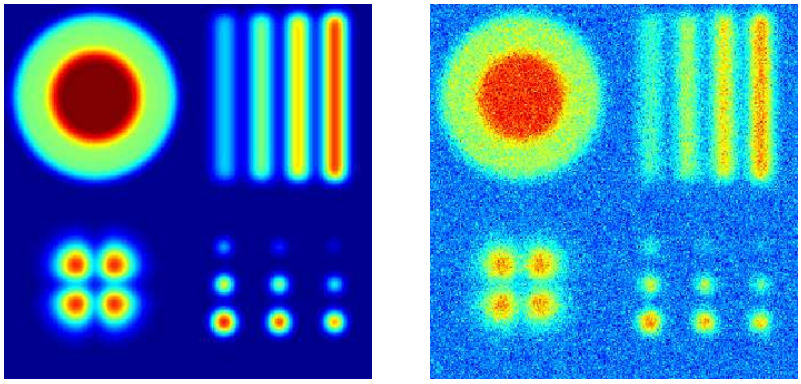

Figure 1.2: Left: image of "circles and bars" under a convolution operator with circular Gaussian kernel (see Section 4.2). Right: corresponding observation Y as in (1.3) with $\sigma = 0.1$.

When dealing with ill-posed operator equations, we always have to act on the assumption that small deviations in the image domain may lead to arbitrarily large errors when simply trying to invert K. Hence, computing a reconstruction $\hat{u}$ of u^* by solving $K\hat{u} = Y$ will lead to an unstable and hence useless solution due to the perturbation of Y by the white noise process ε.

We therefore need to impose some notion of regularity on our estimator. In order to do so, we have to make some assumption on the unknown object. Making use of prior knowledge if available, one might for example assume that it varies slowly from pixel to pixel, that it exhibits sharp edges, or that it has a sparse representation with respect to some fixed basis. Such formulations are usually referred to as *smoothness assumptions*. After such an assumption has been specified, the reconstruction of the object is forced to fulfill it up to a certain degree by construction of the method being used for its computation.

In order to establish this property of the estimator, this thesis focuses on *penalizing complexity*, a popular approach that can be applied to various problem settings (see e.g.

[16; 39; 72; 76; 103]). In this technique, the smoothness assumption is formalized via a so-called *penalty function* $J : H_1 \to \mathbb{R} \cup \{\infty\}$ which is chosen in a way that it becomes large for reconstructions that exhibit a behaviour which is considered unlikely for the true object u^*. Put differently, it gives a certain *measure of complexity* of elements in H_1 which is assumed to take a small value at u^*. In this thesis, we will frequently impose the following assumption on J.

Assumption 1.0.5. *$J : H_1 \to \mathbb{R} \cup \{\infty\}$ is convex, lower semi-continuous and proper. Recall that J is called lower semi-continuous if for all $u \in H_1$ and every $\varepsilon > 0$, there exists a neighborhood U of u such that $J(v) \geq J(u) - \varepsilon$ for all $v \in U$. Furthermore, J is called proper if the domain of J defined by*

$$D(J) := \{u \in H_1 : J(u) \neq \infty\}$$

is nonempty and $J(u) > -\infty$ for all $u \in H_1$.

After a penalty function J has been fixed, the computation of the actual reconstruction $\hat{u}$ is carried out by solving a constrained minimization problem of the form

$$J(u) \to \inf! \quad \text{subject to} \quad D(Y, Ku) \leq q. \tag{1.4}$$

In this formulation, $D : H_2 \times H_2 \to \mathbb{R}$ denotes some notion of *distance* to measure the deviation of Ku from the data Y. While a certain degree of smoothness is required for the estimator $\hat{u}$, its image under K should not be too far from the actual data Y at the same time. We will frequently refer to D as the *data-fit function*. A well-known example of such a function is the so-called least-squares data-fit given by

$$D(Y, Ku) = \frac{1}{2} \|Y - Ku\|_2^2 . \tag{1.5}$$

For this specific choice of D and a penalty function J that meets assumption 1.0.5, a solution $\hat{u}$ of (1.4) might also be computed by minimizing a variational scheme of the form

$$\hat{u}_a = \operatorname*{argmin}_{u \in H_1} \frac{1}{2} \|Y - Ku\|_2^2 + aJ(u). \tag{1.6}$$

Remark 1.0.6. In general, such a reformulation of (1.4) is possible if $G(u) := D(Y, Ku) - q$

satisfies the so-called *Slater condition*, i.e. there exists a $\bar{u} \in H_1$ such that

$$D(Y, K\bar{u}) - q < 0.$$

This can be derived e.g. from [40, Chapter 3, Proposition 5.1].

1.0.3 The regularization parameter

The scalar $q > 0$ in (1.4) ($a > 0$ in (1.6) respectively) regulates the balance between the data-fit and the penalty function and is usually referred to as the *regularization parameter*. The smaller q is chosen, the closer the image of the estimator under K will be to the data Y as the data-fit is emphasized more, yet the estimator will become unstable if q is chosen too small. This will lead to estimators which we will call *undersmoothed*. On the other hand, the larger q is chosen, the smoother the estimator will be, but the data-fit of the image under K will become poor if q is chosen too large. In this case, we will call the estimator *oversmoothed*. In summary, q controls the *trade-off between smoothing and data-fit*. Different choices of the regularization parameter may result in quite different estimators. We illustrate this in Figure 1.0.3 where solutions of (1.6) for different choices of the regularization parameter are presented.

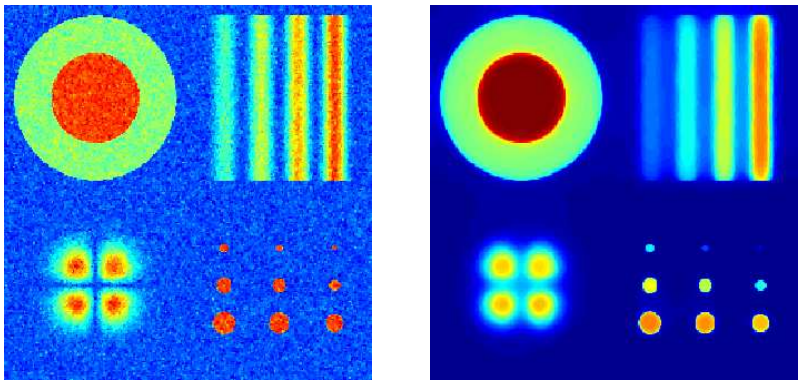

Figure 1.3: Results of (1.6) for $J = TV$ (see Section 2.4) and different parameters a where Y is as in Figure 1.1. Left: undersmoothed estimator, $a = 0.01$ was chosen too small. Right: oversmoothed estimator, $a = 1$ was chosen too big.

While it is of course crucial to choose D and J appropriately to the situation at hand, the choice of q remains critical and has to be done separately for each dataset as different objects exhibit different degrees of smoothness. This gets even more involved as the true

object is usually unknown in practical applications. It is hence desirable to formulate estimation schemes that are data-driven in the sense that no degree of freedom like q in (1.4) is present for each individual dataset.

Many techniques to approach such schemes have been introduced already, for example the discrepancy principle, generalized cross validation, the L-curve method or the unbiased predictive risk estimator, to name but a few. A summary and analysis of these methods is given in [104, Chapter 7]. Moreover, we also mention the Lepskij principle, first introduced in [75], and the risk hull method, see [20].

The major drawback of these methods is that they all aim at choosing a *global* parameter to regulate the influence of a *global* data-fit function (like the least-squares data-fit (1.5)) on the estimator. Smoothness, however, is not a global feature at all and may vary from location to location and from scale to scale within one fixed object. This can for example be seen in the test object u^* displayed in Figure 1.1 which consists of smoother (big circle in the top left; background) and less smooth regions (dots in the bottom right) of different sizes. When computing an estimator given a perturbed version of u^*, those regions would hence require different balancing of smoothing and data-fit; a task that a global scheme like (1.6) simply cannot cope with.

1.0.4 Objectives of the thesis

These considerations lead to the starting-point of this dissertation. We give a survey of its objectives in the following. In view of the challenges just described, the key issue of this thesis is the derivation of a *fully data-driven* estimation scheme that establishes balance between smoothing and data-fit *locally and multiscale adaptive*, meaning that it automatically adapts to the local smoothness of the unknown object on different scales. Rather than by automatically determining a regularization parameter for given data-fit and penalty function, this goal is reached by formulating an appropriate data-fit function D for the optimization problem (1.4) which at the same time allows for automatic determination of an upper bound q.

This specific data-fit function is based on properties of the white noise process ε. We will use an extreme-value statistic of a projection of the residuals - the so-called *multiresolution statistic* and the related *multiresolution criterion* - to formulate the function. In short, this criterion decides whether or not the residuals $Y - K\hat{u}$ of a given estimator $\hat{u}$ still contain nonrandom structures by performing a statistical test on them. For this reason,

our estimation scheme may be regarded as *statistically sound*.

At the same time, we aim at keeping our estimation framework as general as possible. In particular, it should not be too restrictive about possible smoothness assumptions on the object and hence *combinable with a wide range of penalty functions J*. This flexibility should not only be guaranteed on a theoretical level, but also be reflected in a certain modularity of the computation method, meaning that replacements of J can be done without changing the surrounding framework.

In major parts of the thesis, special emphasis is placed on algorithmic aspects that arise from this data-driven estimation scheme. The actual computation of the corresponding estimators amounts to the solution of a *constrained optimization problem*. This problem includes a vast number of inequality constraints and is therefore hard to tackle numerically. Establishing a novel algorithmic framework that guarantees *computability of the estimators in practical applications* should hence be regarded as the main achievement of this dissertation. We reach this goal by applying an *Augmented Lagrangian method* (cf. [43]) which we combine with *Dykstra's algorithm* [38] for computing projections onto the intersection of closed and convex sets. The resulting algorithm turns out to be very efficient and enables computation of the estimators, especially for the numerically involved two-dimensional setting of Example 1.0.1. However, the methodology is so versatile that it could be employed in arbitrary dimensions, in particular to one-dimensional datasets, too. Nonetheless, we focus on the two-dimensional setting in this work.

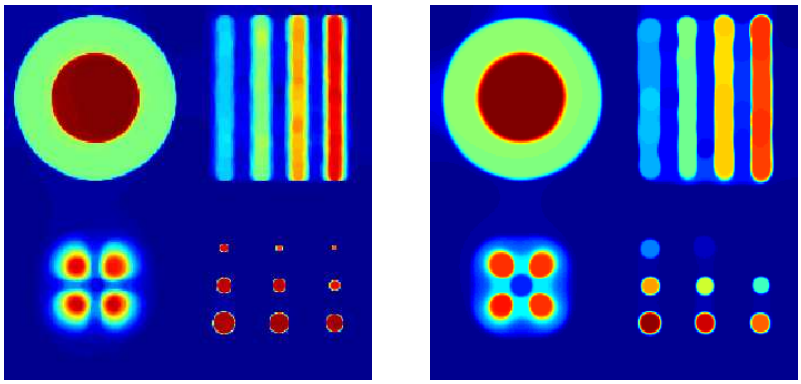

Figure 1.4: Results of our methodology for $J = \mathrm{TV}$ (see Section 2.4). Left: denoising problem of Figure 1.1. Right: inverse problem of Figure 1.2.

In Figure 1.4, we demonstrate the performance of our methodology by showing its results for the observations of Figures 1.1 and 1.2. We remark once more that these estimators

were computed in a completely data-driven manner. By visual inspection we find that the estimators exhibit the desired locally adaptive behavior. Small features are preserved (at least those which are not completely lost in the observation Y), while flat areas are well-smoothed at the same time. As the inverse problem is substantially harder to tackle numerically, it is not surprising that the resulting estimator is obviously further from the true object than the estimator for the denoising problem illustrated in Figure 1.1. Nonetheless, our method gives good and convincing results in both situations.

All in all, the thesis brings together elements from various mathematical disciplines. In order to apply the *statistically* sound estimation scheme to the *inverse problem* at hand, we will make use of *optimization techniques*. When applying these techniques, Dykstra's *projection algorithm* will be used in order to solve a large-scale quadratic program. Furthermore, the resulting algorithms were implemented in Matlab and C++, respectively. This needed to be done very carefully to keep computation times within reason in spite of the expansive algorithms being performed.

1.0.5 Outline

The thesis is organized as follows. We will start with a formal definition of the multiresolution statistic and describe how it can be used to characterize the desired statistically sound estimators in Chapter 2. There, we will also state segmentation techniques and a method to compute penalized least-squares estimators when J is chosen as total variation, both of which will be needed in later chapters. In Chapter 3, we will present our methodology to compute estimators based on the multiresolution statistic. We will illustrate its performance by showing results of numerical experiments and give some extensions including an application to photonic imaging in Chapter 4. The thesis is concluded by a discussion of the results achieved and an outlook on possible future work in Chapter 5.

2 Basic concepts

In this chapter, we will give basic definitions and core ideas needed to reach our goal of computing an estimator of the unknown object u^* in (1.1) in a statistically sound way. We will start with a formal definition of the *multiresolution (MR) statistic*, a description of its properties and an interpretation of the resulting *multiresolution criterion* in Section 2.1. Afterwards, basic ideas on how to apply the MR-statistic to the problem at hand will be presented in Section 2.2. There, we will also lay theoretical foundations for the resulting *statistical multiresolution estimator*, namely conditions for its existence and a consistency result.

As we aim at computing an estimator which is not only statistically sound but also *locally adaptive*, meaning that it adapts to the locally varying smoothness of the true object, certain segmentation techniques are needed. Those will be given in Section 2.3. Finally, we will introduce the total variation semi-norm which we used as an example of a penalty function J in our experiments and state a method for computation of the corresponding minimizers of (1.6) in Section 2.4.

2.1 Definition of the multiresolution statistic

As briefly mentioned in the introduction, the basic idea behind the multiresolution statistic can be summarized as follows: given an observation Y as in (1.1), an estimator $\hat{u}$ is considered satisfactory as long as the resulting residuals

$$r = \sigma^{-1}(Y - K\hat{u}) \tag{2.1}$$

behave like white noise in a certain sense. If the estimator depicts all features of the true object u^* well, r will only consist of noise. If, on the other hand, there is some structure of u^* left in r, the estimator must have missed some of the object's essential features and is hence considered unacceptable.

In order to decide whether or not the residuals still contain some nonrandom structure, we perform a statistical test on them. This test makes use of a set of test functions. To this end, we start by giving the following definition.

Definition 2.1.1. A set

$$\Phi := \{\phi_1, \phi_2, \ldots\} \subset \overline{\operatorname{ran}(K)} \setminus \{0\}$$

of *test functions* ϕ_i with $\|\phi_i\| \leq 1$ will be called a *dictionary*.

While this definition is rather abstract at first glance, a dictionary might for example correspond to the choice of a certain segmentation of the image domain H_2 in the discrete case of Example 1.0.1. We refer to Section 2.3 where different choices of Φ in this setting are presented. For now, we continue by defining the actual statistic.

Definition 2.1.2. Assume that $r \in H_2$ and Φ is a given dictionary. We define the *average function* over $\phi \in \Phi$ as

$$\mu_\phi(r) := \frac{|\langle r, \phi \rangle|}{\|\phi\|}. \tag{2.2}$$

For an additional function $f : [0,1] \to \mathbb{R}$ and $N \in \mathbb{N}$, the *multiresolution (MR) statistic* T_N is defined as

$$T_N(r) := \sup_{1 \leq i \leq N} \mu_{\phi_i}(r) - f(\|\phi_i\|). \tag{2.3}$$

We provide an outline of the history of the MR-statistic. It was first introduced in [98] to detect change points, [99] extended this application to the detection of a signal against a noisy background. The MR-statistic was also used to formulate a stopping rule for the EM algorithm in [9] and [10], therefore applying it to positron emission tomography. In [36] and [37], the authors introduced the statistic to the context of testing qualitative hypotheses in non-parametric regression. In [29] it was first used for non-parametric regression of one-dimensional functions, focusing on local extremes. There, the authors employed it to determine a localized form of the regularization parameter in (1.6) where the penalty function J is chosen to be the total variation semi-norm. This approach was extended to two dimensions in [100] where inhomogeneous diffusion was used as a reconstruction method; a methodology which was later refined in [64]. Confidence regions for the MR-statistic were given in [30] where the resulting estimation scheme was first formulated as a constrained optimization problem.

In this thesis, we introduce the MR-statistic as a tool to perform a statistical test which we motivate in the following. If an estimator $\hat{u}$ failed to recover features of u^*, the mean of some of the residuals will no longer be 0. In order to detect such residuals by means of a dictionary Φ, we test the null hypothesis

$$\mathcal{H}_0 : \mathbf{E}\left(\mu_\phi(r)\right) = 0 \quad \text{for all} \quad \phi \in \Phi \tag{2.4}$$

against the alternative

$$\mathcal{H}_1 : \mathbf{E}\left(\mu_\phi(r)\right) \neq 0 \quad \text{for some} \quad \phi \in \Phi.$$

This concept was first introduced in [99]. In order to use the MR-statistic T_N to perform this test, we note that if r deflects from $\mathcal{H}_0$ along some $\phi \in \Phi$, the residuals there are no longer distributed around zero. Hence, the absolute values of the projections of the residuals onto this ϕ will be significantly larger than one would expect for pure white noise. In other words, if we choose a proper dictionary Φ, the average function $\mu_\phi(r)$ will become large for at least one $\phi \in \Phi$ in this case and so will $T_N(r)$. Consequently, we will reject $\mathcal{H}_0$ if $T_N(r)$ exceeds a certain threshold value q.

While asymptotic behaviour of the statistic was used e.g. in [29] and [100] in order to derive a choice of the critical value q, we propose to choose $q = q_N(\alpha)$ according to the $(1-\alpha)$-quantile of $T_N(\varepsilon)$, that is

$$q = q_N(\alpha) := \inf\left\{q \in \mathbb{R} : \mathbb{P}(T_N(\varepsilon) \leq q) \geq 1 - \alpha\right\} \tag{2.5}$$

for some $\alpha \in (0,1)$. In practice, this quantile may be estimated by performing Monte Carlo simulations of $T_N(\varepsilon)$.

This approach leads to an additional degree of freedom, namely the choice of the *significance level* α: the probability that a given instance of the white noise process ε is bigger than $q_N(\alpha)$ is at most α. In other words, α constitutes an upper bound on the *error of the first kind* when testing a given $r \in H_2$ for the null hypothesis $\mathcal{H}_0$. Note that α is a significance level in the sense just described for the statistical test using all test functions $\phi_1, \ldots, \phi_N$ *simultaneously*. Multiple testing on each individual ϕ_i (which would result in a drastically reduced significance level over all N test functions due to multiplicity) is therefore avoided. Despite the fact that α may be chosen at will, the estimator resulting from our theoretical considerations may still be called fully data-driven as we will see in the next section. For now, we summarize our considerations so far in a formal definition.

Definition 2.1.3. We say that an estimator $\hat{u}$ to a given observation Y as in (1.1) *fulfills the multiresolution (MR) criterion* with respect to a dictionary Φ, $N \in \mathbb{N}$ and a significance level $\alpha > 0$ if

$$T_N(\sigma^{-1}(Y - K\hat{u})) \leq q_N(\alpha).$$

Otherwise, $\hat{u}$ is said to *violate the multiresolution criterion*.

In summary, our statistical test will hence be performed in the following way: for given observation Y and estimator $\hat{u}$, we will reject $\mathcal{H}_0$ with respect to the residuals $r = \sigma^{-1}(Y - K\hat{u})$ and hence the estimator itself if $\hat{u}$ violates the MR-criterion. If, on the other hand, $\hat{u}$ fulfills the MR-criterion, we will not reject the hypothesis that r only contains white noise and will therefore accept the estimator $\hat{u}$. Note that the MR-criterion hence subdivides H_1 into acceptable and unacceptable estimators. We have therefore only characterized a set of estimators which are feasible in terms of the MR-statistic so far. We postpone the question of how to pick a suitable estimator from this set to Section 2.2.

Up to now, we have focused our analysis on the average function μ_ϕ. Yet the second component of the MR-statistic, namely the function f, has not been discussed. For now, just note that it depends on the norm of the test functions ϕ_i only and is not related to the residuals r being tested. It therefore rather modifies the design of the test itself and not the way that the residuals influence its outcome. The interpretation of the test as given in this section is hence independent of the particular choice of f. Put differently, f allows for control of the extent to which test functions with equal norm contribute to the statistic and hence balances ϕ_i's of different norms. A more detailed treatment of this function will be given in Sections 2.2 and 2.3 where a sound choice of f in the discrete two-dimensional setting of Example 1.0.1 will be stated.

2.2 The statistical multiresolution estimator

An important remark about the multiresolution criterion is related to its one-sidedness. While we could have established a lower bound on the statistic in Definition 2.1.3 as well (e.g. by choosing a quantile which is *smaller* than $T_N(\varepsilon)$ with a certain probability, analogously to (2.5)), we only limited its value from above by the critical value $q_N(\alpha)$. As a consequence, an estimator $\hat{u}$ will only violate the MR-criterion if the corresponding residuals r become large in absolute value. This will happen if the estimator $\hat{u}$ is oversmoothed which will for example be the case if $\hat{u} = \hat{u}_q$ is computed as a solution of (1.4) and the regularization parameter q is chosen too big. If, on the other hand, $\hat{u}$ is undersmoothed, the residuals will get small in absolute value and so will $T_N(r)$ and $\hat{u}$ is likely to fulfill the MR-criterion.

Consider for example the case of denoising where $K = \mathrm{Id}$. If we simply take the observation Y as an estimator in this setting (e.g. by choosing $a = 0$ in (1.6)), the residuals r will all be zero and the multiresolution criterion will trivially be fulfilled although the estimator is certainly far from being satisfactory. While this is an exaggerated example, it nonetheless reveals an essential drawback of the MR-criterion: it is not capable of detecting undersmoothing.

2.2.1 Definition and interpretation

After this insight, we are now ready to specify how to pick an estimator from all those that are feasible in the sense of Definition 2.1.3. As the MR-criterion does not guarantee acceptable estimators to exhibit any smoothness at all, we will consequently choose our estimator according to the paradigm that we pick *the smoothest estimator which fulfills the MR-criterion*. We formalize this in the following definition.

Definition 2.2.1. For a dictionary Φ, a significance level $\alpha \in (0, 1)$ and a penalty function $J : H_1 \to \mathbb{R} \cup \{\infty\}$, a solution $\hat{u}_N(\alpha)$ of the optimization problem

$$J(u) \to \inf! \quad \text{subject to} \quad T_N(\sigma^{-1}(Y - Ku)) \leq q_N(\alpha) \tag{2.6}$$

will be called *statistical multiresolution (SMR) estimator*.

At this point, the connection between SMR-estimators and the estimation scheme (1.4) becomes obvious. In terms of the general framework presented there, we have used the MR-

criterion of Definition 2.1.3 to formulate a data-fit function which leads to the automated estimation scheme we claimed in the introductory Chapter 1. Moreover, we also point out the similarity between the SMR-estimator for $J = L^1$ and the Dantzig selector introduced in [19]. The fundamental difference between these two, however, is the fact that the constraints on the residuals for the latter are formulated with respect to the L^∞-norm instead of the SMR-statistic used here.

In view of (2.5) and (2.6), we see that an SMR-estimator is the smoothest estimator which lies within a $(1-\alpha)$ confidence region of the data, also see [30]. While α may still be chosen at will and can be regarded as a regularization parameter by itself (and so can the number N of test functions taken into account), the resulting value of $q_N(\alpha)$ can be used independently of the particular observation Y at hand as long as H_1, H_2, Φ and N remain unchanged. We will therefore continue to call the SMR-estimator *data-driven*.

In a statistical interpretation of Definition 2.2.1, the true object is an element of the feasible region of the optimization problem defined by the first N test functions with a probability of $(1-\alpha)$. This makes the parameter selection rule $q := q_N(\alpha)$ in (2.5) (which could in general be used in combination with any data-fit function of the form $D(Y, Ku) = d(Y - Ku)$ for some $d : H_2 \to \mathbb{R}$) meaningful in a statistical sense, especially in contrast to simply regarding q as a tuning parameter that has no more subtle meaning.

From an algorithmic point of view, note that the number of side constraints of the optimization problem (2.6) becomes extremely large in a two-dimensional setting if the underlying dictionary is chosen in one of the ways that will be suggested in Section 2.3. Tackling this numerically challenging problem is therefore one of the key issues of this thesis. The corresponding methodology will be introduced in Chapter 3.

2.2.2 Theoretical background

For the remainder of this section, we will focus on theoretical issues, giving sufficient conditions for the existence of an SMR-estimator and a consistency result. In this respect, we will also explain how the function f in Definition 2.3 can be chosen appropriately from a theoretical perspective, an open question which was postponed in Section 2.1. As this thesis mainly deals with practical application and algorithmic aspects, we will skip the proofs of the following theorems and refer to [44] for a solid theoretical background of these results.

Theorem 2.2.2. *Assume that $J : H_1 \to \mathbb{R} \cup \{\infty\}$ satisfies Assumption 1.0.5 and that in addition:*

1. *There is a $N_0 \in \mathbb{N}$ such that for all $c \in \mathbb{R}$ the sets*

$$\left\{ u \in H_1 : \sup_{1 \leq n \leq N_0} \mu_{\phi_n}(Ku) + J(u) \leq c \right\}$$

are sequentially weakly pre-compact.

Then an SMR-estimator exists for all $N \geq N_0$ and $\alpha \in (0, 1)$.

We comment on how the assumptions of this theorem translate to the setting being treated in this thesis. First, note that Assumption 1.0.5 imposes a rather weak restriction on J, allowing many popular choices of J with the total variation semi-norm as introduced in Section 2.4 and used in our numerical examples among them. Assumption 1 in the theorem is rather technical and aims at a certain interaction between Φ, K and J: for a given operator K, the dictionary Φ and the penalty function J must be chosen in a way that a deviation in H_1 can either be measured in its image under K by means of Φ or detected due to an increment in the complexity J. In Section 2.4, we will state a sufficient condition for this assumption to hold which allows for an easy verification in the special case where J is chosen to be the total variation semi-norm. For general penalty functions, however, one would have to verify the assumption in a different way.

Let us now investigate the asymptotic behaviour of $\hat{u}_N(\alpha)$ as the noise level σ in (1.1) tends to zero, therefore giving consistency results for the estimator. As N and α serve as regularization parameters of the SMR-estimator, they have to be chosen in a way that $\alpha \to 0$ and $N \to \infty$ at appropriate speed when considering the asymptotic case. Since we imposed only rather weak restrictions on the penalty function J in Theorem 2.2.2, it would be too optimistic to expect norm-convergence of $\hat{u}_N(\alpha)$ to a solution of the equation $Ku = g$ independently of the concrete choice of J. In fact, our result establishes convergence in *Bregman-divergences* (first introduced in [13]) which we formally define here.

Definition 2.2.3. For $u, v \in H_1$ and $J : H_1 \to \mathbb{R}$, we define the *Bregman-divergence of u and v with respect to J* as

$$D_J(u, v) = J(u) - J(v) - J'(v)(u - v)$$

where $J'(v)(u - v)$ denotes the directional derivative of J at v in direction $(u - v)$.

Remark 2.2.4. Clearly, the Bregman-divergence does not define a (quasi-)metric on H_1: It is non-negative but in general not symmetric. Moreover, it does not satisfy the triangle

inequality. The advantage of formalizing asymptotic results like consistency or convergence rates with respect to the Bregman-divergence, however, is the fact that the regularizing properties of the penalty function J being used are incorporated automatically. If, for example, J is slightly more than strictly convex, it was shown in [93] that convergence with respect to the Bregman-divergence already implies convergence in norm. If, however, J fails to be strictly convex (e.g. if it is of linear growth) it is in general hard to establish norm-convergence results, yet convergence results with respect to the Bregman-divergence, though weaker, may still be at hand. The concept of Bregman-divergence has attracted much attention recently, especially in the inverse problems community (cf. [17; 18; 27; 45; 94]).

After these preparations, we are now ready to state the consistency result for the SMR-estimator $\hat{u}_N(\alpha)$ of Definition 2.2.1.

Theorem 2.2.5. *Under the assumption that*

$$g \in \overline{\operatorname{span}\{\Phi\}} \quad \text{and} \quad \sup_{n\in\mathbb{N}} \mu_{\phi_n}(\varepsilon) - f(\|\phi_n\|) < \infty \quad \text{a.s.} \tag{2.7}$$

one can choose appropriate parameters $\alpha = \alpha(\sigma)$ and $N = N(\sigma)$ such that

$$\limsup_{\sigma\to 0^+} \|\hat{u}_N(\alpha)\| < \infty \quad \text{and} \quad \lim_{\sigma\to 0^+} D_J(u^\dagger, \hat{u}_N(\alpha)) = 0 \quad \text{a.s.} \tag{2.8}$$

for all J-minimizing solutions $u^\dagger$ *of* (1.1) *which are characterized by $Ku^\dagger = g$ and*

$$J(u^\dagger) = \inf_{u\in H_1} \{J(u) \ : \ Ku = g\}\,.$$

Proof. See [44, Theorem 3.6].

□

In order to guarantee consistency of the SMR-estimator, we hence need to verify (2.7). Closer examination shows that Φ must be chosen sufficiently rich to guarantee the first assumption made there. For the second assumption to hold, the function f in Definition 2.1.2 must be chosen appropriately. General conditions have been formulated in [36] and [37] for the particular choice

$$f(x) = \sqrt{-\gamma \log x} \quad \text{where} \quad \gamma > 0 \tag{2.9}$$

in terms of the ε-covering number N_ε of Φ, that is the minimal number of ε-balls needed to cover Φ. According to [37, Theorem 7.1], we find that whenever there exist constants $A, B > 0$ with

$$N_{st}(\{\phi \in \Phi \ : \ \|\phi\| \leq t\}) \leq A s^{-B} t^{-\gamma} \quad \text{for all} \quad s, t \in (0, 1], \tag{2.10}$$

the choice of f in (2.9) results in the second assumption in (2.7) to hold and hence in the SMR-estimator to be consistent in the sense of (2.8). Obviously, γ needs to be chosen dependent on the dictionary Φ for (2.10) to be fulfilled. When discussing possible choices of Φ in a discrete two-dimensional setting in Section 2.3, we will state the corresponding values of γ and therefore reveal how f was chosen in our numerical experiments.

We conclude our theoretical considerations by summarizing that the SMR-estimator and the statistic it is based on have a solid background not only due to the heuristics formulated in the last and at the beginning of this section but also from a theoretical point of view. It is hence worthwhile to make an effort to develop efficient numerical methods to solve the optimization problem (2.6) despite the fact that it is extremely large-scale. We once more refer to Chapter 3 where such methods will be derived.

2.3 Choice of dictionary Φ

By now, our considerations about the MR-statistic T_N were rather abstract as the dictionary Φ was only formally defined in Definition 2.1.1. In this section, we will give two concrete examples of possible choices of Φ in a discrete two-dimensional setting and connect them to the consistency result presented in Theorem 2.2.5. We will also indicate methods to quickly evaluate the MR-statistic for these specific choices of Φ. The dictionaries presented here will later be used in order to apply our theory and actually compute SMR-estimators in practice.

2.3.1 Characteristic functions of subsets

We start out by relating the dictionary from Definition 2.1.1 to subsets of the image domain in the discrete two-dimensional case.

Example 2.3.1 (cont. Example 1.0.1)**.** In the setting of Example 1.0.1, we will choose our test functions as

$$\phi = \phi_S := n^{-2}\chi_S$$

where $S \subset X = \{1, \dots, n\}^2$ for all $\phi \in \Phi$ throughout the remainder of this thesis. Here, $\chi_S : X \to \{0, 1\}$ denotes the characteristic function of S which takes the value 1 in S and 0 everywhere else. For this choice of Φ, the MR-statistic (2.3) transforms to

$$T_N(r) = \sup_{1 \le k \le N} \frac{\left|\sum_{(i,j)\in S_k} r_{i,j}\right|}{\sqrt{\#S_k}/n} - f(\sqrt{\#S_k}/n). \tag{2.11}$$

According to this formula, we will frequently identify $\|\phi\|$ with the *scale* of ϕ in the following. This relates our SMR-estimator to the multiscale property which we claimed in the introduction. It can be established by choosing subsets of different sizes and adding their characteristic functions to the dictionary Φ. Furthermore, we will sometimes identify a subset $S \subset X$ with the test function χ_S in a slight abuse of notation if it increases simplicity. In particular, we will say that a set S *yields a violation* for given $\hat{u} \in H_1$ and critical value q if

$$\frac{\left|\sum_{(i,j)\in S} r_{i,j}\right|}{\sqrt{\#S}/n} - f(\sqrt{\#S}/n) > q \quad \text{where} \quad r = \sigma^{-1}(Y - K\hat{u}),$$

i.e. if $\chi_S \in \Phi$ causes the estimator $\hat{u}$ to violate the MR-criterion in the sense of Definition 2.1.3.

2.3.2 Examples of partitionings

We will now describe how to choose a system of subsets

$$\mathcal{P} = \{S_1, S_2, \ldots\} \subset X$$

and therefore a dictionary Φ appropriately for our purpose. In this thesis, such a system will frequently be called a *partitioning* although it not necessarily constitutes what is usually called a partition in image processing, i.e. a disjoint decomposition of X. Nevertheless, the term partitioning will be used for the sake of brevity.

Clearly, such a partitioning should contain sets of different sizes in order to achieve the desired multiscale property of the SMR-estimator. In addition, the partitioning should be rich enough to detect deviations from $\mathcal{H}_0$ in different locations and therefore consist of enough sets to at least cover the whole domain X. On the other hand, it should not be chosen too rich either, as the consistency result of Theorem 2.2.5 would break down otherwise (cf. Subsection 2.3.3). Another important issue is the one of computability. The algorithms we will derive in Chapter 3 require numerous evaluations of $T_N(r)$. We should hence make sure that the average function $\mu_{\chi_S}(r)$ in (2.2) can be computed quickly for all $S \in \mathcal{P}$ when choosing our partitioning.

We will now state two partitionings which meet all of these requirements. In fact, these are the partitionings we used for our experiments. The first one is the so-called *dyadic squares partitioning* used e.g. in [35] and [71]. By splitting the image recursively into four equal subsquares until some pre-specified lowest scale $s_{\min}$ is reached (see Figure 2.1 for an illustration), one receives a partitioning that covers many different scales with comparatively few subsets. We will denote this partitioning by $\mathcal{P}_D$.

A second approach to create a partitioning that consists of squares is to fix a certain set of scales $\{s_1, \ldots, s_m\} \subset \mathbb{N}$ and let $\mathcal{P}$ comprise all squares with such side lengths in the image domain. Such a partitioning will be called *all squares partitioning* and denoted by $\mathcal{P}_A$ in this thesis. Clearly, this partitioning is a superset of $\mathcal{P}_D$ (at least if the corresponding side lengths were considered) containing much more elements than the latter. As a consequence, it is more involved computationally, but allows for a better detection of nonrandom structures

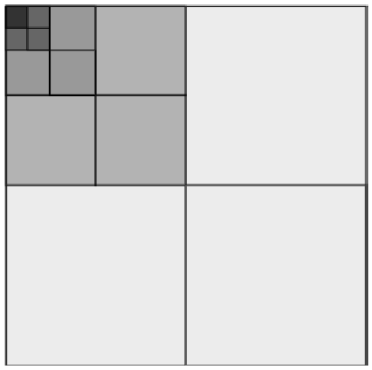

Figure 2.1: Different scales of a dyadic squares partitioning.

in the residuals as the model that the null hypothesis $\mathcal{H}_0$ in (2.4) is based on is refined. Although we assume the subsets in the partitioning to be squares in order to simplify notation, rectangles could be used as well.

We remark that prior information about the geometry of features of the unknown object u^* may be used to individually choose a partitioning. If u^* is believed to consist of, say, long and narrow features, long and narrow subsets should be taken into account in the partitioning. On the other hand, subsets which are shaped in a way that is considered unlikely for features of u^* could be left out in order to speed up computation time. Nonetheless, partitionings that consist of squares like the two presented here are rather general and adequate to many situations as they cover a lot of different geometric features that a test object might exhibit. At the same time, the structure of these partitionings is rather simple and easy to implement in contrast to more advanced segmentation techniques.

2.3.3 Covering number

At this point, we return to the theoretical background of SMR-estimators as established in Subsection 2.2.2. The partitionings $\mathcal{P}_A$ and $\mathcal{P}_D$ allow for a theoretically sound choice of $\gamma > 0$ in the formula (2.9) of the function f in Definition 2.3 of the MR-statistic. In fact, it was proved in [44, Proposition A.6] that for the setting treated in this section and a dictionary consisting of characteristic functions of squares, $\gamma = 2$ guarantees (2.10) to hold true and hence the SMR-estimator to be consistent in the sense of (2.8). We will therefore choose

$$f(x) = \sqrt{-2 \log x} \tag{2.12}$$

throughout the thesis when dealing with the two-dimensional setting at hand.

In general, however, the covering number in (2.10) constitutes an upper bound on the complexity of partitionings that may be used in SMR-estimation. The partitioning may not be chosen too rich in order to preserve consistency of the SMR-estimator. If for example $\mathcal{P}$ is chosen as the system of all closed and convex sets in $\{x \in \mathbb{R}^2 : \|x\| \leq 1\}$, no γ will exist to guarantee (2.10) according to a result shown in [15, Theorem 6] and the estimator's consistency will break down. This corresponds to the heuristic argument that when looking at a random signal on arbitrarily many subsets, one will always find some subset on which the signal appears to be nonrandom, even if it is in fact a realization of a white noise process. This restriction should always be kept in mind when choosing a partitioning.

2.3.4 Fast summation

Both partitionings presented in Subsection 2.3.2 allow for fast computation of the average function μ_ϕ as required, i.e. according to (2.11) for a method to quickly compute sums over all squares in the partitioning. We outline the corresponding method which we used in our implementation. This method makes use of the so-called *matrix of cumulative sums*. Put simply, entry (i, j) of this matrix holds the sum of the residuals r over the discrete rectangle $[1, i] \times [1, j] \subset X$.

Definition 2.3.2. For a given $r \in \mathbb{R}^{n \times n}$, the *matrix of cumulative sums* R is defined by

$$R_{i,j} = \sum_{k=1}^{i} \sum_{l=1}^{j} r_{k,l}$$

and the additional convention that $R_{0,0} = R_{i,0} = R_{0,j} = 0$ for all $i, j = 1, \ldots, n$.

After the matrix R was precomputed, we readily obtain the sum of r over a square $S = [i_1, i_2] \times [j_1, j_2] \subset \{1, \ldots, n\}^2$ via

$$\sum_{(i,j) \in S} r_{i,j} = R_{i_2,j_2} - R_{i_1-1,j_2} - R_{i_2,j_1-1} + R_{i_1-1,j_1-1}. \tag{2.13}$$

This formula corresponds to "cutting" the square from the image domain X. When evaluating (2.11) for a fixed r and a large number of test functions, this approach is clearly faster

than summing up over each individual square $S \in \mathcal{P}$ directly. When incorporating further partitionings into the framework of SMR-estimation, one would have to come up with a similar method that allows for fast summation over the sets contained in the partitioning. Otherwise, the methodology presented in Chapter 3 is likely to become too expansive computationally.

2.4 Total variation

As explained in the introduction, some assumption on the smoothness of the true object u^* is indispensable when dealing with operator equations like (1.1). The actual reconstruction is then carried out in a way that guarantees the resulting estimator to exhibit this smoothness up to a certain level. One idea to achieve this goal is the formulation of a *penalty function* $J : H_1 \to \mathbb{R} \cup \{\infty\}$ as in (1.6) which takes large values for elements in H_1 that lack the expected smoothness. So far, we have only treated J on this abstract level. If SMR-estimators are to be computed in practice, however, some concrete J will have to be specified. In this section, we give a definition of a popular choice of such a penalty function, the *total variation semi-norm* or TV for short, study its properties and state a method how to compute TV-penalized least-squares estimators. As in the previous section, we will restrict our algorithmic considerations to the discrete two-dimensional setting of Example 1.0.1.

Before we start our analysis of total variation, we emphasize that it is not the objective of this thesis to propagate this or any other specific choice of J but to depict how the amount of smoothing needed can be chosen in a data-driven way. It is in fact an advantage of our methodology that it can be combined with a wide range of possible choices of J. We therefore treat J on the abstract level as described above for most of the thesis and only use total variation as an exemplary choice of J in order to demonstrate the performance of our techniques in practical applications. Which specific J to choose depends on the particular situation and aim of the reconstruction and is not the key issue of this thesis. For this reason, we only outline how the computation of TV-penalized least-squares estimators was performed in our experiments and abstain from a more detailed treatment of the method and its background as this is beyond the scope of this thesis.

2.4.1 Definition and properties

Total variation can be defined rigorously in a continuous setting (see e.g. [48]). To this end, let $\Omega \subset \mathbb{R}^2$ be an open subset. For $u \in L^1(\Omega)$, we then define

$$\mathrm{TV}(u) := \int_\Omega |\nabla u|_2 \, \mathrm{d}x$$

which in turn stands symbolically for

$$\int_\Omega |\nabla u|_2 \, \mathrm{d}x := \sup\Big(\int_\Omega u(x) \operatorname{div} \xi(x) \, \mathrm{d}x : \xi \in C_c^1(\Omega; \mathbb{R}^2), |\xi(x)| \leq 1 \text{ for all } x \in \Omega \Big)$$

where $C_c^1(\Omega; \mathbb{R}^2)$ denotes the set of continuously differentiable $\mathbb{R}^2$-valued functions of compact support in Ω and $|v|_2 := \sqrt{v_1^2 + v_2^2}$ is the Euclidean norm on $\mathbb{R}^2$.

TV was introduced as a penalty function in image processing by Rudin, Osher and Fatemi in [97] and has become quite popular ever since. The original total variation denoising problem proposed in [97] is

$$\mathrm{TV}(u) \to \inf! \quad \text{subject to} \quad \int_\Omega Ku = \int_\Omega Y \quad \text{and} \quad \int_\Omega |Y - Ku|^2 = \sigma^2.$$

In a statistical interpretation of this formulation, the side constraints correspond to the assumption that the noise has zero mean (first constraint) and standard deviation σ (second constraint). It was later proved in [24] that this problem is equivalent to

$$\mathrm{TV}(u) \to \inf! \quad \text{subject to} \quad \|Y - Ku\|^2 \leq \sigma^2$$

under fairly mild assumptions.

For the algorithmic aspects covered by this thesis, we need to formulate a discretized version of total variation for the setting of Example 1.0.1. Considering $u \in \mathbb{R}^{n \times n}$, we define the discrete gradient operator $\nabla u = ((\nabla u)^1_{i,j}, (\nabla u)^2_{i,j})_{i,j}$ for $i, j = 1, \ldots, n$ via the forward difference operator:

$$(\nabla u)^1_{i,j} := \begin{cases} u_{i+1,j} - u_{i,j} & \text{if } i = 1, \ldots, n-1 \\ 0 & \text{if } i = n \end{cases}$$

and

$$(\nabla u)^2_{i,j} := \begin{cases} u_{i,j+1} - u_{i,j} & \text{if } j = 1, \ldots, n-1 \\ 0 & \text{if } j = n \end{cases}.$$

The total variation of u is then defined as

$$\mathrm{TV}(u) = \sum_{i=1}^n \sum_{j=1}^n |(\nabla u)_{i,j}|_2 \, .$$

When closer examining this definition, we find that $\mathrm{TV}(u)$ is likely to become large if u

exhibits large amplitude oscillations. While this is a property it shares with other penalty functions that take into account the differential of u, total variation is particularly suitable for reconstruction of sharp edges in two-dimensional objects (which correspond to jumps in one-dimensional objects) when used as a penalty function. On the other hand, TV performs rather poorly in regions where the intensities of u vary slowly from pixel to pixel. In such regions, TV is known to cause the typical "staircasing artifacts". The smoothness assumption under which TV is usually considered a good choice of a penalty function is hence that the true object consists of piecewise constant values with sharp edges between them. Such objects are frequently referred to as being "blocky".

In the following, we give a brief overview of how to compute TV-penalized least-squares estimators in different situations and state the finite element approach we used in our numerical simulations.

2.4.2 Computation of TV-penalized least-squares estimators

Finding TV-penalized least-squares estimators for the problem at hand, i.e. a solution $\hat{u} \in H_1$ of

$$\frac{1}{2}\|Y - Ku\|^2 + a\,\mathrm{TV}(u) \to \inf!, \tag{2.14}$$

is computationally rather easy in a one-dimensional setting for the case of denoising where $K = \mathrm{Id}$. It was shown in [77] (see also [51]) that the *taut-string algorithm* can then be used as a fast solution method. Unfortunately, this result can be extended neither to higher dimensions nor to nontrivial operators in the model (1.1).

In two dimensions, particularly efficient algorithms have been introduced for the case of denoising, too. We just mention the primal-dual approach in [23] which proves to be remarkably fast. However, just like the one-dimensional approach via taut strings, it cannot be extended to non-trivial operators either and is hence not suitable for our purpose.

For the two-dimensional inverse problem case at hand, for example *fixed point algorithms* for solving the Euler-Lagrange equations of (2.14) could be applied. We refer to [104, Chapter 8] (see also [105] and the references therein) where several algorithms are derived from this ansatz. Nonetheless, we took a different approach in our implementation, namely the one of a *finite element method* which we will describe in the following.

Just as the specific choice of $J = \mathrm{TV}$, we use a finite element method to solve (2.14)

only as an exemplary approach in order to illustrate the performance of our algorithms and compute SMR-estimators in practice. In particular, we do not claim that it should be used preferably whenever TV-penalized least-squares estimators need to be computed. Any method that solves (2.14) would be fine for our purpose and could easily be incorporated into our SMR-methodology as we will see in Chapter 3.

In order to derive the method we used in our experiments, we return to the continuous setting. Due to the nondifferentiability of the Euclidean norm at the origin, the TV functional as defined above causes problems when numerical methods are applied to it. To overcome this difficulty, we will hence use an approximation TV_β rather than the TV functional itself. While there are several options to formulate such an approximation (see e.g. [104, Chapter 8]), we stick to

$$\mathrm{TV}_\beta(u) = \int_\Omega \sqrt{|\nabla u|_2^2 + \beta^2}\,\mathrm{d}x \tag{2.15}$$

for a small positive parameter β. In our numerical experiments, we fixed $\beta = 10^{-4}$. On the basis of this approximation, the problem to be solved can be stated as

$$\hat{u} = \mathrm{argmin}_{u\in H_1} \frac{1}{2}\int_\Omega |Y - Ku|_2\,\mathrm{d}x + a\int_\Omega \sqrt{|\nabla u|_2^2 + \beta^2}\,\mathrm{d}x. \tag{2.16}$$

We will approach this problem by solving the corresponding weak Euler-Lagrange equation, that is

$$\int_\Omega (K\hat{u} - Y)Kv + a\frac{\nabla\hat{u}\nabla v}{\sqrt{|\nabla\hat{u}|_2^2 + \beta^2}}\,\mathrm{d}x = 0 \quad \text{for all } v \in L^2(\Omega). \tag{2.17}$$

This equation will be tackled by means of a finite element method, an approach which we will briefly describe in the following. For a very detailed description of finite element methods, we refer to [108], naming just one of the many textbooks on the subject.

The general idea behind finite element methods can be summarized as follows: instead of solving a variational scheme analytically in the continuous setting, an approximation of the solution in some subset $V_N \subset L^2(\Omega)$ of finite dimension N is computed. To this end, a set of so-called *ansatz functions* $\{\psi_1, \ldots, \psi_N\}$ which form a basis of V_N is chosen and used to represent elements $v \in L^2(\Omega)$ by the approximation $v \simeq \sum_{i=1}^N v_i\psi_i$.

Keeping in mind that the resulting method is supposed to be applied to discrete two-dimensional datasets of size $n \times n$, we choose $N = n^2$ and one bilinear ansatz function

$\psi_i = \psi_{k,l}$ centered at pixel (k, l) for all $(k, l) \in \{1, \dots, n\}^2$. Substituting this into (2.17), we get

$$\sum_{i=1}^{N} K^* K \hat{u}_i \int_\Omega \psi_i \psi_j \, \mathrm{d}x + \sum_{i=1}^{N} \hat{u}_i \int_\Omega a \frac{\nabla \psi_i \nabla \psi_j}{\sqrt{|\nabla \hat{u}|_2^2 + \beta^2}} \, \mathrm{d}x = \sum_{i=1}^{N} K^* Y_i \int_\Omega \psi_i \psi_j \, \mathrm{d}x \tag{2.18}$$

where we set $\hat{u} = \sum_{i=1}^{N} \hat{u}_i \psi_i$. In order to simplify notation in this equation, we define the *mass matrix* $M \in \mathbb{R}^{n^2 \times n^2}$ as

$$M_{i,j} = \int_\Omega \psi_i \psi_j \, \mathrm{d}x \tag{2.19}$$

and the *stiffness matrix* $S[u] \in \mathbb{R}^{n^2 \times n^2}$ as

$$S[u]_{i,j} = \int_\Omega \frac{\nabla \psi_i \nabla \psi_j}{\sqrt{|\nabla u|_2^2 + \beta^2}} \, \mathrm{d}x \tag{2.20}$$

and compute a solution of (2.17) and therefore the desired TV-penalized least-squares estimator $\hat{u}$ via

$$M K^* K \hat{u} + a S[\hat{u}] \hat{u} = M K^* Y. \tag{2.21}$$

This gets done by a fixed point iteration. Starting with some initial u_0, we iteratively set u_{k+1} to a solution of this system where the stiffness matrix is formulated with respect to u_k until we receive a good approximation to the solution. We summarize this approach in pseudocode in Algorithm 1. The integrals needed for M and $S[u]$, respectively, are computed numerically. In our implementation, we use the midpoint rule for this purpose. For a proof of convergence of this fixed point iteration (independent of the initial guess being used), we refer to [33, Theorem 4.1].

2.4.3 Existence of TV-penalized SMR-estimators

Having answered the question of how to tackle the problem of computing TV-penalized least-squares estimators numerically, we conclude this section by returning to the sufficient conditions for the existence of SMR-estimators as stated in Theorem 2.2.2. The TV functional is convex and proper. Moreover, it was proved in [1, Theorem 2.3] to be lower

Algorithm 1 Finite element method for TV-penalized least-squares estimation

Require: $Y \in \mathbb{R}^{n\times n}$ (data); $a > 0$ (regularization parameter); $u_0 \in R^{n\times n}$ (initial guess); $\tau > 0$ (tolerance).

Ensure: $\hat{u}_a[\tau]$ is an approximate solution of (2.21) with tolerance τ in the breaking condition.

1: $u \leftarrow u_0$.
2: Compute M as in (2.19).
3: $S \leftarrow S[u]$ as in (2.20).
4: $R \leftarrow MK^*Y$.
5: **while** $\|R - (MK^*Ku + aSu)\| \geq \tau$ **do**
6: $u \leftarrow \tilde{u}$ where $\tilde{u}$ satisfies

$$(MK^*K + aS)\tilde{u} = R.$$

.
7: $S \leftarrow S[u]$ as in (2.20).
8: **end while**
9: $\hat{u}_a[\tau] \leftarrow u$.

semi-continuous, too. It therefore satisfies Assumption 1.0.5.

As explained in Section 2.2, Assumption 1 in Theorem 2.2.2 does not allow for a straightforward verification for general penalty functions J. The specific choice of $J = \mathrm{TV}$, however, leads to the following lemma taken from [44] that gives a sufficient condition for the assumption to hold.

Lemma 2.4.1. If $J = \mathrm{TV}$ and there exists a $\phi \in \Phi$ such that

$$|\langle K\mathbf{1}, \phi\rangle| > 0, \tag{2.22}$$

then Assumption 1 in Theorem 2.2.2 holds. Here, $\mathbf{1}$ denotes the constant 1-function on H_1.

Proof. See [44, Lemma 4.9]. □

Interpreting this result, we find that the dictionary Φ being used to compute an SMR-estimator has to be chosen in a way that misalignment by a constant in H_1 can still be detected in the image under K by means of Φ. Note that this does not only imply an assumption on the dictionary but on the operator K as well. Fortunately, this assumption appears to be rather weak. In particular, it holds true for the two-dimensional setting of

Example 1.0.1 if we choose Φ in one of the ways proposed in Section 2.3 combined with any of the operators K used in our numerical experiments later in this thesis.

3 Augmented Lagrangian method

The SMR-estimator as derived in the previous chapter exhibits all properties which we formulated as our goals in the introduction. Its statistical interpretation and theoretical background give rise to the question of how to actually compute SMR-estimators in practice. A corresponding method is introduced in this chapter.

The computation of SMR-estimators is challenging as it amounts to solve the constrained optimization problem (2.6) which is extremely large-scale, especially in the two-dimensional setting of Example 1.0.1. In present publications on the subject, the authors therefore circumvent an explicit solution method for this problem. Instead, they approach SMR-estimators via an automatic parameter selection method for a modified version of the estimation scheme (1.6) where the scalar parameter is "localized" to a matrix of the same size as the dataset. This parameter is initialized to a constant large enough to guarantee oversmoothing and then reduced locally until the estimator fulfills the criterion. This technique is employed to one-dimensional datasets in [29] and to two-dimensional datasets in [64] and [100]. Moreover, the methodology presented there is restricted to denoising problems.

In this thesis, however, we present a more rigorous approach to the computation of SMR-estimators. The problem (2.6) is tackled directly by means of a technique from optimization, namely an *Augmented Lagrangian method*. Several refinements of this methodology guarantee computability of the desired estimator despite the problem's vast number of inequality constraints. At the same time, our methodology allows for non-trivial operators in the model (1.1) and can be used for all penalty functions J that meet the assumptions made in Theorem 2.2.2. In addition, the resulting method is appealingly modular and allows for an easy replacement of single components.

The chapter starts out with a description of the method which we use for our purpose in Section 3.1. Apart from stating the algorithmic, we also give a convergence result for our setting. When closer examining the Augmented Lagrangian method, we will find that one intermediate step of it consists in a large-scale quadratic program. This program is tackled in Section 3.2 by means of a projection algorithm. In Section 3.3, we demonstrate how the methodology can be extended to the case of Poisson noise instead of white noise in the data model (1.1) as well as discuss a possible modification of the MR-statistic defined in (2.3). Furthermore, an optional nonnegativity constraint is introduced.

3.1 Decomposition-coordination approach

In what follows, we provide a brief derivation of an Augmented Lagrangian method for the optimization problem (2.6) at hand. Augmented Lagrangian methods were originally introduced in [62] and [91] for equality-constrained problems and extended to inequality constraints in [95]. Ever since then, they have become quite popular in optimization as can be seen by the numerous text books and articles on the subject in which different versions of these methods are applied to diverse problem settings (cf. [2; 6; 43; 49; 65; 66; 69],to name but a few).

In the present situation, we start to approach the computation of SMR-estimators by rewriting (2.6) to the equivalent problem

$$J(u) + G(v) \to \inf! \quad \text{subject to} \quad Ku + v = Y. \tag{3.1}$$

Here, $G : H_2 \to \{0, \infty\}$ denotes the characteristic function of the feasible region $\mathcal{C}$ of (2.6), i.e.

$$G(v) = \begin{cases} 0 & \text{if } v \in \mathcal{C} \\ \infty & \text{else} \end{cases} \tag{3.2}$$

where

$$\mathcal{C} := \left\{ v \in H_2 : T_N(\sigma^{-1}(v)) \leq q_N(\alpha) \right\}. \tag{3.3}$$

For an exhaustive analysis of this technique (which is often referred to as the *decomposition-coordination approach*), see [43, Chapter III] where Lagrangian multipliers are used to solve (3.1). Recall the definition of the *Lagrangian function* L:

$$L(u; v; p) = J(u) + G(v) - \langle p, Ku + v - Y \rangle .$$

The Lagrangian function is modified to the *Augmented Lagrangian function* L_λ by adding a quadratic penalty term to it:

$$L_\lambda(u; v; p) = J(u) + G(v) - \langle p, Ku + v - Y \rangle + \frac{1}{2\lambda} \|Ku + v - Y\|^2 \tag{3.4}$$

for some $\lambda > 0$. An *Augmented Lagrangian method* consists in computing a saddle point

$(\hat{u}, \hat{v}, \hat{p})$ of L_λ, that is

$$L_\lambda(\hat{u}; \hat{v}; p) \leq L_\lambda(\hat{u}; \hat{v}; \hat{p}) \leq L_\lambda(u; v; \hat{p}) \quad \text{for all} \quad (u, v, p) \in H_1 \times H_2 \times H_2.$$

We note that each saddle point $(\hat{u}, \hat{v}, \hat{p})$ of the Augmented Lagrangian L_λ is already a saddle point of L and vice versa. Furthermore, for any such saddle point the pair $(\hat{u}, \hat{v})$ is a solution of (3.1) meaning that $\hat{u}$ is a solution of (2.6) and therefore the desired SMR-estimator. This result was originally proved in [96]; the formulation stated here can be found e.g. in [43, Chapter III, Theorem 2.1].

Sufficient conditions for the existence of saddle points are usually harder to come up with. An (abstract) equivalent condition is formulated in the Karush-Kuhn-Tucker Theorem.

Theorem 3.1.1 (Karush-Kuhn-Tucker)**.** *There exists a saddle point $(\hat{u}, \hat{v}, \hat{p})$ of L_λ if and only if*

$$K\hat{u} + \hat{v} = Y, \quad K^*\hat{p} \in \partial J(\hat{u}) \quad \text{and} \quad \hat{p} \in \partial G(\hat{v}). \tag{3.5}$$

Proof. See [40, Chapter III, Proposition 4.1]. □

Remark 3.1.2. According to [40, Chapter III, Theorem 4.1], condition (3.5) is for instance satisfied if there exists an element $u_0 \in H_1$ such that $J(u_0) < \infty$ and G is continuous at Ku_0. The function G is the indicator function on the nonempty convex polyhedron $\mathcal{C} \subset H_2$ (cf. (3.3)) and is hence continuous on the interior $\mathcal{C}^\circ$ of $\mathcal{C}$. Thus, a sufficient condition for the existence of a saddle point of L_λ can be formulated as follows:

$$\text{There exists } u_0 \in K^{-1}\{\mathcal{C}^\circ\} \cap D(J) \tag{3.6}$$

where $D(J)$ is the domain of J as defined in Assumption 1.0.5. This is often referred to as *Slater's constraint qualification* (cf. [40, Chapter III, Section 5]). Note that in our case $0 \in \mathcal{C}^\circ$ and therefore also $0 \in K^{-1}\{\mathcal{C}^\circ\}$ due to the linearity of K. Hence we find that under the rather weak condition $0 \in D(J)$ which is equivalent to $J(0) < \infty$, existence of a saddle point of (3.4) is already guaranteed.

After these preparations, we are now ready to formulate the Augmented Lagrangian method we use to compute a saddle point of L_λ and thus a solution of (2.6). We present the form described in [43, Chapter III, Section 3.2] in Algorithm 2. It consists in successively performing minimization of L_λ with respect to the first and second variable, respectively, and an explicit update step for maximization with respect to the third variable.

Algorithm 2 Augmented Lagrangian method

Require: $Y \in H_2$ (data); $\lambda > 0$ (step length); $\tau > 0$ (tolerance).

Ensure: $(u[\tau], v[\tau])$ is an approximate solution of (3.1) computed in $k[\tau]$ iteration steps with tolerance τ in the breaking criterion.

1: $u_0 \leftarrow 0_{H_1}$ and $v_0 = p_0 \leftarrow 0_{H_2}$.
2: $r \leftarrow \|Ku_0 + v_0 - Y\|$ and $k \leftarrow 0$.
3: **while** $r > \tau$ **do**
4: $k \leftarrow k + 1$.
5: $v_k \leftarrow \tilde{v}$ where $\tilde{v} \in \mathcal{C}$ satisfies

$$\|\tilde{v} - (Y + \lambda p_{k-1} - Ku_{k-1})\|^2 \leq \|v - (Y + \lambda p_{k-1} - Ku_{k-1})\|^2 \tag{3.7}$$

for all $v \in \mathcal{C}$.
6: $u_k \leftarrow \tilde{u}$ where $\tilde{u}$ satisfies

$$\frac{1}{2}\|K\tilde{u} - (Y + \lambda p_{k-1} - v_k)\|^2 + \lambda J(\tilde{u}) \leq \frac{1}{2}\|Ku - (Y + \lambda p_{k-1} - v_k)\|^2 + \lambda J(u) \tag{3.8}$$

for all $u \in H_1$.
7: $p_k \leftarrow p_{k-1} - (Ku_k + v_k - Y)/\lambda$.
8: $r \leftarrow \max(\|Ku_k + v_k - Y\|, \|K(u_k - u_{k-1})\|)$.
9: **end while**
10: $u[\tau] \leftarrow u_k$ and $v[\tau] \leftarrow v_k$ and $k[\tau] \leftarrow k$.

The method described here reduces (3.1) to the unconstrained least-squares problem (3.8) and the quadratic program (3.7). Note that (3.7) is independent of the choice of the penalty function J, while (3.8) is independent of the multiresolution statistic. This modularity makes the method appealing: replacing J is particularly easy. The same holds for a possible replacement of the statistic T_N in (2.3) under certain restrictions as we will see in Section 3.2 (see also Section 3.3 for an example of a modification of the statistic).

We establish convergence of the Augmented Lagrangian method as stated in Algorithm 2 by the following theorem which is the analogue of [43, Chapter III, Theorem 4.1] adapted to our setting.

Theorem 3.1.3. *Under the same assumptions as made in Theorem 2.2.2, every sequence* $\{(u_k, v_k, p_k)\}_{k\geq 1}$ *that is generated by Algorithm 2 is bounded in* $H_1 \times H_2 \times H_2$ *and every weak cluster point is a saddle point of* L_λ. *Moreover,*

$$\|Ku_k + v_k - Y\| = o(k^{-1/2}) \quad \text{and} \quad \|K(u_k - u_{k-1})\| = o(k^{-1/2}).$$

In particular, Algorithm 2 terminates for each outer tolerance $\tau > 0$ and step length $\lambda > 0$.

Proof. Let us assume that $(\hat{u}, \hat{v}, \hat{p})$ is a saddle point of the Augmented Lagrangian $L_\lambda(u, v, p)$ as defined in (3.4) and that $\{(u_k, v_k, p_k)\}_{k\in\mathbb{N}}$ is a sequence generated by Algorithm 2. Further, we introduce the notation

$$\bar{u}_k := u_k - \hat{u}, \quad \bar{v}_k := v_k - \hat{v} \quad \text{and} \quad \bar{p}_k := p_k - \hat{p}. \tag{3.9}$$

From now on, we assume that $k \geq 1$. By repeating the steps (5.6)-(5.25) in the proof of [43, Chapter III, Theorem 4.1], it follows that

$$\left(\|\bar{p}_{k-1}\|^2 + \lambda^{-2}\|K\bar{u}_{k-1}\|^2\right) - \left(\|\bar{p}_k\|^2 + \lambda^{-2}\|K\bar{u}_k\|^2\right) \geq \lambda^{-2}\left(\|K\bar{u}_k + \bar{v}_k\|^2 + \|K\bar{u}_{k-1} - K\bar{u}_k\|^2\right). \tag{3.10}$$

Summing over k and keeping in mind that $K\bar{u}_k + \bar{v}_k = Ku_k + v_k - Y$ and $K\bar{u}_{k-1} - K\bar{u}_k = Ku_{k-1} - Ku_k$ shows

$$\sum_{k=1}^{\infty} \|Ku_k + v_k - Y\|^2 + \|Ku_{k-1} - Ku_k\|^2 \leq \lambda^2\|\hat{p}\|^2 + \|K\hat{u}\|^2 < \infty \tag{3.11}$$

where we have used that $u_0 = p_0 = 0$. As the sum on the left-hand side is finite, both summands must be asymptotically dominated by k^{-1} which leads to

$$\lim_{k\to\infty} \frac{\|Ku_k + v_k - Y\|}{k^{-1/2}} = \lim_{k\to\infty} \frac{\|Ku_{k-1} - Ku_k\|}{k^{-1/2}} = 0.$$

Using Bachmann-Landau notation, we rewrite this to

$$\|Ku_k + v_k - Y\| = \mathrm{o}(k^{-1/2}) \quad \text{and} \quad \|Ku_{k-1} - Ku_k\| = \mathrm{o}(k^{-1/2}).$$

Furthermore, it follows from (3.10) that $\|\bar{p}_k\|^2 + \lambda^{-2}\|K\bar{u}_k\|^2$ is nonincreasing and hence bounded. This together with $\|Ku_k + v_k - Y\| = \mathrm{o}(k^{-1/2})$ implies that

$$\max(\|Ku_k\|, \|v_k\|, \|p_k\|) = \mathcal{O}(1).$$

Together with the optimality condition for (3.8) this in turn implies that for an arbitrary $u \in D(J)$

$$J(u_k) \leq J(u) + \lambda^{-1}\langle Ku_k + v_k - Y - \lambda p_{k-1}, Ku - Ku_k\rangle = \mathcal{O}(1). \tag{3.12}$$

Summarizing, we find that

$$\sup_{1\leq n\leq N} \mu_{\phi_n}(Ku_k) + J(u_k) \leq \|Ku_k\| + J(u_k) \leq c < \infty$$

for a suitably chosen constant $c \in \mathbb{R}$. Thus, it follows from Assumption 1 of Theorem 2.2.2 that $\{u_k\}_{k\in\mathbb{N}}$ is sequentially weakly compact. Now, let $(\tilde{u}, \tilde{v}, \tilde{p})$ be a weak cluster point of $\{(u_k, v_k, p_k)\}_{k\in\mathbb{N}}$ and recall that $(\hat{u}, \hat{v}, \hat{p})$ was assumed to be a saddle point of the Augmented Lagrangian L_λ. Setting $u = \hat{u}$ in (3.12) thus results in

$$\begin{aligned} J(u_k) \leq J(\hat{u}) + \lambda^{-1} \langle Ku_k + v_k - Y, K\hat{u} - Ku_k\rangle + \langle p_{k-1}, Ku_k - K\hat{u}\rangle \\ = J(\hat{u}) + \langle p_{k-1}, Ku_k - K\hat{u}\rangle + o(k^{-1/2}). \quad (3.13) \end{aligned}$$

Using the relation $K\hat{u} + \hat{v} = Y$ we further find

$$\begin{aligned} \langle p_{k-1}, Ku_k - K\hat{u}\rangle &= \langle p_{k-1}, Ku_k - Y + \hat{v}\rangle \\ &= \langle p_{k-1}, Ku_k + v_k - Y\rangle - \langle p_{k-1}, v_k - \hat{v}\rangle = o(k^{-1/2}) - \langle p_{k-1}, v_k - \hat{v}\rangle . \quad (3.14) \end{aligned}$$

From the definition of v_k in (3.7) it follows that

$$\langle Y + \lambda p_{k-1} - (Ku_{k-1} + v_k), \hat{v} - v_k\rangle \leq 0$$

which in turn implies that

$$\begin{aligned} -\langle p_{k-1}, v_k - \hat{v}\rangle &\leq \lambda^{-1} \langle Y - (Ku_{k-1} + v_k), v_k - \hat{v}\rangle \\ &= \lambda^{-1} \langle Y - (Ku_k + v_k), v_k - \hat{v}\rangle + \lambda^{-1} \langle Ku_k - Ku_{k-1}, v_k - \hat{v}\rangle = o(k^{-1/2}). \quad (3.15) \end{aligned}$$

Combining (3.13), (3.14) and (3.15) gives

$$\limsup_{k\to\infty} J(u_k) \leq J(\hat{u}).$$

Now, choose a subsequence $\left\{u_{\rho(k)}\right\}_{k\in\mathbb{N}}$ such that $u_{\rho(k)} \rightharpoonup \tilde{u}$. Then, it follows from the lower semi-continuity of J and the previous estimate that

$$J(\tilde{u}) \leq \liminf_{k\to\infty} J(u_{\rho(k)}) \leq J(\hat{u}).$$

Moreover, we have that $v_{\rho(k)} \in \mathcal{C}$ for all $k \in \mathbb{N}$. Since $\mathcal{C}$ is closed and convex it is also weakly closed and we conclude that $\tilde{v} \in \mathcal{C}$. Since $K\tilde{u} + \tilde{v} = Y$ this shows that $(\tilde{u}, \tilde{v})$ solves (3.1) and thus $J(\tilde{u}) = J(\hat{u})$. □

For a given tolerance $\tau > 0$, Theorem 3.1.3 implies that Algorithm 2 terminates and outputs an approximate solution $(u[\tau], v[\tau])$ of (3.1). However, the breaking condition in Algorithm 2 merely guarantees that the linear constraint in (3.1) is approximated sufficiently well. Moreover, we know from construction that $v[\tau] \in \mathcal{C}$ which implies $G(v[\tau]) = 0$. All in all, it remains to estimate the value of $J(u[\tau])$. This is done in the following corollary.

Corollary 3.1.4. Let $(\hat{u}, \hat{v}, \hat{p}) \in H_1 \times H_2 \times H_2$ be an arbitrary saddle point of L_λ. Then,

$$J(u[\tau]) - J(\hat{u}) \leq \left(\frac{\tau + \|K\hat{u}\|}{\lambda} + 2\|\hat{p}\| \right) \tau$$

for all $\tau > 0$.

Proof. We again use the notation introduced in (3.9). Observe that the estimate in (3.10) implies that the sequence $\|\bar{p}_k\|^2 + \lambda^{-2}\|K\bar{u}_k\|^2$ is nonincreasing. Since $u_0 = p_0 = 0$, we have that

$$\|p_k\| \leq 2\|\hat{p}\| + \lambda^{-1}\|K\hat{u}\|.$$

Now assume that $\tau > 0$ and that $k = k[\tau]$ is such that

$$\max(\|Ku_k + v_k - Y\|, \|Ku_{k-1} - Ku_k\|) \leq \tau.$$

Then, it follows from (3.13) that

$$J(u_k) \leq J(\hat{u}) + \lambda^{-1}\tau^2 + (2\|\hat{p}\| + \lambda^{-1}\|K\hat{u}\|)\tau$$

which proves the assertion. □

The results in Theorem 3.1.3 and Corollary 3.1.4 show that the accuracy of the approximate solution $(u[\tau], v[\tau])$ depends linearly on τ. Furthermore, the choice of the step length λ does not affect the asymptotic behaviour of the algorithm according to Theorem 3.1.3 but influences its accuracy as well: it follows from the definition of L_λ in (3.4) and Corollary 3.1.4 that a small value of λ fosters the linear constraint in (3.1) but may result in slow decay of the objective function J. On the other hand, a large value of λ yields additional

precision of the result but also leads to longer computation times due to more time being spent in (3.7) as the linear constraints are somewhat neglected.

In order to overcome this difficulty and choose λ in a way that balances this trade-off between the runtime of the method and the accuracy of its output, we consider the function

$$E(\lambda) = \left(\frac{\tau + \|K\hat{u}\|}{\lambda} + 2\,\|\hat{p}\| \right) \tau$$

which gives the upper bound on the error in Corollary 3.1.4 dependent on λ. We propose to choose λ according to the point of maximal curvature of the graph of E. Setting $C := (\tau + \|K\hat{u}\|)\tau$, this curvature is given by

$$\kappa_E(\lambda) := \left| \frac{E''(\lambda)}{(1 + E'(\lambda)^2)^{3/2}} \right| = \frac{2C\lambda^{-3}}{(1 + C^2\lambda^{-4})^{3/2}}.$$

By simple calculus, we find that κ_E takes its maximum value for $\lambda_0 = \sqrt{C}$. As $K\hat{u}$ and therefore also C are typically unknown in practical applications, we further propose to use $\|Y\|$ as an approximation of $\|K\hat{u}\|$. This is motivated by the fact that the observation Y is perturbed by *white* noise. In summary, we choose

$$\lambda = \sqrt{(\tau + \|Y\|)\tau}.$$

We close this section by comparing the Augmented Lagrangian method and the local approach of adaptive parameter selection as presented in [64] on an algorithmic level. In contrast to the algorithm stated in [64], the Augmented Lagrangian method solves a penalized least-square problem with a global (i.e. scalar) regularization parameter only, namely λ in (3.8). Local adaptivity is established by modifying the input $Y + \lambda p_{k-1} - v_k$ to the least-squares problem variably in each iteration step rather than by locally modifying the actual parameter. The additional term $\lambda p_{k-1} - v_k$ in the input of (3.8) influences the resulting penalized least-squares estimator in a way that it locally "corrects" areas which were considered as badly estimated according to the MR-criterion before by adjusting Y in these regions. In Section 4.1, we give an illustration of how this correction is carried out depending on the choice of the step length λ. All in all, complications that occur when discretizing the regularization parameter while keeping the input constant at the actual data Y are avoided. Indeed, the Augmented Lagrangian method therefore leads to better results than can be achieved by means of the local parameter selection presented in [64] as indicated by the numerical results presented in Section 4.1.

3.2 The quadratic program

Algorithm 2 as derived in the previous section provides a method to compute the desired SMR-estimators. The method decomposes the original optimization problem (2.6) into the unconstrained optimization problem (3.8) and the quadratic program (3.7).

Note that a solution method for (3.8) depends largely upon the penalty function J. As it is not the goal of this thesis to advertise any specific choice of J but to demonstrate a data-driven and statistically sound way of choosing the amount of smoothing needed, we abstain from discussing such solution methods any further. In our numerical simulations, however, we stick to the finite element method derived in Section 2.4 for the choice of $J = \mathrm{TV}$. Nonetheless, other methods might still be more suitable for the specific choice of $J = \mathrm{TV}$, too. We remark once more that such methods could easily be incorporated in Algorithm 2.

The only step in Algorithm 2, however, in which the MR-criterion comes into play is the optimization problem (3.7). By providing a method to efficiently solve it, the statistical part of SMR-estimation within the Augmented Lagrangian framework would already be covered. Observe that the feasible region $\mathcal{C}$ of (3.7) as defined in (3.3) can be written as

$$\mathcal{C} = \left\{ v \in H_2 : \frac{\langle v, \phi_i \rangle}{\|\phi_i\|} \leq c_i \text{ and } -\frac{\langle v, \phi_i \rangle}{\|\phi_i\|} \leq c_i \text{ for all } i = 1, \ldots, N \right\}$$

where $c_i := q_N(\alpha) + f(\|\phi_i\|)$. From this formulation, it becomes obvious that $\mathcal{C}$ is a polyhedron and (3.7) is in fact a quadratic program with an overall number of $2N$ linear inequality constraints. As N is usually large in practical applications (in particular in the discrete two-dimensional setting of Example 1.0.1 where Φ is chosen in one of the ways suggested in Section 2.3), the program is likely to be extremely large-scale and hence numerically challenging. How to solve it is the subject of this section.

3.2.1 The projection problem and Dykstra's algorithm

In order to tackle the problem

$$\|v - (Y + \lambda p_{k-1} - K u_{k-1})\|^2 \to \inf! \quad \text{subject to } v \in \mathcal{C}, \tag{3.16}$$

we first tried to use an interior point method (see e.g. [86, Chapter 14]). In fact, we used the C++ software package OOQP ("Object-Oriented software for Quadratic Programming",

[47]) and adapted it to our situation. The method implemented there consists in a Mehrotra predictor-corrector algorithm as introduced in [81] with additional Gondzio projections as suggested in [50]. Unfortunately, this approach was not successful. The resulting method failed to compute solutions of (3.16) for comparatively small two-dimensional datasets like $n = 256$ already. For smaller n, solutions could be computed but the runtime was far from practical.

The reason for the interior point method to fail is the vast number of inequalities in the side constraints of (3.16) in the two-dimensional setting at hand. For an image of size 256×256 and an all squares partitioning $\mathcal{P}_A$ including scales from 1 through to 25, for example, we already get an overall number of $2,979,400$ side constraints. In view of this huge number, approaching the problem via an optimization method that covers a wide range of problems simply seems to be too general to be efficient. For this reason, we need to find a method which is better matched for the problem (3.16).

To this end, we closer examine the problem, focusing on the side constraints first. Due to the supremum taken in the definition (2.3) of the MR-statistic T_N, the inequality

$$\frac{|\langle v, \phi_i \rangle|}{\|\phi_i\|} - f(\|\phi_i\|) \leq q_N(\alpha)$$

holds for all $v \in \mathcal{C}$, for all $i = 1, \ldots, N$. In other words, v is an element of all N feasible regions that would be defined by a dictionary that only consists of the single test function ϕ_i. This specific structure of (3.16) can be exploited in order to establish computability of a solution despite the vast number of inequality constraints involved. We rewrite the definition (3.3) of the feasible region $\mathcal{C}$ to

$$\mathcal{C} = \bigcap_{i=1}^{N} C_i \quad \text{where} \quad C_i = \{v \in H_2 : \mu_{\phi_i}(v) \leq c_i\}\,. \tag{3.17}$$

According to this representation, we may formulate the quadratic program (3.16) as the following projection problem:

$$\|v - Y_k\|^2 \to \inf! \quad \text{subject to} \quad v \in \bigcap_{i=1}^{N} C_i \tag{3.18}$$

where $Y_k := Y + \lambda p_{k-1} - K u_{k-1}$. It is straightforward to show that all C_i are closed and convex. We have therefore derived a formulation in which the computation of a solution v_k of (3.16) amounts to compute the *projection of Y_k onto the intersection $\mathcal{C}$ of closed and*

convex sets C_i. This situation is illustrated in Figure 3.2.1.

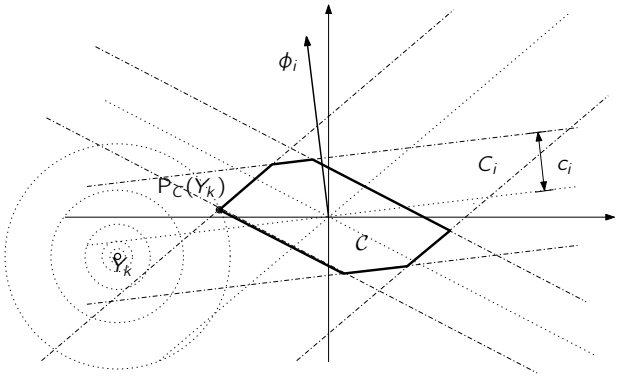

Figure 3.1: The admissible set $\mathcal{C}$ as intersection of the sets C_i as in (3.17).

In order to cope with this problem, we apply Dykstra's algorithm as introduced in [11] (not to be confused with Dijkstra's algorithm from graph theory). This algorithm takes an element $v \in H_2$ and closed and convex sets $D_1, \ldots, D_M \subset H_2$ as arguments. It then creates a sequence converging to the projection of v onto the intersection $\bigcap_{m=1}^{M} D_m$ by successively performing projections onto individual D_m's. An exact version of the algorithm is noted in pseudocode in Algorithm 3. There, $P_D(\cdot)$ denotes the projection onto $D \subset H_2$ and $S_D = P_D - \mathrm{Id}$ is the corresponding projection step.

Closer examination of the algorithm shows that for each D_m, the projection step taken in the last iteration for that very same D_m gets reversed on $h_{k,m-1}$. The resulting element is then projected onto D_m giving the updated iterate $h_{k,m}$. The corresponding projection step gets saved in $Q_{k,m}$ as it will be reversed when the algorithm reaches D_m again in the next cycle. All in all, this procedure is not very intuitive at first glance, but it nevertheless works as expected according to the convergence result presented in Theorem 3.2.1 below. We provide an outline of the background of the algorithm.

Dykstra's algorithm is based on a modification of the classical alternating projection method first established in [106]. It was introduced to projections onto the intersection of closed and convex cones in $\mathbb{R}^n$ in [38] and generalized to a Hilbert space setting in [11]. The algorithm was re-discovered in [52] where it was derived in a primal-dual framework, see also [46]. This different approach to the algorithm leads to a more natural derivation of it and a resulting simpler proof of convergence.

An interesting approach to solving the best approximation problem with Bregman projec-

Algorithm 3 Dykstra's algorithm

Require: $h \in H_2$ (element to be projected); $D_1, \ldots, D_M \subset H_2$ (closed and convex sets).
Ensure: $\{h_k\}_{k\in\mathbb{N}}$ is a sequence that converges strongly to $P_{\mathcal{D}}(h)$ where $\mathcal{D} = \bigcap_{m=1,\ldots,M} D_m$.

$h_{0,0} \leftarrow h$
for $m = 1$ to M **do**
  $h_{0,m} \leftarrow P_{D_m}(h_{0,m-1})$
  $Q_{0,m} \leftarrow S_{D_m}(h_{0,m-1})$
end for
$h_1 \leftarrow h_{0,M}$
$k \leftarrow 1$
for $k \geq 1$ **do**
  $h_{k,0} \leftarrow h_k$
  for $m = 1$ to M **do**
    $h_{k,m} \leftarrow P_{D_m}(h_{k,m-1} - Q_{k-1,m})$
    $Q_{k,m} \leftarrow S_{D_m}(h_{k,m-1} - Q_{k-1,m})$
  end for
  $h_{k+1} \leftarrow h_{k,M}$
  $k \leftarrow k + 1$
end for

tions by combining them with Dykstra's algorithm was proposed in [22]. For further analysis and generalizations of this approach, we also refer to [5] and [12]. Another modification of Dykstra's algorithm that deals with projections onto half-spaces was introduced in [14].

Applications of Dykstra's algorithm include signal recovery (cf. [25]) as well as finding a nearest diagonally dominant or symmetric matrix as described in [42; 82; 83; 92]. A rather general approach to applications of projection algorithms in image reconstruction was taken in [21].

For the polyhedral case at hand in our application, Dykstra's algorithm was proved in [68] to coincide with Hildreth's method introduced in [63]. Therefore, the following theorem taken from [32] that establishes linear convergence of Dykstra's algorithm in the polyhedral case can be seen as a different formulation of the theorem proved in [67] for Hildreth's method. The theorem as stated here was further improved in [88] and [107] where estimates of the constants on the right-hand side were given.

Theorem 3.2.1. *Let $\{h_k\}_{k\in\mathbb{N}}$ be the sequence generated by Dykstra's algorithm and $P_{\mathcal{D}}(h)$ be the projection of the input h onto $\mathcal{D}$. Then there exist constants $\rho > 0$ and $0 \leq c < 1$ such that*

$$\|h_k - P_{\mathcal{D}}(h)\| \leq \rho c^k$$

for all $k \in \mathbb{N}$.

Proof. See [32, Theorem 3.8]. □

Remark 3.2.2. The constant c on the right-hand side increases with the number M of convex sets which intersection form the set $\mathcal{D}$ that h is to be projected on. This is not surprising as the complexity of $\mathcal{D}$ increases with the number of sets it is formed from. The convergence rate therefore improves with decreasing M. For further details and estimates for the constants ρ and c, we again refer to [88] and [107].

Note that Dykstra's algorithm needs to be modified to an inexact version before applying it in practice. As the desired projection is in general achieved asymptotically only, some notion of a breaking criterion needs to be formulated in order to stop the algorithm once a sufficiently exact solution was computed. In our application, we use the criterion

$$T_N(h_k) - q_N(\alpha) \leq \tau$$

where $\tau > 0$ is some given tolerance. In other words, we stop the algorithm as soon as the MR-statistic of the current iterate is sufficiently close to the critical value. Since Dykstra's algorithm always approaches the projection of h onto $\mathcal{D}$ from outside of $\mathcal{D}$, measuring the distance from the current iterate to the feasible region and using it for a stopping criterion is a natural approach.

In general applications, however, such a measure might no be available. We refer to [7] for a robust criterion which is only based on previous iterates and projection steps of the algorithm. There, the authors especially prove that stopping the algorithm once the difference between two iterates h_k and h_{k-1} gets sufficiently small does not necessarily result in trustable solutions by giving a counter-example. As the stopping criterion suggested in [7] did not result in additional precision in the context of the Augmented Lagrangian method in our experiments though, we decided to stick to the criterion formulated above.

3.2.2 Increasing efficiency

Note that application of Dykstra's algorithm is particularly appealing if the projections P_{D_m} can be easily computed or even stated explicitly, as it is the case within the Augmented

Lagrangian framework. This fact turns the algorithm into the method of choice for the problem at hand. A first approach to use Dykstra's algorithm to solve (3.18) is to set $M = N$ and $D_m = C_m$ for all $m = 1, \dots, M$. The required projection step onto a fixed C_m is then given by

$$S_{C_m}(h) = \begin{cases} -\operatorname{sign}\left(\langle h, \phi_m \rangle\right) \frac{|\langle h, \phi_m \rangle| - c_m}{\|\phi_m\|} & \text{if } \mu_{\phi_m}(h) > c_m \\ 0 & \text{else} \end{cases}. \tag{3.19}$$

In view of Remark 3.2.2, however, it is clearly desirable to decrease the number M of convex sets that enter Dykstra's algorithm. In order to do so, we take a more sophisticated approach than the one just presented. We subdivide the index set $\{1, \dots, N\}$ into $I_1, \dots, I_M$ where

$$\langle \phi_i, \phi_j \rangle = 0 \quad \text{for all} \quad i, j \in I_m, \quad \text{for all} \quad m = 1, \dots, M, \tag{3.20}$$

and regroup $C_1, \dots, C_N$ into $D_1, \dots, D_M$ via

$$D_m = \bigcap_{i \in I_m} C_i. \tag{3.21}$$

Due to the pairwise orthogonality of $\{\phi_i : i \in I_m\}$ for all $m = 1, \dots, M$, the projection step from some h onto each D_m can still be computed easily: Identify the set

$$V_m = \{i \in I_m : \mu_{\phi_i}(h) > c_i\}$$

of indices in I_m for which h violates the side condition of (3.18) and set

$$S_{D_m}(h) = -\sum_{i \in V_m} \operatorname{sign}\left(\langle h, \phi_i \rangle\right) \frac{|\langle h, \phi_i \rangle| - c_i}{\|\phi_i\|}. \tag{3.22}$$

To keep M small, we choose $I_1 \subset \{1, \dots, N\}$ as the biggest set such that $\langle \phi_i, \phi_j \rangle = 0$ holds for all $i, j \in I_1$. We then choose $I_2 \subset \{1, \dots, N\} \setminus I_1$ with the same property and continue in this way until all indices are utilized. While this procedure does not necessarily result into M being minimal with the desired property, it still yields a distinct reduction of N.

We substantiate this approach for the discrete two-dimensional setting of Example 1.0.1 and dictionaries consisting of characteristic functions of the squares in the partitionings $\mathcal{P}_A$ and $\mathcal{P}_D$ as described in Section 2.3. In what follows, we will sometimes identify a test

function $\phi_i = n^{-2}\chi_{S_i}$ with the square S_i in a slight abuse of notation.

First, note that $\langle \phi_i, \phi_j \rangle = 0$ in this setting means that the corresponding squares S_i and S_j are disjoint. Consequently, if a dyadic squares partition $\mathcal{P}_D$ is used, all squares of the same size will be grouped into one D_m as they are disjoint by construction of the partitioning. For a dataset $Y \in \mathbb{R}^{n\times n}$, we will hence get an amount of $M = \lceil \log_2 n \rceil$ sets that enter Algorithm 3 in this case. This rather small number of sets makes the algorithm particularly fast if combined with a dyadic squares partitioning.

If, on the other hand, an all squares partitioning $\mathcal{P}_A$ is used, we proceed as follows: we loop over all scales involved in increasing order. For each scale s, we start out by grouping $[1, s] \times [1, s]$ and all consecutive squares that fit into the image domain into one system $D_{s,(1,1)}$. Next, we misalign $[1, s] \times [1, s]$ by one pixel and form the corresponding $D_{s,(1,2)}$. Iterating this procedure, we hence loop over all

$$(k, l) \in \{1, \dots, \min(n - s + 1, s)\}^2$$

and form the system

$$D_{s,(k,l)} = \{S \in \mathcal{P}_A : S = [is + k, (i + 1)s + k - 1] \times [js + l, (j + 1)s + l - 1] \\ \text{and } i = 0, \dots, \lfloor (n - k)/s \rfloor \text{ and } j = 0, \dots, \lfloor (n - l)/s \rfloor\}$$

for each such pair. The minimum taken in the limit of k and l corresponds to distinguishing whether or not s is bigger than $n/2$. By this restriction, empty systems are avoided.

Indeed, the procedure just described severely reduces the number of sets that enter Algorithm 3. If all possible scales $\{1, \dots, n\}$ are taken into account for $\mathcal{P}_A$, we will get an overall number of

$$N = N(n) = \sum_{j=1}^{n} (n - j + 1)^2 = \sum_{j=1}^{n} j^2 = \frac{n(n + 1)(2n + 1)}{6}$$

squares in $\mathcal{P}_A$. Put differently, the admissible set $\mathcal{C}$ in (3.17) is the intersection of $N(n)$ sets in H_2. For a digital image with resolution 256×256, for example, this results in a vast number of $N \sim 5 \cdot 10^7$ squares in (3.16).

If we group independent side-conditions, that is side-conditions corresponding to squares in X with empty intersection as described above, however, we will receive $\min(n-s+1, s)^2$

sets on each scale s and therefore an overall of

$$M = M(n) = \sum_{j=1}^{n} \min(n-j+1, j)^2 = \frac{n(n+1)(2n+1)}{6} - \frac{(n+1)n^2}{4}.$$

In the 256×256 example, the number of sets is thereby reduced to $M \sim 10^7$. While this number is still too big in practical applications, prior information about the true object u^* might be used in order to allow only D_m of specific sizes to enter the algorithm. By such a restriction, M can be ensured to be of reasonable size. In fact, we restricted our partitioning as used in our simulations to all squares of side lengths from 1 through to 25 pixels. This further reduced the number of sets to $M = 5,525$ for 256×256 images and guarantees computability of the SMR-estimator by means of our methodology within reasonable time while covering all $2,979,400$ inequality constraints mentioned earlier. We remark that the restriction to the scales named above is justified by the fact that most of the features we expect our test objects to exhibit occur on the scales taken into account. Leaving out the computationally involved larger scales does therefore not result in considerably worse reconstruction quality.

3.2.3 Implementation

Having demonstrated how to reduce the number of sets that enter Dykstra's algorithm to a reasonable level, we now add a few remarks about some details of our implementation of the algorithm. First of all, note that the projection steps in (3.22) are constant over a fixed square $S \in \mathcal{P}$. For this reason, only one scalar per square needs to be saved and subtracted pointwise when the corresponding projection gets reversed in the next cycle. By proceeding this way, a lot of memory can be saved. When using an all squares partitioning $\mathcal{P}_A$, we identify each $D_m = D_{s,(k,l)} \in \mathcal{P}_A$ by its scale s and the pair (k,l) as described above. For each such triple, we save a matrix $\Pi_{k,l,s} \in \mathbb{R}^{\lfloor (n-k)/s \rfloor \times \lfloor (n-l)/s \rfloor}$ holding the projection steps for all $S \in D_m$. As only the projection performed in the last cycle needs to be saved, these matrices can be overwritten in each cycle of Dykstra's algorithm.

A second remark is concerned with the subtraction of the projection steps taken in the previous iteration as needed in line 11 of Algorithm 3. In our implementation, we made sure that this subtraction is performed only if an actual projection took place for the corresponding set in the previous cycle, i.e. the corresponding $\Pi_{k,l,s}$ is not all-zero. We therefore save a vector of booleans of length M (which is the number of systems D_m that $\mathcal{P}_A$

is subdivided into) and keep track of the $D_{s,(k,l)}$ for which a nontrivial projection was carried out in the last iteration. This prevents from looping over all-zero matrices for nothing and saves runtime, especially in later iterations when the iterate is already an element of most of the D_m's and hence many projections are trivial.

All in all, we paid close attention to carefully translate the methodology of this chapter into program code, but do not claim that our implementation is particularly efficient in all details. Certainly, there is still room for improvements. We hence abstain from more detailed simulations to test the runtime of our program and profiling it.

We conclude this section by pointing out that an employment of parallel versions of Dykstra's algorithm would lead to a drastic acceleration of it in practice. For simultaneous versions of the algorithm, we refer to [53] (in $\mathbb{R}^n$), [46] (in general Hilbert spaces) and [26]. By using multiple processors or a graphics processing unit (GPU), implementations of these algorithms could save a huge amount of runtime in comparison to the sequential version stated here in Algorithm 3. In our experiments, however, we did not push this idea any further as our main goal was to deliver a proof of concept rather than a perfectly tuned implementation. The runtime of our framework including the improvements in the algorithmic as given in this section is within reason for moderately large images, too, and hence allows for experimenting with our methods satisfactorily.

3.3 Some Extensions

The approach of combining Augmented Lagrangian techniques with Dykstra's algorithm for the solution of (3.1) proves to be remarkably versatile. In particular, substantial modifications in the model (1.1) on the one hand and in the MR-statistic (2.3) on the other hand are possible without changing the algorithmic methodology. We indicate this by three examples with special appeal for applications.

First, we demonstrate how modifications of the original MR-statistic can be incorporated into our Augmented Lagrangian framework in Subsection 3.3.1. Afterwards, an alteration of our methodology to handle an alternative data model in which the noise is assumed to be Poisson distributed is suggested. Finally, we provide a modification of Algorithm 2 which allows for imposing an additional nonnegativity constraint in Subsection 3.3.3.

3.3.1 Transformed Residuals

In some employments (as e.g. image denoising, cf. Subsection 4.1.4), it is useful to study transformations of the residual $r = \sigma^{-1}(Y - K\hat{u})$ where $\hat{u}$ is some estimator of the true solution of $Ku = g$ rather than r itself. To this end, we consider a given transformation $\Lambda : H_2 \to H_2$ and introduce

$$\mu_{\Lambda,\phi}(r) = \frac{|\langle \Lambda(r), \phi \rangle|}{\|\phi\|}$$

as a modified version of the average function (2.2). Here, we require that Λ is continuous and that

$$\Lambda_\phi : v \mapsto \langle \Lambda(v), \phi \rangle$$

is convex for all $\phi \in \Phi$. Then, the feasible sets (3.17) in the projection problem (3.18) are replaced by

$$\mathcal{C}_\Lambda = \bigcap_{i=1}^{N} C_{\Lambda,i} \quad \text{where} \quad C_{\Lambda,i} = \{v \in H_2 : \mu_{\Lambda,\phi_i}(v) \leq c_i\}. \tag{3.23}$$

Due to the convexity and continuity assumptions on Λ, all $C_{\Lambda,i}$ are closed and convex. We can therefore still apply Dykstra's algorithm in order to compute the projections needed in the Augmented Lagrangian method. One should make sure though that a particular choice

of Λ still allows for an explicit statement of the projection onto all single $C_{\Lambda,i}$'s (like (3.22) for $\Lambda = \mathrm{Id}$) as numerous of these projections are performed in Dykstra's Algorithm 2. The method is hence likely to become computationally infeasible otherwise. We give an example of a function Λ that enables such explicit projections.

Example 3.3.1 (cont. Example 1.0.1)**.** Let $H_1 = H_2$ and X be as in Example 1.0.1. We consider the mapping defined by

$$(\Lambda(v))(x) = v(x)^2 \quad \text{for all} \quad x \in X. \tag{3.24}$$

Then, Λ is continuous and the mappings $v \mapsto \Lambda_\phi(v) = \langle \Lambda(v), \phi \rangle$ are convex for all $\phi \in H_2$.

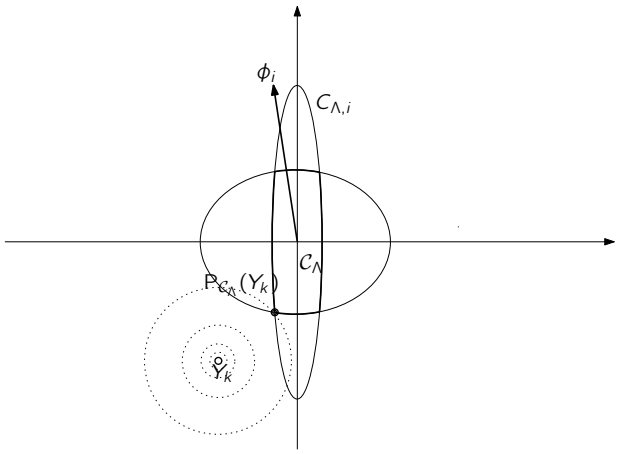

Figure 3.2: The admissible set $\mathcal{C}_\Lambda$ as intersection of the sets $C_{\Lambda,i}$ as in (3.23).

Thus, the admissible set $\mathcal{C}_\Lambda$ is the intersection of the elliptic cylinders $C_{\Lambda,n}$ in the (finite-dimensional) space H_2 (cf. Figure 3.2). We finally note that for the case where Φ consists of characteristic functions of measurable subsets of X, the sets $C_{\Lambda,n}$ take the form of circular cylinders, and the projections P_{D_m} can be computed explicitly in a similar fashion as in (3.22). Just observe that in contrast to the projections given there, the resulting projection steps are not necessarily constant over a fixed square $S \in \mathcal{P}$ any more when dealing with transformed residuals $\Lambda(r)$. As a consequence, not only one scalar per square $S \in \mathcal{P}$ needs to be saved (as described in Subsection 3.2.3), but a matrix of the same size as S. This increases the amount of memory needed by the algorithm.

3.3.2 Poisson Noise

Some applications give rise to the question whether SMR-estimation might be extended to situations in which the underlying data model differs from (1.1). Especially a perturbation with Poisson noise instead of a white noise process is of interest. Let us therefore assume that H_2 is as in Example 1.0.1 and that $(Ku^*)_{i,j} \in \mathbb{N}$ for all $(i,j) \in X$ and $u \in H_1$. We consider the model

$$Y_{i,j} \sim \text{Pois}((Ku^*)_{i,j}) \quad \text{for all} \quad (i,j) \in X. \tag{3.25}$$

In order to apply our methodology to this situation, we make use of a variance-stabilizing transformation. For sufficiently large values of $(Ku^*)_{i,j}$ the central limit theorem states that

$$Z_{i,j} := \frac{Y_{i,j} - (Ku^*)_{i,j}}{\sqrt{(Ku^*)_{i,j}}} \tag{3.26}$$

is approximately standard normally distributed.

Note that the true object u^* is usually not accessible in practical applications. Hence we cannot apply the transformation as stated above directly. When performing Algorithm 2, we therefore use u_{k-1} as an approximation of u^* in iteration step k. This ansatz leaves the Augmented Lagrangian framework unchanged while Dykstra's Algorithm 3 needs to be slightly modified. Instead of projecting onto the intersection $\mathcal{C}$ of the sets C_n as described in (3.17), we now project in the k-th step of Algorithm 2 onto

$$\mathcal{C}_P[k] = \bigcap_{i=1}^{N} C_{P,i}[k] \quad \text{where} \quad C_{P,i}[k] = \left\{ v \in H_2 \ : \ \mu_{\phi_i}\left(v / \sqrt{K u_{k-1}} \right) \leq c_i \right\}$$

with a pointwise division by the square root of Ku_{k-1}. Note that all $C_{P,i}[k]$ are closed and convex. Furthermore, projections onto single $C_{P,i}[k]$'s can still be stated explicitly. Dykstra's algorithm thus remains a feasible method to compute the desired projections for these modified sets, too.

We note that with this modification, the projection problem (3.7) changes in each iteration step of Algorithm 2 and Theorem 3.1.3 does not hold anymore. So far, we have not come up with a similar convergence analysis. However, it follows from the proof of Theorem 3.1.3 that each stationary point $(\hat{u}, \hat{v}, \hat{p})$ is a saddle point of the Augmented Lagrangian function $L_\lambda(u, v, p)$ in (3.4), where G is the indicator function on the set

$$\mathcal{C}_P = \bigcap_{i=1}^{N} C_{P,i} \quad \text{where} \quad C_{P,i} = \left\{ v \in H_2 \ : \ \mu_{\phi_i}(v/\sqrt{K\hat{u}}) \leq c_i \right\}.$$

Put differently, $\hat{u}$ is a solution of

$$J(u) \to \text{inf!} \quad \text{subject to} \quad \mu_{\phi_i}\left(\frac{Y - Ku}{\sqrt{Ku}}\right) \leq c_i, \quad \text{for all } 1 \leq i \leq N.$$

Furthermore, we stress that the assumption that $(Ku^*)_{i,j}$ is sufficiently large is crucial for the transformation (3.26) to work as expected. If this value drops below a level of about 10, the transformed variable $Z_{i,j}$ is far from being standard normally distributed. Nonetheless, the method presented here still works surprisingly well even if some observations $Y_{i,j}$ exhibit lower intensities, see Section 4.2.

3.3.3 Nonnegativity

In some applications, the true object u^* is a priori known to exhibit nonnegative values only. Hence, when dealing with such problems, one would like to constrain the SMR-estimator to be pointwise nonnegative, too. We show how our method can be extended to allow for this additional constraint. Throughout this subsection, $u \geq 0$ for some $u \in \mathbb{R}^{n \times n}$ will denote pointwise nonnegativity, i.e. that $u_{i,j} \geq 0$ for all $i, j = 1, \ldots, n$.

In order to introduce the modified version of Algorithm 2, we proceed analogously to Section 3.1 where the original Augmented Lagrangian method was derived. We start out by stating the extended version of the optimization problem (2.6):

$$J(u) \to \text{inf!} \quad \text{subject to} \quad T_N(\sigma^{-1}(Y - Ku)) \leq q_N(\alpha) \quad \text{and} \quad u \geq 0. \tag{3.27}$$

Taking the decomposition-coordination approach, the equivalent equality constrained problem (i.e. the analogue of (3.1)) is given by

$$J(u) + G(v) + H(w) \to \text{inf!} \quad \text{subject to} \quad Ku + v = Y \quad \text{and} \quad u = w \tag{3.28}$$

with G as in (3.2) and

$$H(w) = \begin{cases} 0 & \text{if } w \geq 0 \\ \infty & \text{else} \end{cases}.$$

Next, we state the Augmented Lagrangian function (3.4) with the additional constraint added:

$$\begin{aligned} L_{\lambda,\nu}(u; v; w; p; q) = J(u) + G(v) + H(w) - \langle p, Ku + v - Y \rangle \\ - \langle q, u - w \rangle + \frac{1}{2\lambda} \|Ku + v - Y\|^2 + \frac{1}{2\nu} \|u - w\|^2 \end{aligned}$$

for some $\lambda, \nu > 0$. After these preparations we are now ready to state the modified version of Algorithm 2 that additionally imposes nonnegativity in Algorithm 4. Just as the unmodified version, it aims at computing a saddle point of the Augmented Lagrangian function by alternately minimizing and maximizing with respect to the different variables.

Note that the computational effort per iteration step is not significantly increased if nonnegativity is additionally imposed. Updating w_k in (3.30) amounts to simply setting negative values of $(u_k - \nu q_{k-1})$ to zero, while q_k is updated explicitly. Moreover, the modification of the input in (3.29) does not lead to a problem that is harder to solve than (3.8). Nonetheless, Algorithm 4 is likely to perform a bigger number of iteration steps than Algorithm 2 when started on the same input which results in a longer overall runtime.

Remark 3.3.2. While we focused on nonnegativity here, a generalization to a constraint of the form $u \geq l$ for some lower bound l in (3.27) is straightforward. In addition, u could in a similar way be constrained to exhibit values in a certain interval only by also imposing an upper bound on it.

Algorithm 4 Nonnegatively Constrained Augmented Lagrangian Method

Require: $Y \in H_2$ (data); $\lambda > 0$, $\nu > 0$ (step lengths); $\tau \geq 0$ (tolerance).

Ensure: $(u[\tau], v[\tau], w[\tau])$ is an approximate solution of (3.28) computed in $k[\tau]$ iteration steps with tolerance τ in the breaking criterion.

1: $u_0 = w_0 \leftarrow 0_{H_1}$ and $v_0 = p_0 = q_0 \leftarrow 0_{H_2}$.

2: $r \leftarrow \|Ku_0 + v_0 - Y\|$ and $k \leftarrow 0$.

3: **while** $r > \tau$ **do**

4: $k \leftarrow k + 1$.

5: $v_k \leftarrow \tilde{v}$ where $\tilde{v} \in \mathcal{C}$ satisfies

$$\|\tilde{v} - (Y + \lambda p_{k-1} - Ku_{k-1})\|^2 \leq \|v - (Y + \lambda p_{k-1} - Ku_{k-1})\|^2$$

for all $v \in \mathcal{C}$.

6: $u_k \leftarrow \tilde{u}$ where $\tilde{u}$ satisfies

$$\begin{aligned} &\frac{1}{2}\|K\tilde{u} - (Y + \lambda p_{k-1} - v_k)\|^2 + \lambda J(\tilde{u}) + \frac{\lambda}{2\nu}\|\tilde{u} - (w_{k-1} + \nu q_{k-1})\|^2 \\ &\quad \leq \frac{1}{2}\|Ku - (Y + \lambda p_{k-1} - v_k)\|^2 + \lambda J(u) + \frac{\lambda}{2\nu}\|u - (w_{k-1} + \nu q_{k-1})\|^2 \end{aligned} \tag{3.29}$$

for all $u \in H_1$.

7: $w_k \leftarrow \tilde{w}$ where $\tilde{w} \geq 0$ satisfies

$$\|\tilde{w} - (u_k - \nu q_{k-1})\|^2 \leq \|w - (u_k - \nu q_{k-1})\|^2 \tag{3.30}$$

for all $w \in H_1$ with $w \geq 0$.

8: $p_k \leftarrow p_{k-1} - (Ku_k + v_k - Y)/\lambda$.

9: $q_k \leftarrow q_{k-1} - (u_k - w_k)/\nu$.

10: $r \leftarrow \max(\|Ku_k + v_k - Y\|, \|u_k - w_k\|, \|K(u_k - u_{k-1})\|)$.

11: **end while**

12: $u[\tau] \leftarrow u_k$ and $v[\tau] \leftarrow v_k$ and $w[\tau] \leftarrow w_k$ and $k[\tau] \leftarrow k$.

4 Applications and results

We now demonstrate the performance of SMR-estimators computed by the Augmented Lagrangian method of Chapter 3 by presenting numerical results. Throughout the chapter, we will focus on the discrete two-dimensional setting of Example 1.0.1. In order to indicate the versatility of the method, we will apply it to different operators, noise distributions and true objects in the underlying model (1.1). The presentation of the material is divided into two sections: denoising problems are treated in Section 4.1, deconvolution problems as an exemplary class of ill-posed inverse problems in Section 4.2.

4.1 Denoising

In this section, we will present SMR-estimators computed by our Augmented Lagrangian methodology for two-dimensional denoising problems. Throughout the section, we will hence assume that the data Y is given as

$$Y_{i,j} = u^*_{i,j} + \sigma\varepsilon_{i,j} \quad \text{where} \quad \varepsilon_{i,j} \overset{i.i.d.}{\sim} \mathcal{N}(0,1) \quad \text{for all} \quad i,j = 1,\dots,n. \tag{4.1}$$

First, we will present results for datasets which were simulated using synthetic test objects u^* in Subsection 4.1.1. One of the objects processed there will be used to illustrate how local adaptivity is established within our algorithm in Subsection 4.1.2. In order to assess the quality of the SMR-estimators, we will compare them to the results of *adaptive weights smoothing*, a state of the art method for denoising problems in Subsection 4.1.3. Finally, we will demonstrate in Subsection 4.1.4 how our algorithmic framework can also be applied to denoising of natural images by using transformed residuals as indicated in Section 3.3.

4.1.1 Synthetic test objects

Testing our algorithmic on synthetic test objects allows for a good evaluation of its performance as the underlying true object u^* is known and can be used as a reference for the result. In our experiments, we used four different objects. One of them is the "circles and bars" object which we presented along with its noisy counterpart in the introduction in Figure 1.1. The other three objects - which we will call "shapes", "squares" and "sticks" in the following - are given in Figure 4.1. There, we also show the corresponding noisy observations Y which we used in our experiments. Note that these objects exhibit different degrees of smoothness, varying locally and from scale to scale. They can hence be regarded

as well-suited to test the SMR-estimators for the desired properties which we formulated as our goals in the introduction.

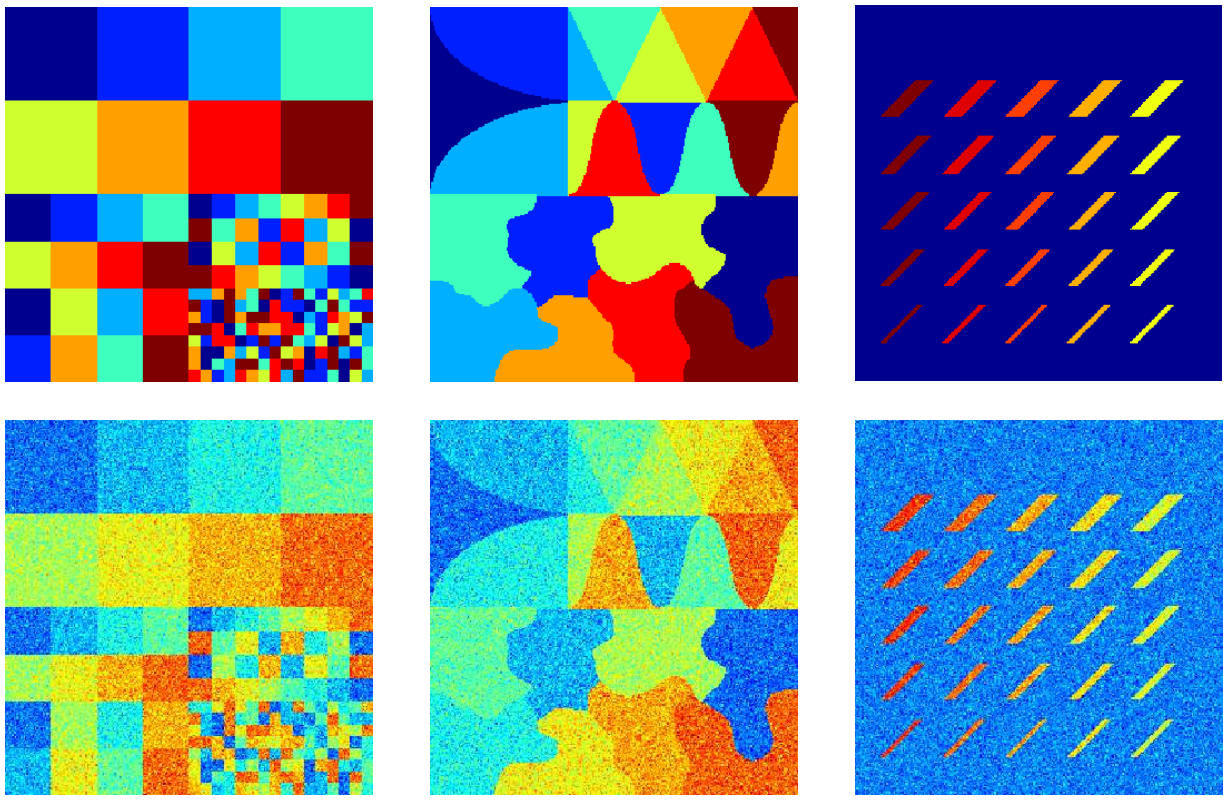

Figure 4.1: Synthetic test objects of size 256×256. From left to right: "squares", "shapes" and "sticks". Top: original object, scaled in $[0, 1]$. Bottom: perturbed observation Y as in (4.1) with $\sigma = 0.1$.

Our experiments were carried out for both the dyadic squares partitioning $\mathcal{P}_D$ (where the minimum scale was fixed to $s_{\min} = 4$) and the all squares partitioning $\mathcal{P}_A$ (taking into account scales from 1 through to 25). Furthermore, we used $J = \mathrm{TV}$ for all results given in this subsection. The corresponding results of the Augmented Lagrangian method are depicted in Figures 4.2 and 4.3. Visual inspection of the results reveals that both partitionings deliver good results and especially exhibit the desired locally adaptive behaviour. Note that small features like the nine dots in the bottom-right of "circles and bars" are preserved, while larger areas like the two big circles in the top-left are well smoothed at the same time. For the "sticks" object, the decreasing intensities from left to right are well reconstructed and all edges are particularly sharp, independent of the size of the feature.

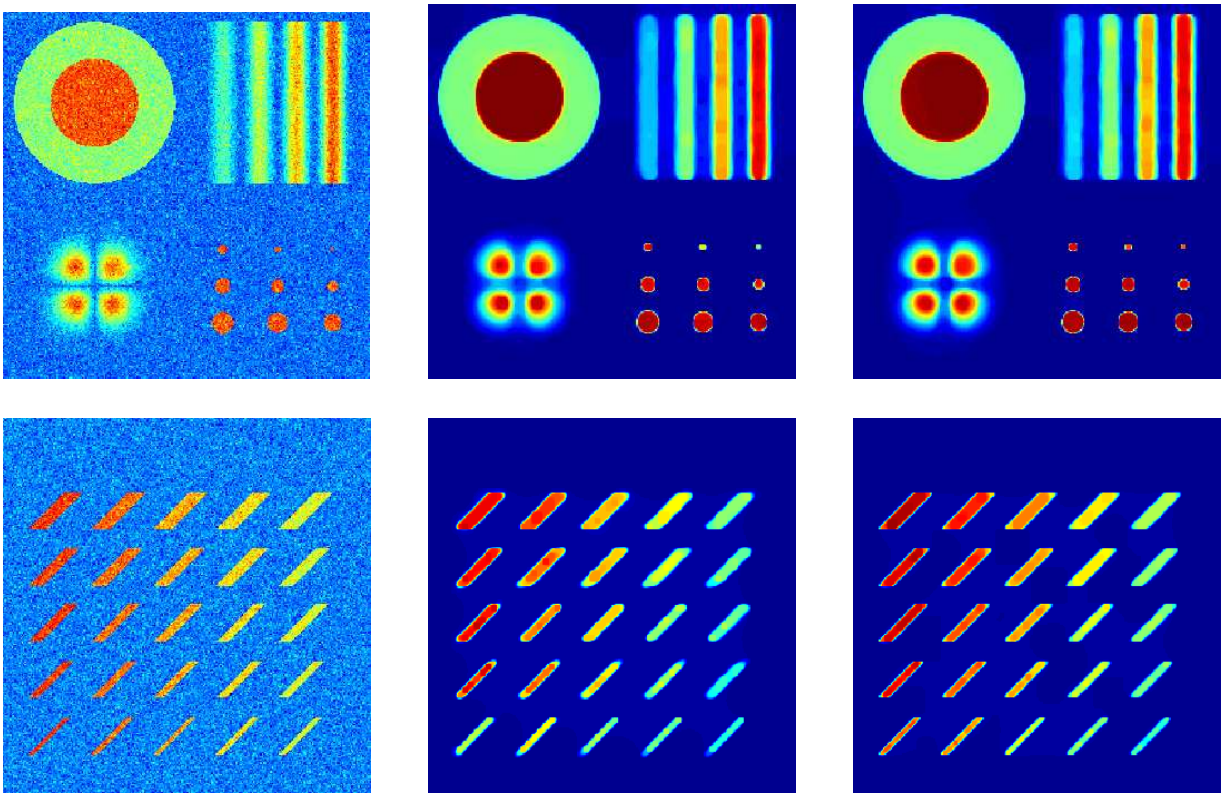

Figure 4.2: Results of Augmented Lagrangian method. Left: noisy data $Y = u + \sigma\varepsilon$, $\sigma = 0.1$. Middle: result using dyadic squares partitioning. Right: result using all squares partitioning.

The results also show that by means of an all squares partitioning, more details can be resolved than by a dyadic squares partitioning which can especially be observed in the reconstructions of the "circles and bars" object. This results from the fact that $\#\mathcal{P}_A \gg \#\mathcal{P}_D$ and the SMR-estimator hence locally adapts on more regions if the former is used. On the other hand, the higher complexity of $\mathcal{P}_A$ also has a drawback: using $\mathcal{P}_A$ instead of $\mathcal{P}_D$ drastically increases the runtime of the Augmented Lagrangian method. In fact, the results shown in Figures 4.2 and 4.3 were computed about ten times faster on average for $\mathcal{P}_D$ than for $\mathcal{P}_A$. This is an immediate consequence of the larger number of sets D_m that enter Dykstra's Algorithm 3. According to the formulae derived at the end of Section 3.2, the number M of those sets is given by 8 for a dyadic squares partitioning in contrast to $5,525$ for an all squares partitioning. In summary, we have to decide between reconstruction quality and runtime according to the practical application at hand and the related goal of

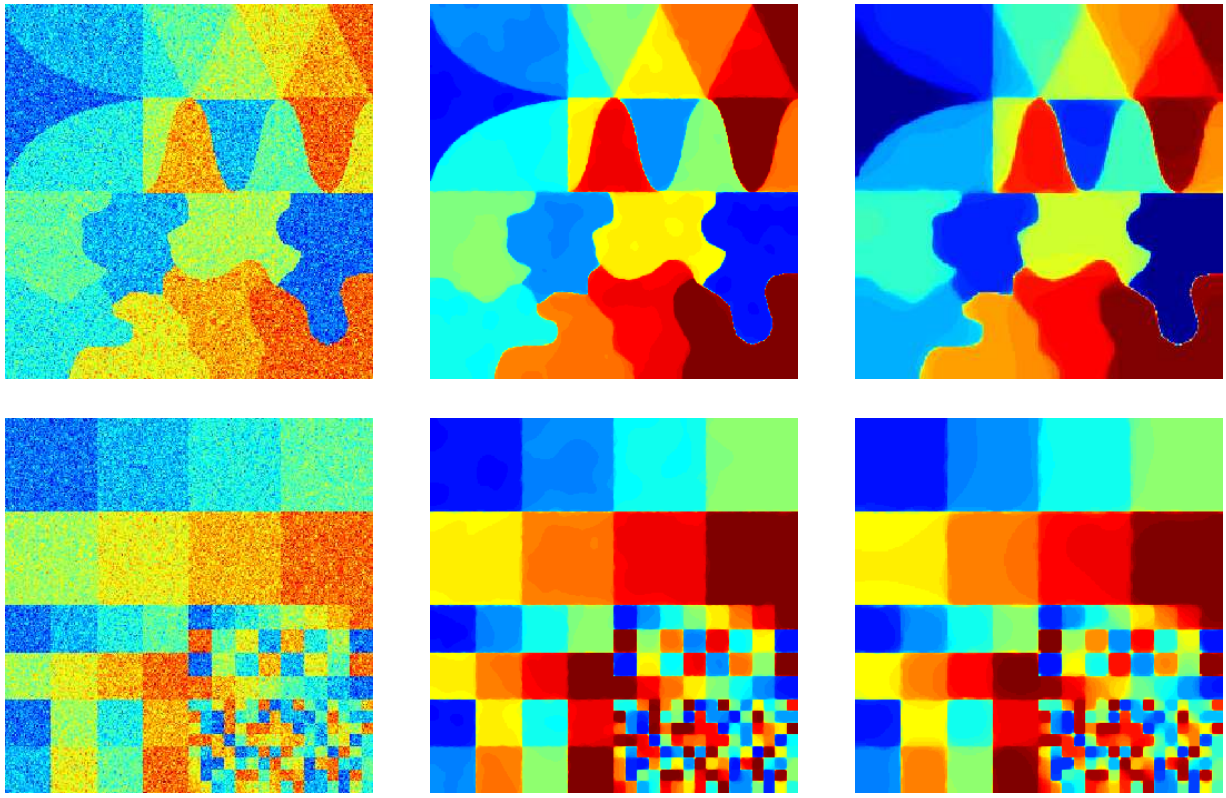

Figure 4.3: Results of Augmented Lagrangian method (contd.)

the reconstruction when choosing one of these partitionings.

In Figure 4.4, we compare the SMR-estimators computed by the Augmented Lagrangian method for "circles and bars" and "shapes" to those computed by the automatic local parameter adjustment presented in [64], a method which we briefly described at the end of Section 3.1. As a dyadic squares partitioning was used in [64], we draw the comparison to our method for both the dyadic and the all squares partitioning. Clearly, the Augmented Lagrangian method presented in this thesis outperforms the local parameter adjustment in these examples already if a dyadic squares partitioning is used. While keeping edges sharp and reconstructing intensities well, the results of the Augmented Lagrangian method also preserve smoothness in the background and on top of flat features. The results of the local parameter adjustment look quite undersmoothed in contrast. Computing SMR-estimators rigorously by solving the constrained optimization problem (2.6) instead of circumventing the problem and reducing the regularization parameter locally is hence worthwhile as it seems to lead to additional precision of the results.

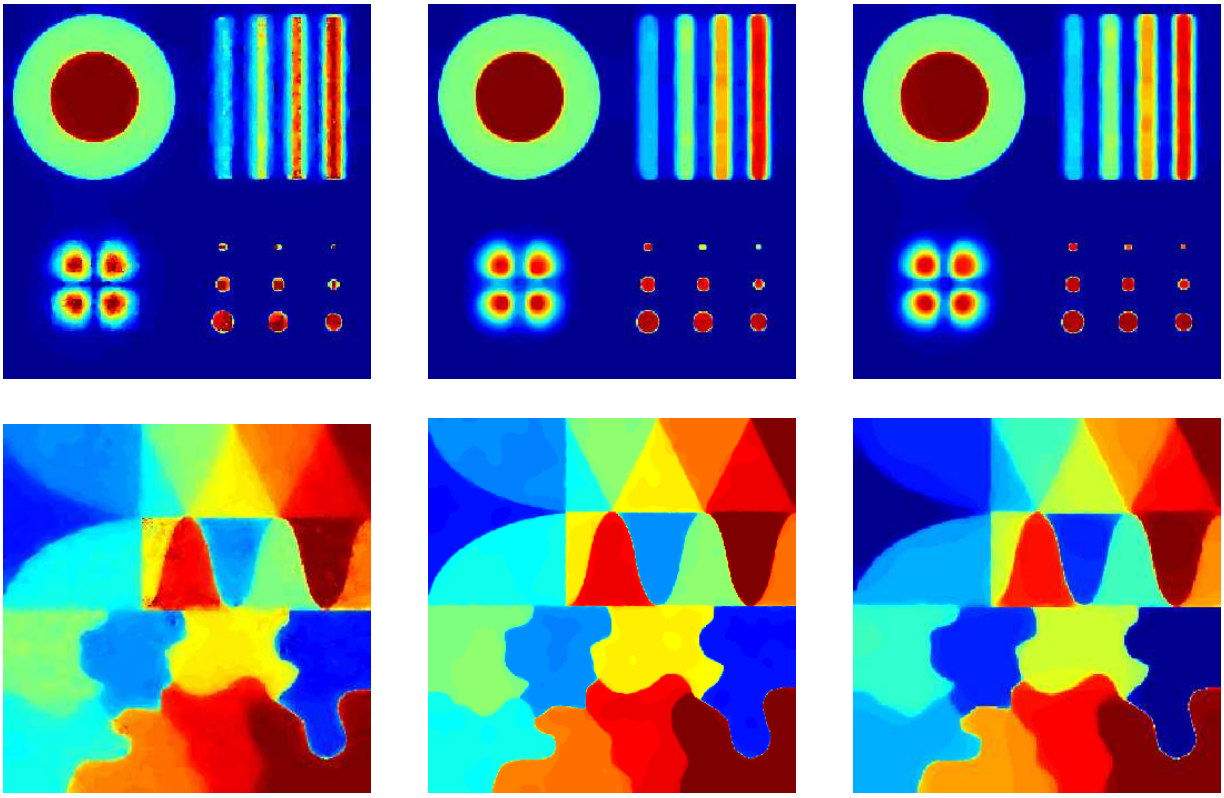

Figure 4.4: Comparison of results. Left: local parameter adjustment (cf. [64]). Middle: Augmented Lagrangian method using dyadic squares partitioning. Right: Augmented Lagrangian method using all squares partitioning.

According to the results shown here, we have found our method of choice for the computation of SMR-estimators in denoising problems. The results of the Augmented Lagrangian method exhibit an appealing locally and multiscale adaptive nature and can still be computed within reasonable time. Moreover, the theoretical background provided in Sections 2.1 and 2.2 makes the estimator statistically sound. The goals formulated in the introduction have hence been reached for denoising problems.

4.1.2 Illustration of local adaptivity

We will now illustrate how local adaptivity is established within the Augmented Lagrangian method. As we already mentioned in Section 3.1, this is done by locally modifying the input

$$Z_k := Y + \lambda p_{k-1} - v_k$$

to the J-penalized least-squares problem (3.8) which is solved in iteration step k of Algorithm 2 rather than by locally modifying the regularization parameter as in the methodology presented in [64]. To illustrate this, we processed the noisy version of "circles and bars" as displayed in Figure 1.1 with different choices of the step length λ. In order to clarify the effects we would like to demonstrate, we restrict our illustrations to column 230 of the object which runs through the last column of the nine little dots and the highest "bar". Choosing this one-dimensional slice of the dataset especially avoids confusing rescaling effects.

In Figure 4.5, we illustrate intermediate results of the Augmented Lagrangian method presented in Algorithm 2. In the top row, the step length was chosen as $\lambda = 1$. The left column shows the data Y (red) and the input Z_k to the TV-penalized functional (blue) in the last iteration step of the Augmented Lagrangian method before the stopping condition was fulfilled. The right column displays the true object (green), the SMR-estimator $\hat{u}_N(0.9)$ as computed by the Augmented Lagrangian method (blue) and the global estimator $\hat{u}_\lambda$ (red) computed via (1.6) with regularization parameter λ. In the bottom row, the same quantities are displayed, but this time for $\lambda = 0.01$.

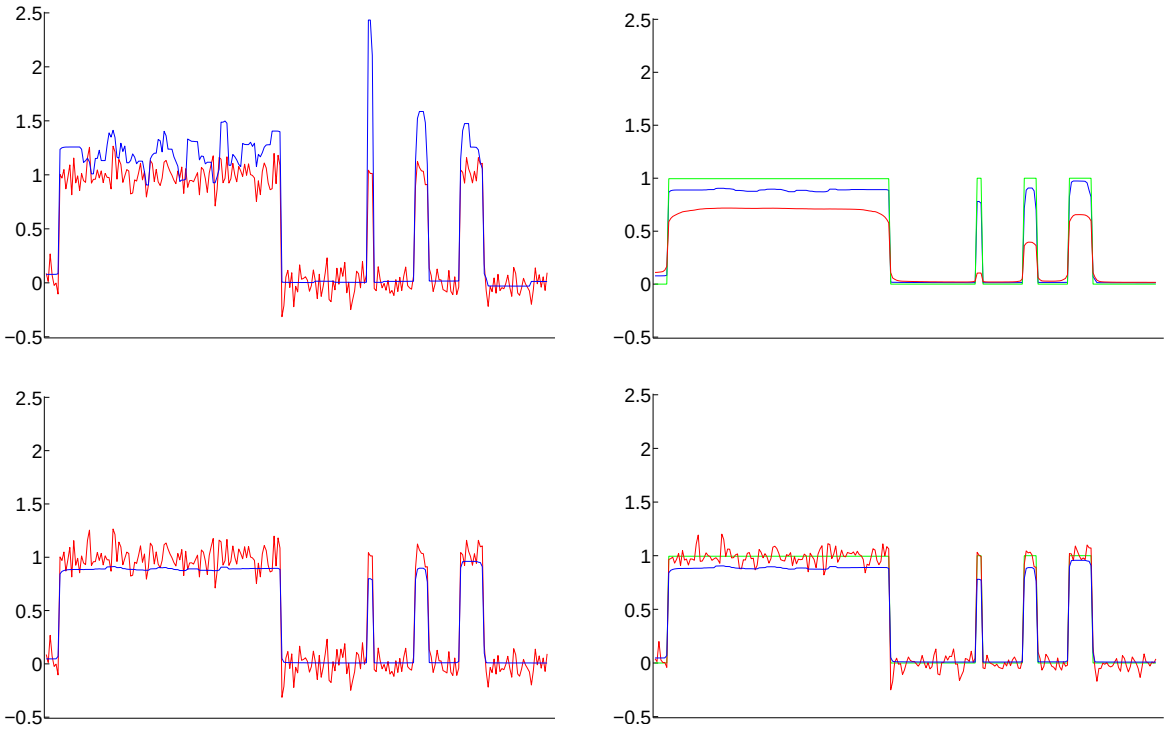

Figure 4.5: Illustration of local adaptivity; one-dimensional cut through "circles and bars" in column 230. Top: $\lambda = 1$. Bottom: $\lambda = 0.01$.

As $\lambda = 1$ is a rather large choice of a step length for the dataset Y at hand, we see that the global estimator in the top right is clearly oversmoothed, especially for the leftmost dot. In the Augmented Lagrangian method, this gets corrected locally by drastically increasing intensities in the corresponding regions of Z_k while the impact on better reconstructed regions is smaller, see top left. Although this modified input to (3.8) in the last iteration step still gets smoothed heavily as the parameter λ remains unchanged, the increased intensities lead to a much better approximation of the final result to the true object.

For the small step length $\lambda = 0.01$, the global estimator almost matches the observation as the data-fit is strongly emphasized. The smoothing of the algorithm's outcome hence gets done to the Z_k's already, yielding an input to (3.8) in the last iteration which almost coincides with the final SMR-estimator. During the first iteration steps of the Augmented Lagrangian method, the image of the solution u_k of (3.8) under K is close to the data due to the small regularization parameter λ used there. As a consequence, the residuals of u_k are small in absolute value and little to no projections at all are performed by Dykstra's algorithm during the quadratic program step (3.7). It is therefore rather the update of the dual variable p_k in each step that leads to the modification of the input Z_k.

As expected, the final outcome of the algorithm is identical within a small tolerance for both choices of the step length. Independently of this parameter, the algorithm converges to the same result asymptotically according to Theorem 3.1.3.

4.1.3 Comparison to AWS

In order to evaluate the quality of the SMR-estimators shown in Subsection 4.1.1, we will now draw a comparison to a state of the art method in the field of denoising. *Adaptive weights smoothing* (AWS) as introduced in [90] is a natural choice of such a method. The estimators computed by AWS exhibit a locally adaptive nature, just as SMR-estimators, and are hence particularly well-suited for such a comparison. To process our datasets, we used the R-package "aws" [89].

We performed a simulation study in which we compared SMR-estimators for $J = \mathrm{TV}$ to estimators computed via AWS for three different noise levels σ. We simulated 100 observations according to the data model (4.1) and processed them with both methods. As the underlying true object u^*, we used "circles and bars" of size 256×256 as shown in Figure 1.1. For both reconstruction methods, we computed the average of two different distance measures between the respective estimator and the true object. To be exact, we

used the *mean squared error* (MSE)

$$\mathrm{MSE}(u, v) := \frac{1}{\#X} \sum_{(i,j)\in X} (u_{i,j} - v_{i,j})^2$$

and the *mean symmetric Bregman divergence* (MSB) with respect to $J = \mathrm{TV}$ where the symmetric Bregman divergence is defined for general functions J as

$$D_J^{\mathrm{sym}}(u, v) := \frac{1}{2} \langle J'(u) - J'(v), u - v \rangle . \tag{4.2}$$

As such scalar values are often insufficient to reliably classify the similarity between two images due to their complexity, we also show exemplary plots of the estimators resulting from the datasets used in Subsection 4.1.1. By combining the interpretation of the distance measures received from the simulations with simple visual inspection of the plots, a good assessment of the overall quality of the estimators is guaranteed.

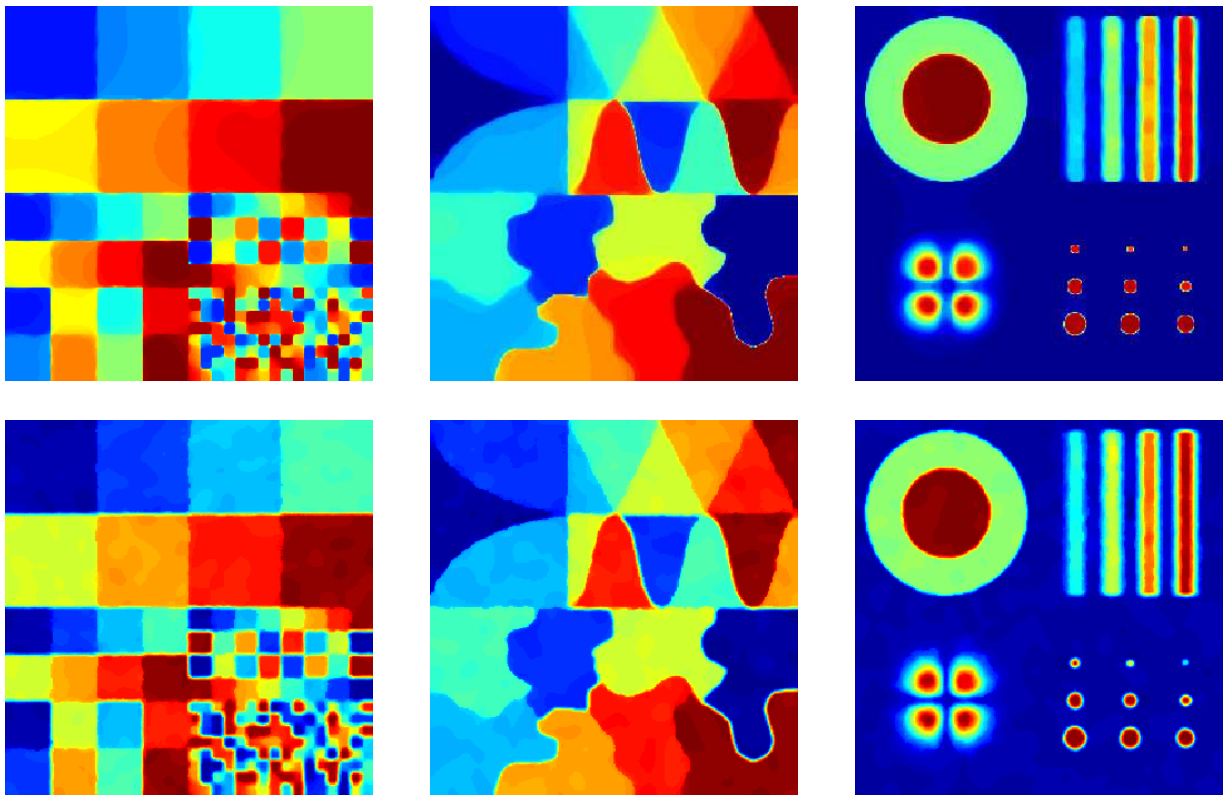

Figure 4.6: Comparison between SMR-estimation and AWS for $\sigma = 0.1$, observations Y are as in Figures 1.1 and 4.1. Top: SMR-estimators. Bottom: AWS.

	$\sigma = 0.1$		$\sigma = 0.25$		$\sigma = 0.5$	
	MSE	MSB	MSE	MSB	MSE	MSB
$\hat{u}_N(0.9)$	0.001	0.003	0.004	0.004	0.009	0.005
$\hat{u}_{\text{AWS}}$	0.002	0.009	0.003	0.011	0.005	0.016

Table 4.1: Comparison between SMR-estimators and AWS for different noise levels in the model (4.1), simulation study. True object was "circles and bars" of size 256×256. Numbers are averaged over 100 simulations.

We provide the estimators resulting from the datasets shown in Figures 1.1 and 4.1 (at a noise level of $\sigma = 0.1$) in Figure 4.6. Visual inspection reveals that the SMR-estimator as computed by our methodology is superior to the result of AWS in most regions. Observe that sharp edges as in "squares", "shapes" and the bottom right of "circles and bars" are clearly better reconstructed by the SMR-estimator. AWS tends to oversmooth such edges as can e.g. be seen for the smaller "squares" in the bottom right. On the other hand, smooth transitions like those in the bottom left of "circles and bars" are particularly well reconstructed by AWS. The SMR-estimator in this region looks slightly oversmoothed in contrast. The visual impression that SMR-estimation yields better reconstructions than AWS for this noise level coincides with the results of our simulation study given in Table 4.1. Both quantities evaluated there are smaller for SMR-estimators. All in all, SMR-estimation outperforms AWS for this noise level.

Observe that the ratio of the distance measures in Table 4.1 changes with increasing noise level σ. For the maximum value of $\sigma = 0.5$, the MSB is still distinctly lower for SMR-estimators, but the MSE is now smaller for estimators computed via AWS. The corresponding exemplary plots provided in Figure 4.7 yet reveal that SMR-estimators yield a reconstruction quality that is competitive with AWS in this rather extreme setting, too. Especially the smoothing of flat features in "shapes" and "squares" is convincing, while on the other hand the intensities of the little dots in "circles and bars" are much better reconstructed by the estimators computed via AWS.

In summary, we find that for the smaller noise level of $\sigma = 0.1$, SMR-estimators exhibit a higher reconstruction quality than estimators computed via AWS, especially for blocky features. This is remarkable as AWS is usually considered an excellent state of the art technique for denoising applications. According to our simulation study, the difference in the MSE between the two estimation schemes compared here gets smaller with increasing

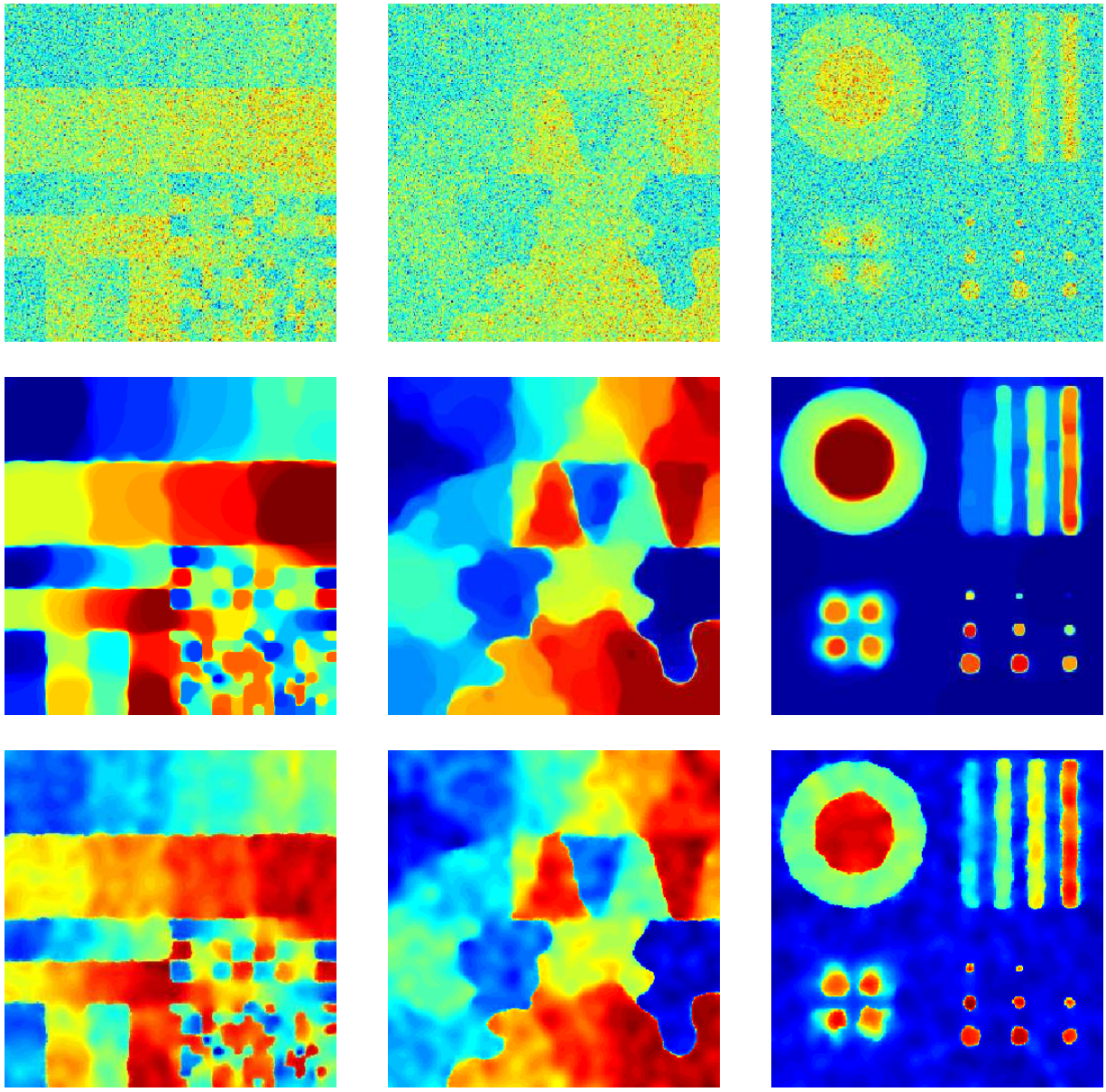

Figure 4.7: Comparison between SMR-estimation and AWS for $\sigma = 0.5$. Top: observations Y. Middle: SMR-estimators. Bottom: AWS.

noise level σ, with AWS giving better results from a certain level on. Nonetheless, the reconstructions for a high noise level of $\sigma = 0.5$ as shown in Figure 4.7 suggest that SMR-estimators are competitive in this setting, too. A substantial drawback of our method in comparison to AWS, however, is its computation time which is disproportionately longer.

4.1.4 Natural images

Concluding our presentation of results for denoising problems, we will now give an example on how the transformed residuals introduced in Section 3.3 might be used in practice. As described in [99] (see also Section 2.1), the multiresolution statistic T_N on H_2 in combination with a dictionary Φ consisting of systems of subsets of X (such as $\mathcal{P}_D$ or $\mathcal{P}_A$) can be considered as a likelihood-ratio statistic. By means of the MR-statistic, the null hypothesis that a given signal is a realization of a white noise process is tested against the alternative that the underlying Gaussian process has non-zero *but constant* mean on some $S \in \mathcal{P}$. Clearly, the power of this test increases with the size of S in the alternative. In the context of image denoising, this means that it is particularly powerful for images which contain large areas with constant gray values.

Figure 4.8: Standard test images. Left: "cameraman". Middle: "lena". Right: "roof".

Natural images such as photographs, however, are seldom composed of such areas. Instead, a substantial part of these images consists of oscillating patterns as these often occur in textures as e.g. fabric, wood, hair or grass. This becomes obvious in the standard test images depicted in Figure 4.8 (at a resolution of 256×256 and scaled in $[0, 1]$). Here, as expected, the statistical test as formulated so far performs rather poorly.

In order to illustrate this, we simulate noisy observations Y of the test images in Figure 4.8 according to (4.1) with $\sigma = 0.1$ (cf. left column of Figure 4.10) and compute a TV-penalized least-squares estimator $\hat{u}_a$ via (1.6) with $a = 0.1$ (cf. middle column of Figure 4.10). We intend to examine how well oversmoothed regions in $\hat{u}_a$ are detected by the statistic $T_N(Y - \hat{u}_a)$: the left picture in Figure 4.9 depicts the local averages of the residuals $r = Y - \hat{u}_a$ for the "roof"-image, that is

$$\mu_S(r) = \frac{\left|\sum_{(i,j)\in S} r_{i,j}\right|}{\sqrt{\#S}}$$

for all 5×5-squares $S \subset X$. Large values indicate locations where, according to the MR-statistic $T_N(r)$, the residual fails to resemble white noise (i.e. where the estimator $\hat{u}$ is considered oversmoothed). Although some relevant parts are detected (e.g. parts of the roof), it becomes visually clear that the localization is rather poor.

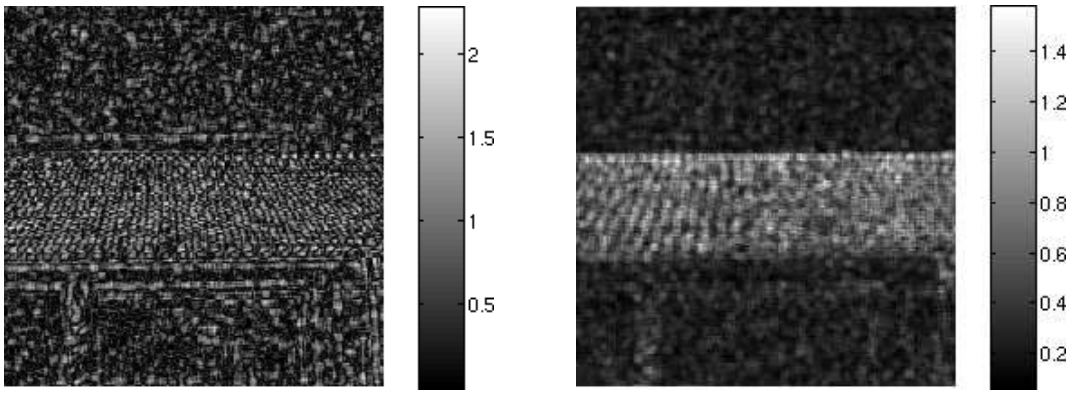

Figure 4.9: Local averages of the residuals for "roof" image on 5×5-squares. Left: ordinary $\mu_S(r)$. Right: using squared residuals $\mu_s(r^2)$.

The performance can be improved significantly by applying the multiresolution statistic to the pointwise squared residuals as described in Example 3.3.1. The right image in Figure 4.9 depicts the corresponding local averages $\mu_S(r^2)$ and indicates that the localization of oversmoothed regions in $\hat{u}$ at the scale 5×5 is substantially improved. This is a good motivation for incorporating the local averages of the squared residuals in the SMR-estimator model (2.6). The resulting estimation procedure constitutes a multiscale generalization of the model suggested in [34].

Since $\varepsilon_{i,j} \overset{i.i.d.}{\sim} \mathcal{N}(0,1)$ for all $(i,j) \in X$, the statistic

$$\sqrt{\#S}\mu_S(\varepsilon^2) = \sum_{(i,j)\in S} \varepsilon_{i,j}^2$$

is χ^2-distributed with $\#S$ degrees of freedom. Therefore, the statistics

$$\frac{1}{\sqrt{2}}\left(\mu_S(\varepsilon^2) - \sqrt{\#S}\right)$$

have mean zero and variance one, though are not identically distributed for different scales $\#S$. As a consequence, the statistic $T_N(\varepsilon)$ in (2.3) (in the present situation with $f(s) = s$ and rescaled by $1/\sqrt{2}$) is not necessarily a good choice since it corresponds to the extreme-value statistic of *non-identically distributed* random variables. This constitutes a substantial drawback of this modified version.

Figure 4.10: Global reconstructions and SMR-estimators. Left: noisy data $Y = u + \sigma\varepsilon$ with $\sigma = 0.1$. Middle: global TV-penalized least-squares reconstruction $\hat{u}$. Right: SMR-estimators $\hat{u}_N(0.9)$.

A possible modification of the SMR-estimator approach consists in considering the individual scales separately. To be precise, we set for all $\phi_i = \chi_{S_i}$ with $\#S_i = s^2$ the threshold value c_i in (3.17) to $c_i = c_s := \sigma^2(\sqrt{2}q_s(\alpha) + s)$, where $q_s(\alpha)$ is the $(1-\alpha)$-quantile of

$$t_s(\varepsilon) = \sup_{\substack{S \in \mathcal{P} \\ \#S = s^2}} \left(\frac{1}{\sqrt{2}} \left(\mu_S(\varepsilon^2) - s \right) \right).$$

In other words, we perform a scale-wise test instead of handling all scales involved at once. By modifying the SMR-estimator paradigm in this way, a different statistical interpretation arises: for the true image u^* the residual $r = \sigma^{-1}(Y - u^*)$ satisfies $t_s(r) \leq q_s(\alpha)$ *for a fixed scale* $s = \#S$ with probability of at least $1 - \alpha$. However, this does not imply that this holds *for all scales simultaneously*. Put differently, the probability that the true solution lies in the admissible domain of the convex problem (3.17) is in general significantly smaller than $1 - \alpha$ due to the multiple tests being performed (one for each scale involved). This means that the feasible region of (3.17) does not constitute a $(1-\alpha)$-confidence region for the SMR-estimator $\hat{u}_N(\alpha)$ as it was the case for the original approach (2.3).

Despite these problems, we illustrate the applicability of this modified approach by studying some examples. Figure 4.10 shows the noisy counterparts Y of the test images in Figure 4.8 simulated according to the model (4.1) with $\sigma = 0.1$ (left column), global estimators $\hat{u}_a$ as in (2.14) where $J = \mathrm{TV}$ and $a = 0.1$ (middle column) and the SMR-estimator $\hat{u}_N(0.9)$. We note that this specific choice of a is rather arbitrary but already shows the benefit of our method. While smooth regions like the sky in "cameraman" are still undersmoothed by the global estimators, they also exhibit a significant oversmoothing in textured regions. Both of these disadvantages could not be removed at the same time when increasing and decreasing the global parameter a, respectively.

The SMR-estimators as depicted in Figure 4.10, however, exhibit good reconstructions in all regions, independent of the local smoothness of the true object. Textured regions like the feather in "lena" or the shingles in "roof" are well reconstructed and so are smoother regions like sky and background. The SMR-estimator hence shows the desired locally adaptive nature which we formulated as our goal and clearly outperforms the global estimator. Moreover, it is still statistically sound, although we changed our original paradigm as described above. In summary, the methodology presented in this subsection is well-suited for denoising of natural images. As its theoretical background deflects from the one presented in the rest of the thesis, we abstain from studying this approach any further.

4.2 Deconvolution

In contrast to the algorithms presented in [64] and [100], the Augmented Lagrangian method of Chapter 3 also allows for computation of SMR-estimators if the operator K in the underlying model (1.1) is non-trivial or even ill-posed. We will now therefore turn our attention to such inverse problems, focusing on the class of *convolution operators*. In the discrete two-dimensional setting of Example 1.0.1, these operators take the form

$$(Ku)_{i,j} = (k * u)_{i,j} := \sum_{(k,l)\in\mathbb{Z}^2} k_{i-k,j-l} u_{k,l} \tag{4.3}$$

where k is a square-summable *kernel* on the lattice $\mathbb{Z}^2$ and $u \in H_1$ is extended by zero-padding. A kernel which is of special interest is the *circular Gaussian kernel*

$$k_{i,j} = \frac{1}{2\pi\sigma_K^2} e^{-\frac{i^2+j^2}{2\sigma_K^2}} \tag{4.4}$$

with standard deviation σ_K. Applying a convolution operator to an object $u \in \mathbb{R}^{n\times n}$ has a certain blurring effect on the object (as can be seen in the first line of Figures 4.11 and 4.12). The corresponding class of inverse problems is therefore frequently referred to as *deblurring problems* in image processing.

As we did in our presentation of denoising results, we start out by showing SMR-estimators computed by our methodology for synthetic test objects. Afterwards, we compare our results to those of other data-driven estimation schemes for inverse problems and to so-called oracles in Subsections 4.2.2 and 4.2.3. An application of our methodology in the field of fluorescence microscopy in 4.2.4 concludes the section.

4.2.1 Synthetic test objects

In order to test our methodology on deconvolution problems in which the operator is defined via a circular Gaussian kernel, we created synthetic test data using the objects "circles and bars", "shapes" and "squares" at a resolution of 256×256 as shown in Figure 4.1. The standard deviation of the kernel of the convolution operator was set to $\sigma_K = 4$ and the noise level to $\sigma = 0.1$.

We computed SMR-estimators using our Augmented Lagrangian methodology and chose J as total variation (cf. Section 2.4) and the L^2-norm, respectively. The latter choice of J

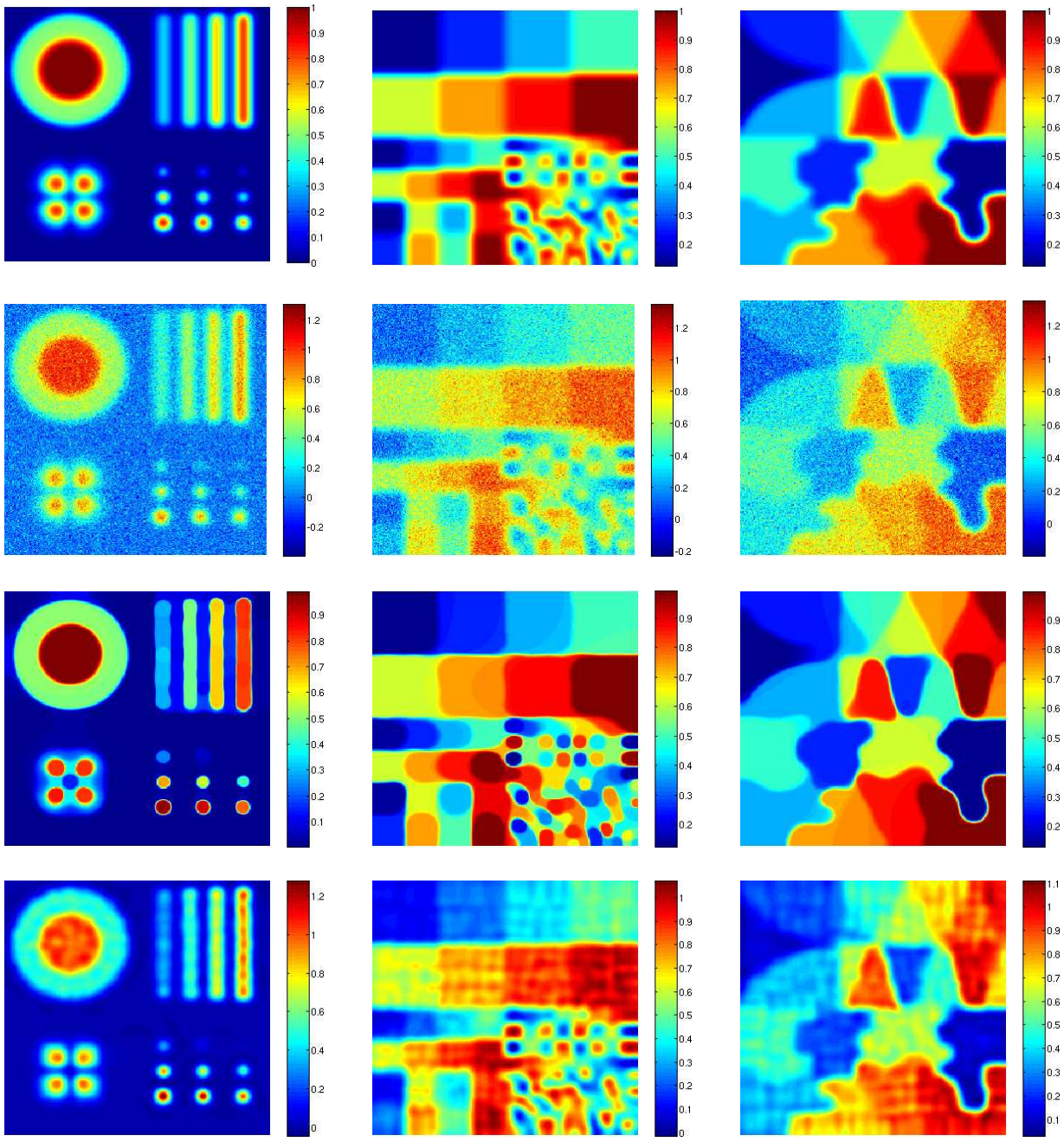

Figure 4.11: Results of deconvolution. Top: convolved objects Ku^* where $\sigma_K = 4$. Second: observations Y, noise level $\sigma = 0.1$. Third: SMR-estimator $u_N(0.9)$ where $J = \mathrm{TV}$. Bottom: SMR-estimator $u_N(0.9)$ where $J = \mathrm{L}^2$.

leads to the well-known *Tikhonov-Philips-regularization* introduced in [101]:

$$\hat{u}_a = \operatorname*{argmin}_{u \in H_1} \frac{1}{2} \|Y - Ku\|^2 + a \|u\|^2 .$$

It is a standard fact in inverse problems that a solution to this scheme is given by

$$\hat{u}_a = (K^*K + aI)^{-1} K^*Y.$$

This alternative choice of J is meant to illustrate the versatility of our approach with respect to the option of using different penalty functions. As Tikhonov-Philips-regularization results in simple rescaling when used for denoising problems, we did not employ it in Section 4.1 already.

The convolved objects, datasets and results are given in Figure 4.11. Visual inspection of these reveals that - just as in the case of denoising - SMR-estimators exhibit the desired locally adaptive nature for deconvolution problems, too. Moreover, the different choices of J lead to good reconstructions in regions which match the underlying smoothness assumption corresponding to the penalty function being used (blocky structures for TV, low intensities for L^2), as expected. The deconvolution effect itself is satisfactory, but some smaller features of the original objects could not be reconstructed by the SMR-estimators (like e.g. the little dots in "circles and bars" and some of the "squares"). In view of the datasets in the second row of the figure, this is not surprising though. These features are literally lost in the observations.

To test our methodology in even more involved situations, we repeated our experiments for a convolution operator with a kernel of bigger variance. To be exact, we set $\sigma_K = 10$ and reduced the noise level to $\sigma = 0.05$. The datasets and results for this setting are given in Figure 4.12. In this rather extreme situation, it would be unrealistic to expect an estimator that is as close to the true object as e.g. those shown in the denoising examples of Section 4.1. Nonetheless, the SMR-estimators are still capable of reconstructing at least some of the objects' features which are hard to detect visually in the datasets, especially the triangles in "shapes" and some of the "squares". When comparing these results to those of other automatic estimation schemes in the next subsection, we will see that these results can be regarded as satisfactory, too.

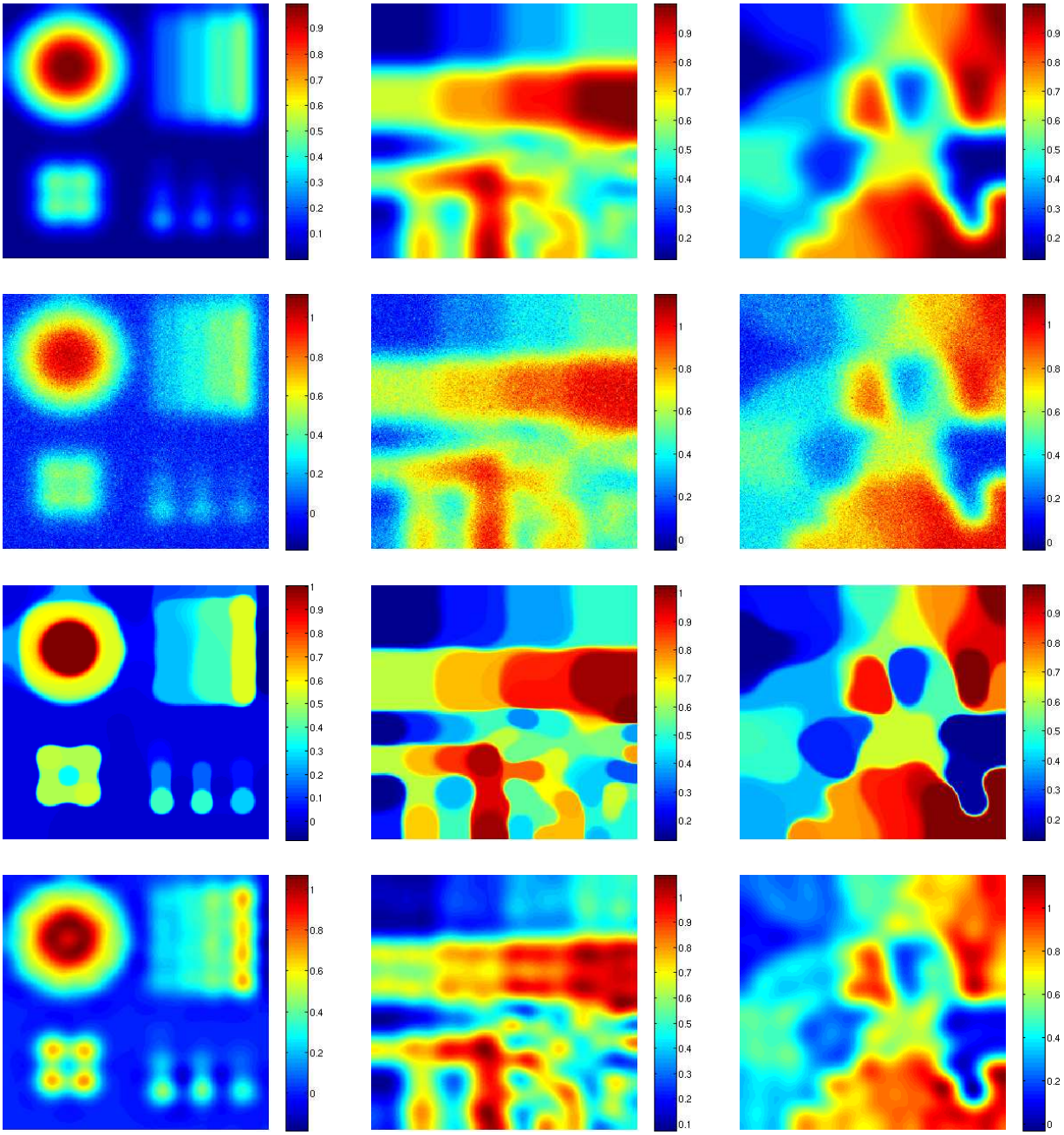

Figure 4.12: Results of deconvolution. Top: convolved objects Ku^* where $\sigma_K = 10$. Second row: observations Y, noise level $\sigma = 0.05$. Third row: SMR-estimator $u_N(0.9)$ where $J = \mathrm{TV}$. Bottom: SMR-estimator $u_N(0.9)$ where $J = \mathrm{L}^2$.

4.2.2 Comparison to other methods

As a reference for the SMR-estimators for deconvolution problems shown in the previous subsection, we now present the results of two data-driven parameter selection schemes for the regularization parameter, namely the *L-curve method* and the *Lepskij principle*.

Both of these approaches are based on the choice of a global regularization parameter in the reconstruction scheme (1.6) and are therefore not locally adaptive at all (and so are the oracles which we will present in Subsection 4.2.3). As local adaptivity is one of the central advantages of SMR-estimators, a fair comparison should be drawn to other locally adaptive estimation techniques. To the author's knowledge, the only works that deal with such techniques in an inverse problems setting are [28], [73] and [74]. The framework treated therein is restricted to linear first-kind integral equations though and can therefore not be used for a comparison to our general linear inverse problem setting. For this reason, we stick to the methods of comparison named above.

We start out with a brief derivation of the L-curve method. When plotting the logarithm of the squared norm of the residuals $Y - K\hat{u}_a$ against the logarithm of the squared norm of the estimator $\hat{u}_a$ for different values of a, the resulting graph is usually L-shaped (see e.g. [104, Chapter 7]). The *L-curve method* aims at choosing the regularization parameter a_{LC} that corresponds to the "corner" of this curve. In [58], it was proposed to use the point of maximum curvature. For an analysis and further details of the L-curve method, we also refer to [54; 55; 57]. Following the lines of [104, Chapter 7], we set $R(a) := \|Y - K\hat{u}_a\|^2$ and $S(a) := \|\hat{u}_a\|^2$. By simple calculus we obtain that the curvature of the graph of $(\log R(a), \log S(a))$ is then given by

$$\kappa(a) = -\frac{R(a)S(a)(aR(a) + a^2S(a)) + (R(a)S(a))/S'(a)}{(R^2(a) + a^2S^2(a))^{3/2}}.$$

We fix a set of candidate parameters $\{a_1, \dots, a_M\}$ and choose the L-curve method parameter $a_{\mathrm{LC}} := a_{\tilde{j}}$ according to

$$\tilde{j} := \max_{j=1,\dots,M} \kappa(a_j).$$

This parameter is then used to compute a solution $\hat{u}_{\mathrm{LC}} = \hat{u}_{a_{\mathrm{LC}}}$ of (1.6).

In our simulations, we applied the L-curve method to TV-penalized least-squares estimation and compared the results to SMR-estimators computed via our Augmented Lagrangian methodology. As we did in Subsection 4.1.3, we combine visual inspection and averaged distance measures from a simulation study to compare the quality of the reconstructions.

Figures 4.13 and 4.14 show the SMR-estimators and the results of the L-curve method for the datasets processed in Subsection 4.2.1 for the objects "circles and bars" and "squares" of size 256×256 and the different convolution kernels and noise levels used there (i.e. $\sigma_K = 4$ combined with $\sigma = 0.1$ and $\sigma_K = 10$ combined with $\sigma = 0.05$). Table 4.2 shows the results of our simulation study in which we used the "circles and bars" object resized to 128×128. Consequently, we also halved the kernels' standard deviations to $\sigma_K = 2$ and $\sigma_K = 5$, respectively, in this study. We combined both kernels with two different noise levels each ($\sigma = 0.1$ and $\sigma = 0.05$), performed 100 simulations and evaluated the same quantities as in 4.1.3, namely MSE and MSB.

Interpreting the results in Figure 4.13, we find that the SMR-estimators shown there are clearly superior to the corresponding results of the L-curve method. In particular, the desired local-adaptivity of the SMR-estimators we already mentioned in our interpretation in Subsection 4.2.1 is convincing, while the results of the L-curve method look severely undersmoothed in comparison. This leads to the MSB of the latter being much larger than the one of the SMR-estimator, see Table 4.2. Note, however, that the MSE's shown there are distinctly larger for the SMR-estimator. Nonetheless, visual inspection of Figure 4.13 undoubtedly shows that the SMR-estimator outperforms the L-curve method in this setting. This indicates that the MSE should not always be regarded as a particularly trustable measure for the similarity of images. When applied to the datasets for the kernel with $\sigma_K = 10$, the L-curve method performs almost as good (or even slightly better) than SMR-estimators, see Figure 4.14. However, the visual difference between the corresponding estimators is not as big as the numbers in Table 4.2 suggest.

		$\sigma = 0.05$		$\sigma = 0.1$	
		MSE	MSB	MSE	MSB
$\sigma_K = 2$	$\hat{u}_N(0.9)$	0.006	0.008	0.008	0.009
	$\hat{u}_{\mathrm{LC}}$	0.004	0.011	0.006	0.013
$\sigma_K = 5$	$\hat{u}_N(0.9)$	0.028	0.019	0.031	0.020
	$\hat{u}_{\mathrm{LC}}$	0.020	0.014	0.021	0.019

Table 4.2: Comparison between SMR-estimation and L-curve method, simulation study. True object was "circles and bars" resized to 128×128. Numbers are averaged over 100 simulations.

Our second method of comparison, the *Lepskij principle*, was first introduced in [75] and is based on the idea to choose the estimator which balances bias and variance in its

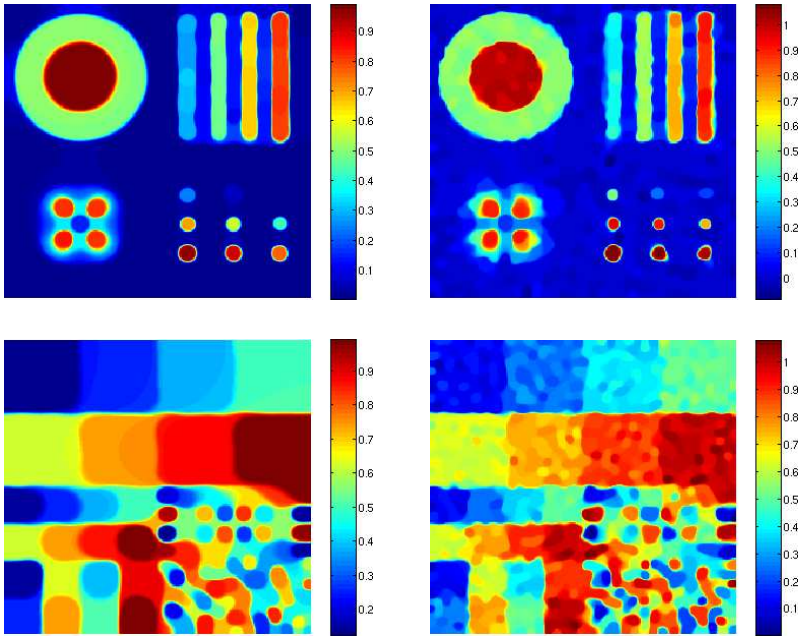

Figure 4.13: Comparison between SMR-estimation (left) and L-curve method (right) for datasets of Figure 4.11, i.e. $\sigma_K = 4$ and $\sigma = 0.1$. Top: "circles and bars". Bottom: "squares".

mean integrated square error (MISE). For this reason, it is also referred to as the *balancing principle*. We provide an outline of the principle and present a formulation of it in the stochastic noise setting at hand. For further details of the method and an analysis of it in different settings, we refer to [4; 8; 59; 78; 80].

For fixed penalty function J and data Y, let $u^\dagger$ denote the J-minimizing solution of (1.1) which we introduced in Theorem 2.2.5. Furthermore, let $R_a : H_2 \to H_1$ denote the operator that maps Y onto the J-penalized least-squares estimator with regularization parameter a, i.e. $\hat{u}_a = R_a Y$ where $\hat{u}_a$ is a solution of (1.6). According to [8], the MISE of $\hat{u}_a$ satisfies the bias-variance decomposition

$$\mathbf{E}\left(\left\|\hat{u}_a - u^\dagger\right\|^2\right) = \mathbf{E}\left(\|R_a\varepsilon\|^2\right) + \left\|\mathbf{E}\left(\hat{u}_a\right) - u^\dagger\right\|^2.$$

While the bias term $\left\|\mathbf{E}\left(\hat{u}_a\right) - u^\dagger\right\|^2$ on the right-hand side of this equation is typically not accessible as $u^\dagger$ is unknown, the variance term $\mathbf{E}\left(\|R_a\varepsilon\|^2\right)$ can be estimated by simulations.

To actually apply the Lepskij principle, we again fix a candidate set of regularization parameters $\{a_1, \ldots, a_M\}$ where $a_{i+1} = qa_i$ for some $q > 1$, for all $i =, 1 \ldots, M-1$, and

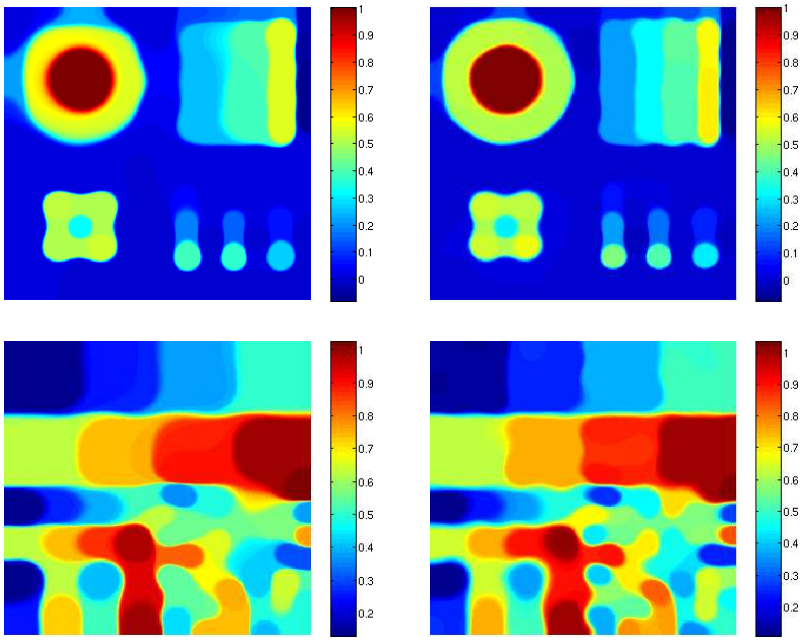

Figure 4.14: Comparison between SMR-estimation (left) and L-curve method (right) for datasets of Figure 4.12, i.e. $\sigma_K = 10$ and $\sigma = 0.05$. Top: "circles and bars". Bottom: "squares".

estimate

$$\Psi(j) := 2\mathbf{E}\left(\left\|R_{a_j}\varepsilon\right\|^2\right)$$

for all $j = 1, \ldots, M$ by Monte Carlo simulations. Using the notation of [78], we then choose the Lepskij parameter $a_{\mathrm{LEP}} = a_{\bar{j}}$ according to

$$\bar{j} := \max_{j=1,\ldots,M}\left\{\left\|\hat{u}_{\alpha_j} - \hat{u}_{\alpha_k}\right\|^2 \leq 2\Psi(j) \text{ for all } k \leq j\right\}.$$

We applied the Lepskij principle to Tikhonov regularization and compared its results to the SMR-estimators for the choice of $J = L^2$. As for the L-curve method, we provide both images and a simulation study. Note, however, that we abstain from giving the MSB in this study as $D^{\mathrm{sym}}_{L^2}(u, v) = \frac{1}{2}\|u - v\|^2$ and the MSB hence coincides with the MSE for $J = L^2$.

By visual inspection of the results shown in Figure 4.15, it becomes obvious that for the kernel with the rather small standard deviation of $\sigma_K = 4$, the SMR-estimator outperforms the Lepskij principle. In particular, the local-adaptivity in the reconstruction of "circles and bars" is convincing and leads to a much better reconstruction than achieved by means of

the Lepskij principle. This visual impression coincides with the results of our simulation study provided in Table 4.3.

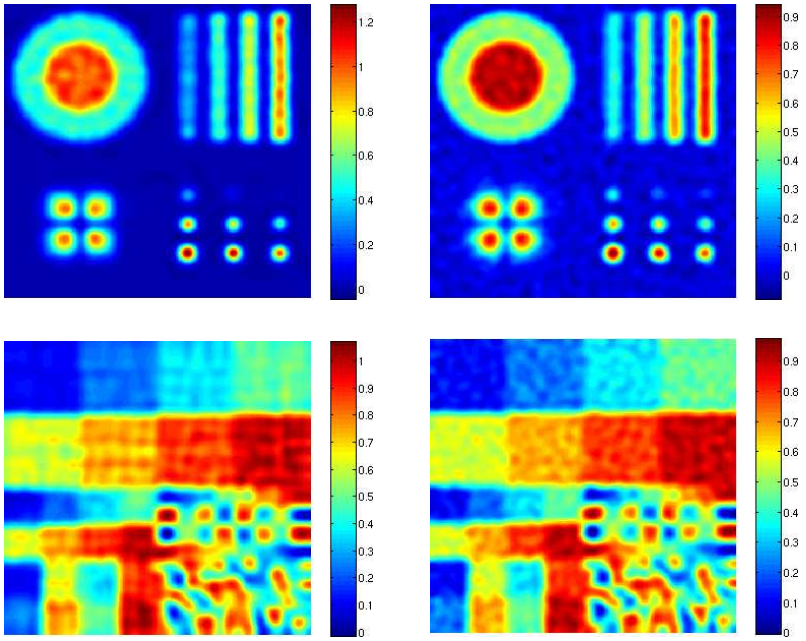

Figure 4.15: Comparison between SMR-estimation (left) and Lepskij principle (right) for datasets of Figure 4.11, i.e. $\sigma_K = 4$ and $\sigma = 0.1$. Top: "circles and bars". Bottom: "squares".

		$\sigma = 0.05$	$\sigma = 0.1$
		MSE	MSE
$\sigma_K = 2$	$\hat{u}_N(0.9)$	0.006	0.008
	$\hat{u}_{\text{LEP}}$	0.010	0.012
$\sigma_K = 5$	$\hat{u}_N(0.9)$	0.025	0.027
	$\hat{u}_{\text{LEP}}$	0.027	0.029

Table 4.3: Comparison between SMR-estimators and Lepskij principle, simulation study. True object was "circles and bars" resized to 128×128. Numbers are averaged over 100 simulations.

For the datasets corresponding to the choice of $\sigma_K = 10$, the Lepskij principle delivers results which are quite similar to the SMR-estimators, see Figure 4.16, yet the results of the Lepskij principle visually appear to be a bit smoother than the SMR-estimators. The

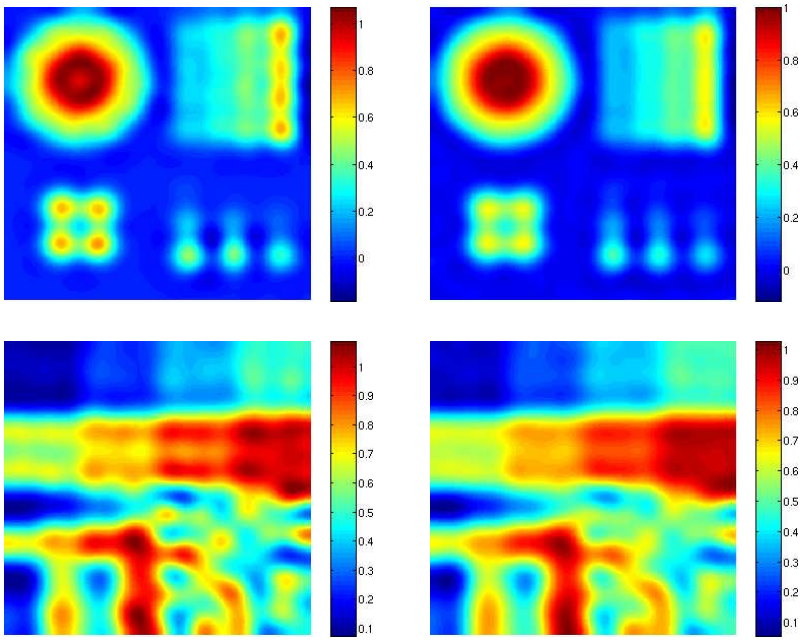

Figure 4.16: Comparison between SMR-estimation (left) and Lepskij principle (right) for datasets of Figure 4.12, i.e. $\sigma_K = 10$ and $\sigma = 0.05$. Top: "circles and bars". Bottom: "squares".

corresponding simulation study, however, shows that the MSE of the SMR-estimator is smaller than the one of the result of the Lepskij principle.

In summary, we see that SMR-estimation clearly outperforms the two methods we used for our comparison when applied to the datasets corresponding to the standard deviation $\sigma_K = 4$ of the convolution kernel. For the kernel with $\sigma_K = 10$, the differences to the results of the other methods are rather small. We conjecture that this is due to the rather extreme blurring operator being used there. As many of the objects' features are irretrievably lost in the observation Y, the different approaches we applied all fail to reconstruct them and only deliver a result which is rather far from the true object u^*. For this reason, SMR-estimation techniques cannot significantly add to the quality of the reconstruction in this setting.

4.2.3 Comparison to oracles

All datasets used for illustrations in this section so far are of synthetic nature. In particular, the underlying true objects u^* are known in contrast to practical applications. This knowledge allows for explicit computation of so-called *oracles* which correspond to a solution of

(1.6) with a regularization parameter that is optimal in a certain sense. This subsection starts out with a formal definition of oracles. Afterwards, such oracles will be compared to the SMR-estimators computed by our Augmented Lagrangian methodology.

In order to compute an oracle for a known object u^*, a certain number of observations $Y_1, \dots, Y_m$ is simulated according to the model (1.1). For each Y_i, the regularization parameter a_i in (1.6) is chosen such that some prefixed notion of distance $D : H_1 \times H_1 \to \mathbb{R}$ between the resulting estimator $\hat{u}_{a_i}$ and u^* is minimized. The oracle with respect to D is then defined as $\hat{u}_D := \hat{u}_{a_D}$ where the oracle parameter $a_D = \frac{1}{m} \sum_{i=1}^{m} a_i$ is the average of the parameters a_i. Oracles hence correspond to a choice of the regularization parameter a in

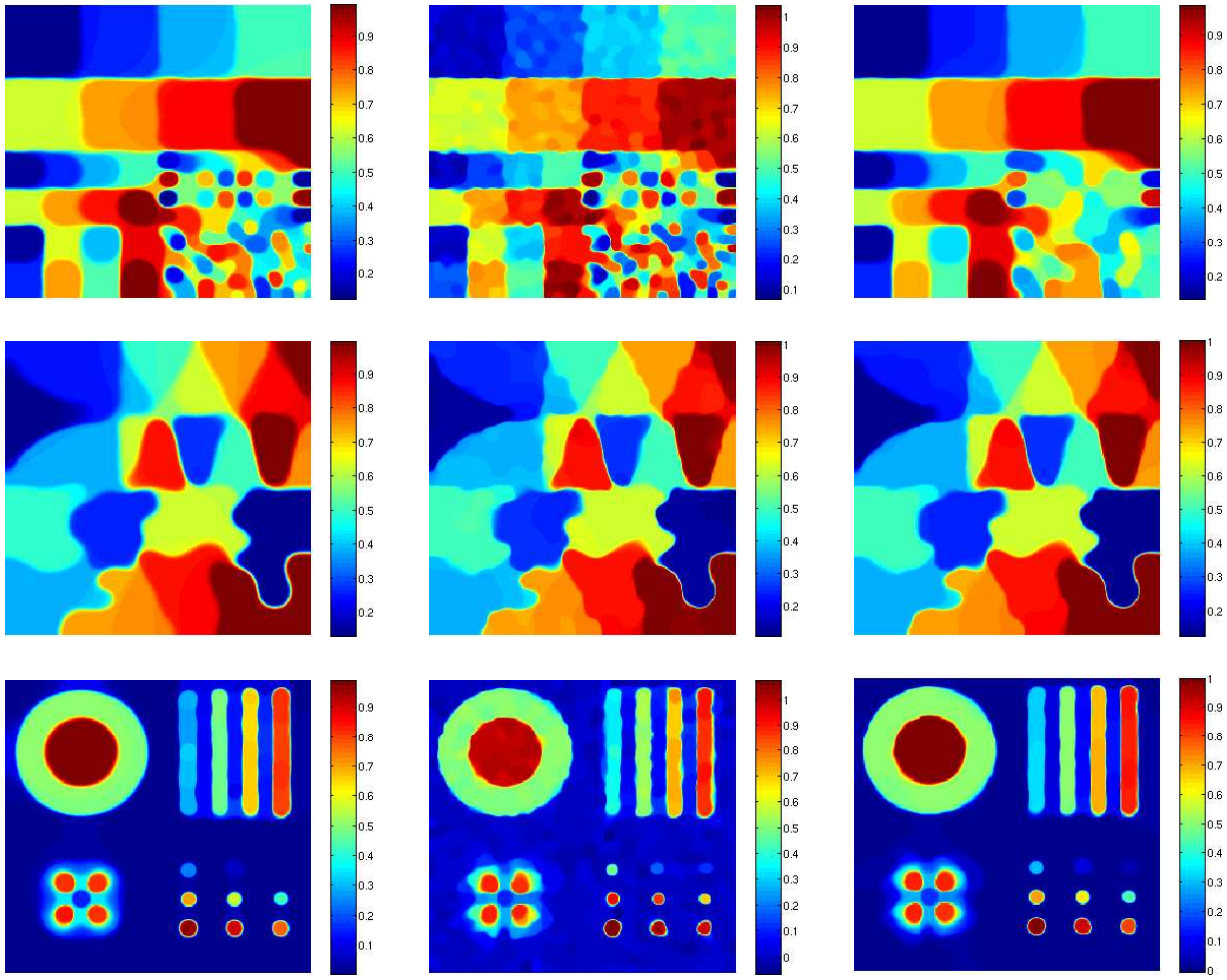

Figure 4.17: Comparison between SMR-estimators and oracles for the datasets of Figure 4.11, i.e. $\sigma_K = 4$ and $\sigma = 0.1$. Left: SMR-estimator. Middle: L^2-oracle. Right: Bregman-oracle.

(1.6) which is approximately optimal with respect to D. For our comparisons, we computed SMR-estimators for the choice of $J = \mathrm{TV}$. As distance measures, we used the L^2-norm $D = L^2$ and $D = D_{\mathrm{TV}}^{\mathrm{sym}}$, where the latter is the symmetric Bregman-divergence defined for general functions J in (4.2).

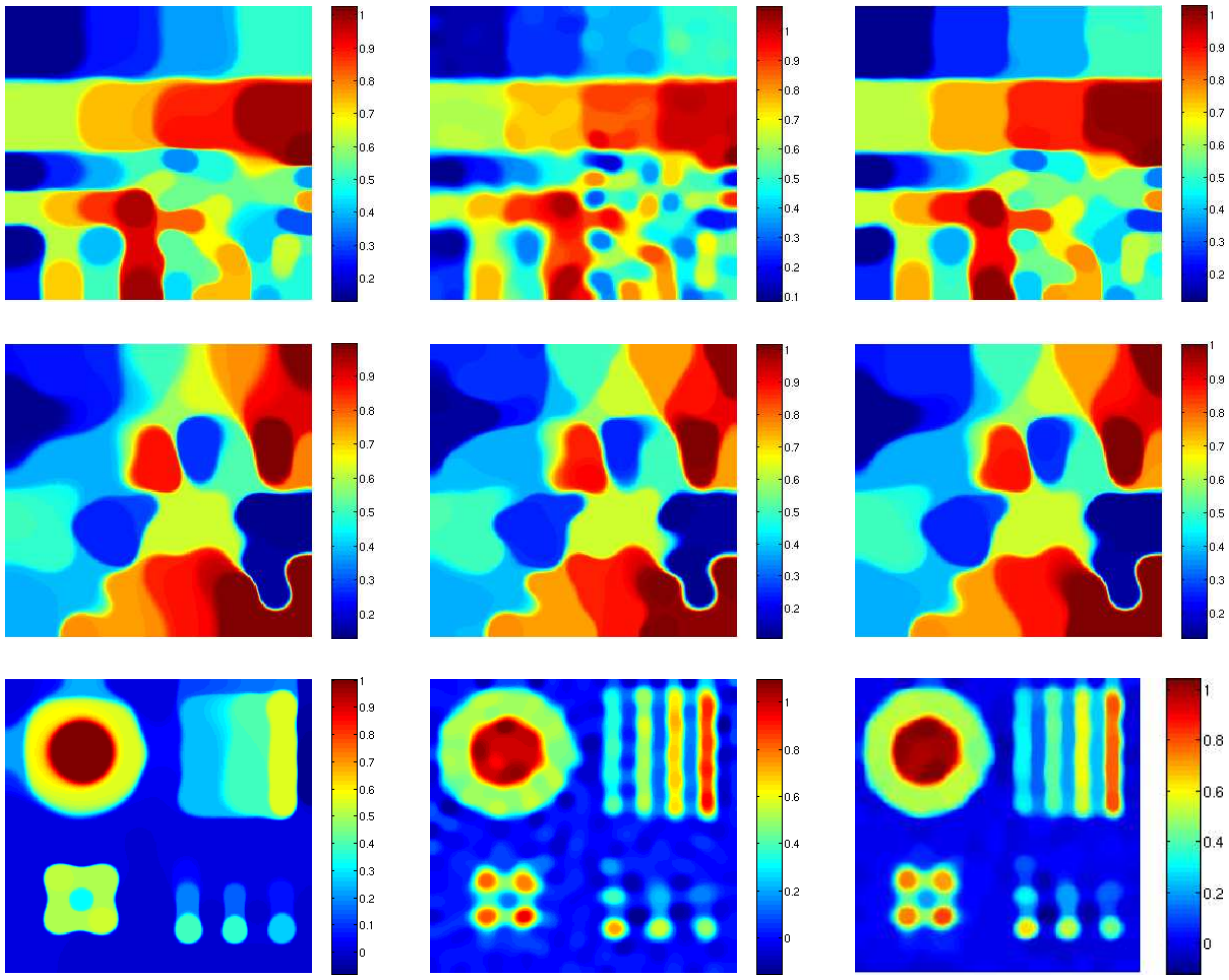

Figure 4.18: Comparison between SMR-estimators and oracles for the datasets of Figure 4.12, i.e. $\sigma_K = 10$ and $\sigma = 0.05$. Left: SMR-estimator. Middle: L^2-oracle. Right: Bregman-oracle.

In our experiments, we fixed $m = 10$ which appears to be sufficiently large in order to achieve stable results of a_D and computed the L^2- and Bregman-oracles for the datasets of Figures 4.11 and 4.12. The comparisons to the corresponding SMR-estimators are given in Figures 4.17 and 4.18. Visual inspection reveals that the SMR-estimators of the datasets corresponding to the smaller variance of the kernel (i.e. $\sigma_K = 4$ in (4.4)) in Figure 4.17

perform almost as good as or even better than the oracles. Especially the locally adaptive nature of the SMR-estimator for the "squares" object is convincing: While the L^2-oracle is considerably undersmoothed on larger features but catches most of the small squares in the bottom right, the Bregman-oracle is well smoothed on the bigger squares but fails to recover most of the small squares. The SMR-estimator, however, combines both of these advantages and is clearly superior to the (already globally optimal) oracles. For the "shapes" object, a similar effect - yet far less strong - becomes visible. Finally, for the reconstructions of the "circles and bars" object, the L^2-oracle is strongly undersmoothed, yet the Bregman-oracle performs slightly better than the SMR-estimator. For the kernel with bigger variance (i.e. $\sigma_K = 10$), the SMR-estimator looks quite similar to the oracles for the "shapes" object. The SMR-estimators for "squares" and "circles and bars", however, are inferior to the L^2- and Bregman-oracle, respectively.

At this point, we emphasize that when comparing a given estimator to oracles, one should always keep in mind that computation of the latter is based on the true object u^*. Choosing the regularization parameter optimal in the sense described above would not be possible if u^* was not accessible. Oracles therefore do not correspond to what is usually called an estimator. For this reason, we abstain from a comparison of SMR-estimators with oracles based on simulations and stick to visual inspection of the images. The use of prior knowledge about u^* makes oracles unrealistically strong in comparison to estimators that do not exploit this prior information. The quality of the SMR-estimators should hence still be considered as satisfactory despite their inferiority to oracles in some of our comparisons.

4.2.4 Fluorescence microscopy

In order to illustrate the performance of our approach in practical applications, we give an example from confocal microscopy. When recording images with this kind of microscope, the original object gets blurred by a Gaussian kernel and perturbed with Poisson noise. Moreover, the true object is always known a priori to exhibit nonnegative values only. In other words, the observations can be modelled according to (3.25) with the additional assumption that $u^* \geq 0$ holds pointwise. Therefore, combining the modification of the Augmented Lagrangian method to Poisson distributed noise and the incorporation of a nonnegativity constraint as described in Section 3.3 allows for the computation of SMR-estimators for such datasets.

The images depicted in the left column of Figure 4.19 show two recordings of PtK2 cells

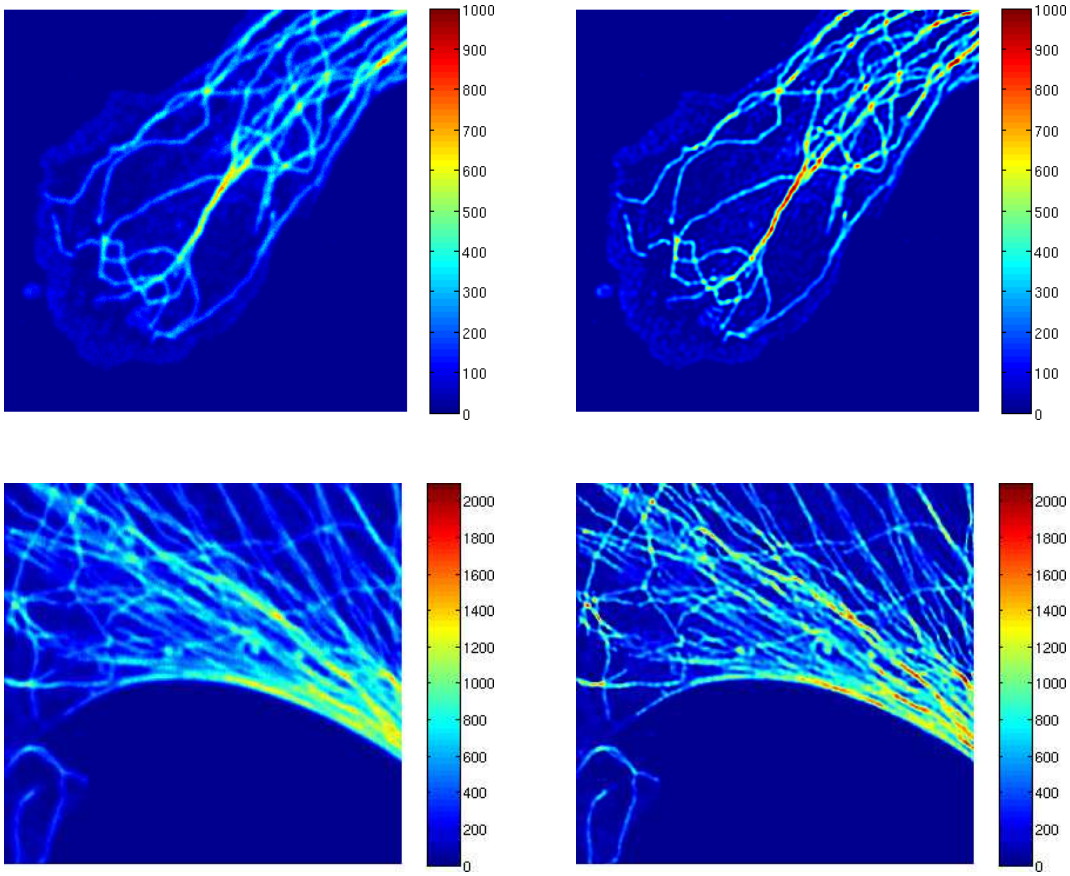

Figure 4.19: Confocal microscopy data. Left: fluorescence microscopy data of PtK2 cells in potorous tridactylus kidney. Right: SMR-estimator $\hat{u}_N(0.90)$.

taken from the kidney of potorous tridactylus. These datasets were kindly made available to us by the Department of NanoBiophotonics, Max Planck Institute for Biophysical Chemistry, Göttingen, Germany. Before the recording, the protein β-tubulin was tagged with a fluorescent marker such that it can be traced by the microscope. The images in Figure 4.19 show an area of 18×18 μm^2 at a resolution of 798×798 pixels. The point spread function of the optical system (i.e. the kernel k in (4.3)) can be modelled as a Gaussian kernel with full width at half maximum of 230nm which corresponds to $\sigma = 4.3422$ in (4.4). The SMR-estimators as computed by the Augmented Lagrangian method for these datasets are shown in the right column of the figure.

In the present situation we are in the delicate position to have a reference image at hand by means of which we can evaluate the result of our method: STED (STimulated Emission Depletion) microscopy constitutes a relatively new method that is capable of

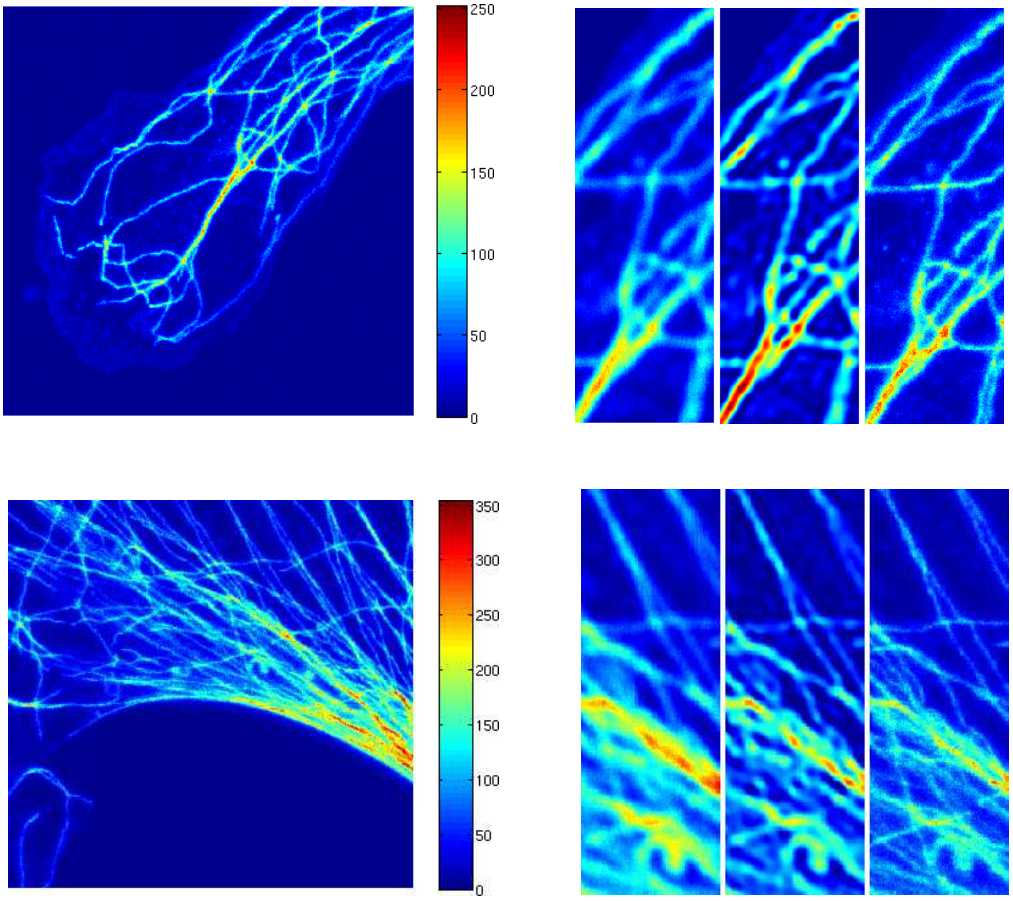

Figure 4.20: STED microscopy data. Left: STED microscopy recording of the PtK2 cell data sets. Right: detail comparisons between confocal recording (left), SMR-estimator $\hat{u}_N(0.90)$ (middle) and STED recording (right).

recording images at a remarkably high resolution. This method was first introduced by Hell and Wichmann in [61] (see also [60]) and research is currently advanced e.g. at the Department of NanoBiophotonics of the Max Planck Institute for Biophysical Chemistry in Göttingen. The left column of Figure 4.20 depicts STED recordings of the PtK2 cell data sets in Figure 4.19. Comparison of the SMR-estimator with the STED recordings in the right column of Figure 4.20 shows that our SMR-estimator technique chooses a reasonable amount of regularization: no artifacts due to undersmoothing are generated and on the other hand almost all (multiscale) geometrical features that are present in the high-resolution STED recording become visible in the reconstruction. The reconstruction quality of the SMR-estimator can hence be considered as satisfactory.

5 Discussion and outlook

This concluding chapter starts out with a summary of the results we have achieved. We will go through Chapters 2, 3 and 4, examining to which degree we reached the goals we set ourselves in the introductory Chapter 1. Afterwards we will give some ideas about possible future research in the field of multiresolution statistics focusing on extensions and improvements of the Augmented Lagrangian method introduced in Chapter 3.

5.1 Summary

In the introduction, we formulated the goal of finding an automated estimation scheme for the inverse problem model (1.1) which meets the following requirements:

1. fully data-driven,
2. statistically sound,
3. combinable with a wide range of penalty functions J,
4. locally and multiscale adaptive,
5. computationally feasible (especially for two-dimensional datasets).

Indeed, the MR-criterion and the resulting SMR-estimator as introduced in Sections 2.1 and 2.2 have a background which already establishes most of the desired attributes on a theoretical level. According to the interpretation given in Section 2.1, it is statistically sound. Moreover, the only degrees of freedom when computing SMR-estimators lie in the choice of a significance level α for the critical value $q_N(\alpha)$ in (2.5) and a dictionary Φ. Once these two choices have been fixed, arbitrarily many datasets on the same grid may be processed without changing any parameters individually. For this reason, we are justified in calling the SMR-estimator fully data-driven. Theorem 2.2.2 gives conditions on the penalty function J under which existence of the SMR-estimator is guaranteed. Unfortunately, verification of Assumption 1 made there is not straightforward for a given J. Nonetheless, it holds at least if J was chosen as total variation under fairly mild assumptions as we show in Theorem 2.4.1. Apart from this restriction, we also reached our goal of finding an estimator that is combinable with different penalty functions J.

All in all, the theoretical foundations laid in Chapter 2 are quite solid and motivate the development of a methodology which allows for numerical computation of SMR-estimators

in order to study how well the theoretical background translates into practical applications. Such an algorithmic approach to the actual computation of SMR-estimators is taken in Chapter 3.

There, we directly tackle the constrained minimization problem (2.6) by means of a technique from optimization theory, namely an Augmented Lagrangian method. We decided to use such a method for the reason that it is appealingly modular. In case one would like to use a different penalty function J or modify the statistic T_N, the method only needs to be altered in parts while the framework itself may be kept. This results from the corresponding intermediate steps (3.8) and (3.7) in Algorithm 2 being independent of the statistic and the penalty function being used, respectively.

On a theoretical level, we prove convergence of the method in Theorem 3.1.3 if combined with penalty functions J which fulfill the assumptions of Theorem 2.2.2. By means of this result, we found a method which theoretically enables us to compute SMR-estimators in a rather broad framework, especially allowing to use any penalty function J for which existence of SMR-estimators is guaranteed by means of Theorem 2.2.2.

In order to establish numerical computability, however, we still faced the problem of having to solve the quadratic program (3.7) within each iteration step of the Augmented Lagrangian method. As mentioned in Section 3.2, this problem is too large-scale to be successfully tackled by means of a general optimization method. The breakthrough towards computability of SMR-estimators was hence the application of Dykstra's algorithm to this problem. The option of explicitly projecting onto the single sets C_i in (3.17) in our framework makes the algorithm an efficient method to handle the huge number of side constraints in (3.7). In summary, application of Dykstra's algorithm in the context of the Augmented Lagrangian method guarantees computability of SMR-estimators in practice, for denoising problems as well as for inverse problems with non-trivial operators.

In Section 3.3, we indicated the versatility of the Augmented Lagrangian approach by presenting some possible extensions of the methodology. How to replace the statistic within the framework was indicated by using transformed residuals. As these still allow for explicit projections within Dykstra's algorithm, the resulting estimators were kept computable within reasonable time. As some applications assume an underlying data model which differs from the one stated in (1.1), we also studied how to employ our algorithmic to observations that are perturbed by Poisson rather than Gaussian distributed noise. By a simple transformation, we were able to adapt our framework to this situation. Nonetheless, the transformed

variables are only asymptotically normally distributed. Furthermore, as the true object is not accessible in practical applications, we use the current iterate as an approximation of it. These additional sources of possible errors show that the case of Poisson distributed noise still needs more research in order to come up with more sound approaches. Apart from the Poisson noise extension, we also demonstrated how our algorithm can be extended by an additional nonnegativity constraint; an extension that proves to be quite useful in many applications.

In Chapter 4, we presented results of our methodology which clearly show that SMR-estimators as computed by the Augmented Lagrangian method exhibit all attributes which were formulated as our goals before. Especially, the discretization of the parameter as an essential drawback of the methodology in [64] is avoided while the desired local adaptivity is improved. In Subsection 4.1.2, we also illustrated how this local adaptivity is established within the Augmented Lagrangian method. Moreover, we used additional applications - namely the denoising of natural images and the deconvolution of fluorescence microscopy recordings - to indicate how the algorithmic extensions of Section 3.3 may be employed in different situations.

To sum up, we see that the algorithmic of Chapter 3 guarantees for translation of the good theoretical properties of SMR-estimators into practice. Apart from the restrictions mentioned above, we have therefore achieved all goals which we set ourselves in the introduction.

5.2 Future work

We will now indicate how research on SMR-estimators might be carried on in the future. Several ideas to modify and extend the algorithmic framework presented in this thesis will be given in the following.

First of all, the rather general framework should be tested in additional practical situations. As integrating different penalty functions into the Augmented Lagrangian methodology is quite easy, experiments with such varying choices of J should be carried out. This should not only be done by performing tests on synthetic objects, but in practical situations as well. Apart from the application to fluorescence microscopy presented in this thesis, there are more situations in which linear inverse problems occur in practice and which hence allow for straightforward application of our algorithmic.

A second idea which would require minor modifications of our framework only is the use of alternative dictionaries. By restricting our considerations to characteristic functions of subsets and choosing these subsets from systems of squares, we already made a certain assumption on the geometry of features within the unknown object that are to be reconstructed. When creating systems of differently shaped geometric objects, however, prior information about the object may be exploited in special situations. Moreover, choosing a dictionary that consists of isotropic functions could even result in a certain independence of the shapes within the object. In both cases, one would have to come up with a method which guarantees fast evaluation of the MR-statistic (like the one for squares given in Section 2.3) in order to keep the runtime of the Augmented Lagrangian method - in particular of Dykstra's algorithm as a part of it - within reason. In addition, a decomposition of the resulting dictionaries into subsets which allow for simultaneous projections onto the feasible sets (as presented for the partitioning $\mathcal{P}_A$ in Subsection 3.2.2) should be provided.

Another starting-point for improvement and generalization is the MR-statistic itself. Using the statistic to test convex transformations of the residuals rather than the residuals themselves (see Section 3.3) can be quite useful in some situations as we illustrated by processing natural images in Section 4.1. Further transformations Λ could be developed and easily integrated into our framework by providing explicit projections onto the resulting sets $C_{\Lambda,i}$ in (3.23). In addition to such simple modifications, completely re-designed statistics for noise models which differ from (1.1) would also be desirable. While we adapted our methodology to the case of Poisson distributed noise in Section 3.3, the underlying transformation used for this purpose is clearly not satisfactory as it will be quite inexact

if the image of the true object under the operator K exhibits low intensities. Moreover, it leads to an additional approximation error as the true object is usually not accessible in practical situations. In order to cope with Poisson or other distributions, one would have to formulate a different idea on how to detect nonrandom structures in the residuals that is more sophisticated than simply linking it to the normal distribution by a variance stabilizing transformation.

Furthermore, applying the methodology to datasets of dimensions higher than two is an interesting idea. While the extension of the methodology itself to this case is straightforward, numerical and implementation details need special attention. In fact, we already made a first attempt to apply the Augmented Lagrangian method to three-dimensional datasets, but abstained from pushing our experiments any further as computation of SMR-estimators with our implementation simply took too long in this case. Nonetheless, a generalization to higher dimensions is possible and might be an issue in future work.

Improving the runtime of our algorithm not only for a possible application in higher dimensions but also for the two-dimensional case discussed in this thesis is another subject in this context. Improvements on an algorithmic level might for example be achieved by employing one of the parallelized versions of Dykstra's algorithm mentioned in Section 3.2 or by developing parallelized algorithms which compute the penalized least-square estimators as needed within the Augmented Lagrangian method in (3.8). The corresponding implementation might make use of multiple processors or of a graphics processing unit and save a vast amount of runtime. Apart from that, computation times might also be improved on a rather technical level by implementing the algorithmic more efficiently than we did for our experiments. As mentioned in several places, we paid close attention to write a solid implementation which is efficient enough to guarantee runtimes that allow for convenient numerical experiments. We do not claim, however, that our implementation could not be outperformed in terms of runtime. Still, our work primarily served as a proof of concept and did not aim at the development of a perfectly implemented software.

In summary, the methodology established in this thesis can be considered as a breakthrough in terms of computability and runtime in the field of SMR-estimation to a certain extent. Yet there are still open questions and further ideas which might be subject to future research.

Bibliography

[1] R. Acar and C. Vogel. Analysis of bounded variation penalty methods for ill-posed problems. *Inverse Problems*, 10:1217–1229, 1994.

[2] J. S. Arora, A. I. Chahande, and J. K. Paeng. Multiplier methods for engineering optimization. *International Journal for Numerical Methods in Engineering*, 32(7):1485–1525, 1991.

[3] A. Bakushinskii and A. Goncharsky. *Ill-Posed Problems: Theory and Applications*. Kluwer, Dordrecht, 1995.

[4] F. Bauer and T. Hohage. A Lepskij-type stopping rule for regularized Newton methods. *Inverse Problems*, 21(6):1975, 2005.

[5] H. H. Bauschke and A. S. Lewis. Dykstra's algorithm with Bregman projections: a convergence proof. *Optimization*, 48:409–427, 1998.

[6] D. P. Bertsekas. *Constrained optimization and Lagrange multiplier methods*. Computer Science and Applied Mathematics. Academic Press Inc. [Harcourt Brace Jovanovich Publishers], New York, 1982.

[7] E. G. Birgin and M. Raydan. Robust stopping criteria for Dykstra's algorithm. *SIAM J. Sci. Comput.*, 26(4):1405–1414, 2005.

[8] N. Bissantz, T. Hohage, A. Munk, and F. Ruymgaart. Convergence rates of general regularization methods for statistical inverse problems and applications. *SIAM J. Numer. Anal.*, 45(6):2610–2636, 2007.

[9] N. Bissantz, B. Mair, and A. Munk. A multi-scale stopping criterion for MLEM reconstructions in PET. *IEEE Nucl. Sci. Symp. Conf. Rec.*, 6:3376–3379, 2006.

[10] N. Bissantz, B. Mair, and A. Munk. A statistical stopping rule for MLEM reconstructions in PET. *IEEE Nucl. Sci. Symp. Conf. Rec.*, 8:4198–4200, 2008.

[11] J. P. Boyle and R. L. Dykstra. A method for finding projections onto the intersection of convex sets in Hilbert spaces. In *Advances in order restricted statistical inference (Iowa City, Iowa, 1985)*, volume 37 of *Lecture Notes in Statist.*, pages 28–47. Springer, Berlin, 1986.

[12] L. Bregman, Y. Censor, and S. Reich. Dykstra's algorithm as the nonlinear extension of Bregman's optimization method. *Journal of Convex Analysis*, 6(2):319–334, 1999.

[13] L. M. Bregman. A relaxation method of finding a common point of convex sets and its application to the solution of problems in convex programming. *Ž. Vyčisl. Mat. i Mat. Fiz.*, 7:620–631, 1967.

[14] L. M. Bregman, Y. Censor, S. Reich, and Y. Zepkowitz-Malachi. Finding the projection of a point onto the intersection of convex sets via projections onto half-spaces. *J. Approx. Theory*, 124(2):194–218, 2003.

[15] E. M. Bronšteĭn. ε-entropy of convex sets and functions. *Sibirsk. Mat. Ž.*, 17(3):508–514, 715, 1976.

[16] F. Bunea, A. Tsybakov, and M. Wegkamp. Aggregation and sparsity via ℓ 1 penalized least squares. In *Learning Theory*, volume 4005 of *Lecture Notes in Computer Science*, pages 379–391. Springer Berlin / Heidelberg, 2006.

[17] M. Burger and S. Osher. Convergence rates of convex variational regularization. *Inverse Problems*, 20(5):1411–1421, 2004.

[18] M. Burger, E. Resmerita, and L. He. Error estimation for Bregman iterations and inverse scale space methods in image restoration. *Computing*, 81(2-3):109–135, 2007.

[19] E. Candes and T. Tao. The Dantzig selector: statistical estimation when p is much larger than n. *Annals of Statistics*, 35(6):2313–2351, 2007.

[20] L. Cavalier and Y. Golubev. Risk hull method and regularization by projections of ill-posed inverse problems. *The Annals of Statistics*, 34(4):1653–1677, 2006.

[21] Y. Censor and G. T. Herman. On some optimization techniques in image reconstruction from projections. *Applied Numerical Mathematics*, 3(5):365 – 391, 1987.

[22] Y. Censor and S. Reich. The Dykstra algorithm with Bregman projections. *Communications in Applied Analysis*, 2:407–419, 1998.

[23] A. Chambolle. An algorithm for total variation minimization and applications. *J. Math. Imaging Vision*, 20(1–2):89–97, 2004.

[24] A. Chambolle and P. Lions. Image recovery via total variation minimization and related problems. *Numerische Mathematik*, 76(2):167–188, 1997.

[25] P. L. Combettes. Signal recovery by best feasible approximation. *IEEE Transactions on Image Processing*, 2:269–271, Apr. 1993.

[26] G. Crombez. Finding projections onto the intersection of convex sets in Hilbert spaces. *Numer. Funct. Anal. Optimiz.*, 16(5–6):637–652, 1995.

[27] I. Csiszár. Why least squares and maximum entropy? An axiomatic approach to inference for linear inverse problems. *Ann. Statist.*, 19(4):2032–2066, 1991.

[28] C. Cui, P. K. Lamm, and T. L. Scofield. Local regularization for n-dimensional integral equations with applications to image processing. *Inverse Problems*, 23(4):1611–1633, 2007.

[29] P. L. Davies and A. Kovac. Local extremes, runs, strings and multiresolution. *Ann. Statist.*, 29(1):1–65, 2001. With discussion and rejoinder by the authors.

[30] P. L. Davies, A. Kovac, and M. Meise. Nonparametric regression, confidence regions and regularization. *Ann. Statist.*, 37(5B):2597–2625, 2009.

[31] H. Dette, A. Munk, and T. Wagner. Estimating the variance in nonparametric regression - what is a reasonable choice? *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, 60(4):751–764, 1998.

[32] F. Deutsch and H. Hundal. The rate of convergence of Dykstra's cyclic projections algorithm: the polyhedral case. *Numer. Funct. Anal. Optim.*, 15(5-6):537–565, 1994.

[33] D. C. Dobson and C. R. Vogel. Convergence of an iterative method for total variation denoising. *SIAM J. Numer. Anal.*, 34(5):1779–1791, 1997.

[34] Y. Dong, M. Hintermüller, and M. Rincon-Camacho. Automated regularization parameter selection in a multi-scale total variation model for image restoration. Technical report, Institute of Mathematics and Scientific Computing, 2008. IFB Report 22.

[35] D. L. Donoho. Cart and best-ortho-basis: a connection. *Ann. Statist.*, 25(5):1870–1911, 1997.

[36] L. Dümbgen and V. G. Spokoiny. Multiscale testing of qualitative hypotheses. *Ann. Statist.*, 29(1):124–152, 2001.

[37] L. Dümbgen and G. Walther. Multiscale inference about a density. *Ann. Statist.*, 36(4):1758–1785, 2008.

[38] R. L. Dykstra. An algorithm for restricted least squares regression. *Journal of the American Statistical Association*, 78(384):837–842, 1983.

[39] P. Eggermont and V. LaRiccia. *Maximum Penalized Likelihood Estimation: Regression*. Springer Verlag, 2009.

[40] I. Ekeland and R. Temam. *Convex analysis and variational problems*, volume 1 of *Studies in Mathematics and its Applications*. North-Holland Publishing Co., Amsterdam-Oxford, 1976.

[41] H. W. Engl, M. Hanke, and A. Neubauer. *Regularization of Inverse Problems*. Kluwer Academic Publisher, Dordrecht Boston London, 1996.

[42] R. Escalante and M. Raydan. Dykstra's algorithm for a constrained least-squares matrix problem. *Numerical Linear Algebra with Applications*, 3(6):459–471, 1996.

[43] M. Fortin and R. Glowinski. *Augmented Lagrangian methods*, volume 15 of *Studies in Mathematics and its Applications*. North-Holland Publishing Co., Amsterdam, 1983. Applications to the numerical solution of boundary value problems, Translated from the French by B. Hunt and D. C. Spicer.

[44] K. Frick, P. Marnitz, and A. Munk. Shape constrained regularization by statistical multiresolution for inverse problems. arXiv:1003.3323, 2010.

[45] K. Frick and O. Scherzer. Regularization of ill-posed linear equations by the non-stationary Augmented Lagrangian Method. *J. Integral Equations Appl.*, 2010. accepted.

[46] N. Gaffke and R. Mathar. A cyclic projection algorithm via duality. *Metrika*, 36:29–54, 1989.

[47] E. M. Gertz and S. J. Wright. OOQP - object-oriented software for quadratic programming. http://pages.cs.wisc.edu/swright/ooqp/, 2008.

[48] E. Giusti. *Minimal Surfaces and Functions of Bounded Variation*, volume 80 of *Monographs in Mathematics*. Birkhäuser Verlag, Basel, 1984.

[49] R. Glowinski and P. Le Tallec. *Augmented Lagrangian and Operator-Splitting Methods in Nonlinear Mechanics*, volume 9 of *SIAM Studies in Applied Mathematics*. SIAM, Philadelphia, 1989.

[50] J. Gondzio. Multiple centrality corrections in a primal-dual method for linear programming. *Computational Optimization and Applications*, 6:137–156, 1995.

[51] M. Grasmair. The equivalence of the taut string algorithm and BV-regularization. *J. Math. Imaging Vision*, 27(1):59–66, 2007.

[52] S.-P. Han. A successive projection method. *Math. Program.*, 40(1):1–14, 1988.

[53] S.-P. Han and G. Lou. A parallel algorithm for a class of convex programs. *SIAM Journal on Control and Optimization*, 26(2):345–355, 1988.

[54] M. Hanke. Limitations of the L-curve method in ill-posed problems. *BIT Numerical Mathematics*, 36:287–301, 1996. 10.1007/BF01731984.

[55] P. C. Hansen. Analysis of discrete ill-posed problems by means of the L-curve. *SIAM Review*, 34(4):561–580, 1992.

[56] P. C. Hansen. *Rank-Deficient and Discrete Ill-Posed Problems: Numerical Aspects of Linear Inversion*, volume 4 of *Monographs on Mathematical Modeling and Computation*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 1998.

[57] P. C. Hansen. The L-curve and its use in the numerical treatment of inverse problems. In *Computational Inverse Problems in Electrocardiology, ed. P. Johnston, Advances in Computational Bioengineering*, pages 119–142. WIT Press, 2000.

[58] P. C. Hansen and D. P. O'Leary. The use of the L-curve in the regularization of discrete ill-posed problems. *SIAM J. Sci. Comput.*, 14(6):1487–1503, 1993.

[59] T. Hein. A unified approach for regularizing descretized linear ill-posed problems. *Mathematical Modelling and Analysis*, 14(4):451–466, 2009.

[60] S. W. Hell. Far-Field Optical Nanoscopy. *Science*, 316(5828):1153–1158, 2007.

[61] S. W. Hell and J. Wichmann. Breaking the diffraction resolution limit by stimulated emission: stimulated-emission-depletion fluorescence microscopy. *Opt. Lett.*, 19(11):780–782, 1994.

[62] M. R. Hestenes. Multiplier and gradient methods. *J. Optimization Theory Appl.*, 4:303–320, 1969.

[63] C. Hildreth. A quadratic programming procedure. *Naval Research Logistics Quarterly*, 4(1):79–85, 1957.

[64] T. Hotz, P. Marnitz, R. Stichtenoth, L. Davies, Z. Kabluchko, and A. Munk. Locally adaptive image denoising by a statistical multiresolution criterion. arXiv:1001.5447, 2010.

[65] K. Ito and K. Kunisch. Augmented Lagrangian methods for nonsmooth, convex optimization in Hilbert spaces. *Nonlinear Anal.*, 41(5-6, Ser. A: Theory Methods):591–616, 2000.

[66] A. N. Iusem. Augmented Lagrangian methods and proximal point methods for convex optimization. *Inverstigación Operativa*, 8:11–50, 1999.

[67] A. N. Iusem and A. R. de Pierro. On the convergence properties of Hildreth's quadratic programming algorithm. *Math. Program.*, 47(1):37–51, 1990.

[68] A. N. Iusem and A. R. De Pierro. On the convergence of Han's method for convex programming with quadratic objective. *Math. Program.*, 52(2):265–284, 1991.

[69] A. N. Iusem and R. G. Otero. Augmented Lagrangian methods for cone-constrained convex optimization in Banach spaces. *J. Nonlinear Convex Anal.*, 3:155–176, 2002.

[70] A. Kirsch. *An Introduction to the Mathematical Theory of Inverse Problems*. Springer, New York, Berlin, Heidelberg, 1996.

[71] E. Kolaczyk, J. Ju, and S. Gopal. Multiscale, multigranular statistical image segmentation. *JASA*, 100(472):1358–1369, 2005.

[72] E. Kolaczyk and R. Nowak. Multiscale likelihood analysis and complexity penalized estimation. *Annals of Statistics*, 32(2):500–527, 2004.

[73] P. K. Lamm. Variable-smoothing local regularization methods for first-kind integral equations. *Inverse Problems*, 19(1):195–216, 2003.

[74] P. K. Lamm and Z. Dai. On local regularization methods for linear Volterra equations and nonlinear equations of Hammerstein type. *Inverse Problems*, 21(5):1773–1790, 2005.

[75] O. V. Lepskij. On a problem of adaptive estimation in Gaussian white noise. *Theory of Probability and its Applications*, 35(3):454–466, 1990.

[76] J. Liu and P. Moulin. Complexity-regularized image denoising. *IEEE Transactions on Image Processing*, 10(6):841–851, 2001.

[77] E. Mammen and S. van de Geer. Locally adaptive regression splines. *Ann. Statist.*, 25(1):387–413, 1997.

[78] P. Mathé. The Lepskii principle revisited. *Inverse Problems*, 22(3):L11–L15, 2006.

[79] P. Mathé and S. V. Pereverzev. Discretization strategy for linear ill-posed problems in variable Hilbert scales. *Inverse Problems*, 19(6):1263–1277, 2003.

[80] P. Mathé and S. V. Pereverzev. Geometry of linear ill-posed problems in variable Hilbert scales. *Inverse Problems*, 19(3):789–803, 2003.

[81] S. Mehrotra. On the implementation of a primal-dual interior point method. *SIAM Journal on Optimization*, 2(4):575–601, 1992.

[82] M. Mendoza, M. Raydan, and P. Tarazaga. Computing the nearest diagonally dominant matrix. *Numerical Linear Algebra with Applications*, 5(6):461–474, 1998.

[83] M. Monsalve, J. Moreno, R. Escalante, and M. Raydan. Selective alternating projections to find the nearest SDD+ matrix. *Applied Mathematics and Computation*, 145(2-3):205 – 220, 2003.

[84] V. A. Morozov. *Methods for Solving Incorrectly Posed Problems*. CRC Press, Boca Raton, CA, 1993.

[85] A. Munk, N. Bissantz, T. Wagner, and G. Freitag. On difference-based variance estimation in nonparametric regression when the covariate is high dimensional. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, 67(1):19–41, 2005.

[86] J. Nocedal and S. J. Wright. *Numerical optimization*. Springer Series in Operations Research. Springer-Verlag, New York, 1999.

[87] M. Nussbaum and S. Pereverzev. The degree of ill-posedness in stochastic and deterministic noise models. Technical Report 509, WIAS, 1999. Preprint.

[88] C. Perkins. A convergence analysis of Dykstra's algorithm for polyhedral sets. *SIAM J. Numer. Anal.*, 40(2):792–804, 2002.

[89] J. Polzehl. R-package aws: Adaptive weights smoothing, 2006.

[90] J. Polzehl and V. G. Spokoiny. Adaptive weights smoothing with applications to image restoration. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, 62(2):335–354, 2000.

[91] M. J. D. Powell. A method for nonlinear constraints in minimization problems. In *Optimization (Sympos., Univ. Keele, Keele, 1968)*, pages 283–298. Academic Press, London, 1969.

[92] M. Raydan and P. Tarazaga. Primal and polar approach for computing the symmetric diagonally dominant projection. *Numerical Linear Algebra with Applications*, 9(5):333–345, 2002.

[93] E. Resmerita. On total convexity, Bregman projections and stability in Banach spaces. *J. Convex Anal.*, 11(1):1–16, 2004.

[94] E. Resmerita. Regularization of ill-posed problems in Banach spaces: convergence rates. *Inverse Problems*, 21(4):1303–1314, 2005.

[95] R. T. Rockafellar. A dual approach to solving nonlinear programming problems by unconstrained optimization. *Math. Programming*, 5:354–373, 1973.

[96] R. T. Rockafellar. Augmented Lagrange multiplier functions and duality in nonconvex programming. *SIAM Journal on Control*, 12:268–285, 1974.

[97] L. I. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Phys. D*, 60:259–268, 1992.

[98] D. O. Siegmund and E. S. Venkatraman. Using the generalized likelihood ratio statistic for sequential detection of a change-point. *The Annals of Statistics*, 23(1):255–271, 1995.

[99] D. O. Siegmund and B. Yakir. Tail probabilities for the null distribution of scanning statistics. *Bernoulli*, 6(2):191–213, 2000.

[100] R. Stichtenoth. *Signal and image denoising using inhomogeneous diffusion*. Ph.D. dissertation, Fachbereich Mathematik der Universität Duisburg – Essen, 2008.

[101] A. N. Tikhonov. On the solution of ill-posed problems and the method of regularization. *Dokl. Akad. Nauk SSSR*, 151:501–504, 1963.

[102] A. N. Tikhonov, A. V. Goncharsky, V. V. Stephanov, and A. G. Yagola. *Numerical Methods for the Solution of Ill-Posed Problems*. Kluwer Academic Press, Dordrecht, 1995.

[103] S. van de Geer. Least squares estimation with complexity penalties. *Mathematical Methods of Statistics*, 10(3):355–374, 2001.

[104] C. R. Vogel. *Computational methods for inverse problems*, volume 23 of *Frontiers in Applied Mathematics*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2002. With a foreword by H. T. Banks.

[105] C. R. Vogel and M. E. Oman. Iterative methods for total variation denoising. *SIAM J. Sci. Comput.*, 17(1):227–238, 1996.

[106] J. von Neumann. *Functional Operators Vol. II. The Geometry of Orthogonal Spaces*, volume 22 of *Annals of Mathematics Studies*. Princeton University Press, Princeton, N.J., 1950.

[107] S. Xu. Estimation of the convergence rate of Dykstra's cyclic projections algorithm in polyhedral case. *Acta Math. Appl. Sinica (English Ser.)*, 16(2):217–220, 2000.

[108] O. C. Zienkiewicz. *The Finite Element Method*. McGraw-Hill, London ; New York, 3d expanded and rev. ed. edition, 1977.

Curriculum Vitae

Dipl.-Math. Philipp Marnitz

born on May 3, 1981 in Hamburg
German citizen

1987-2000	Schooling *Abitur* at *Stormarnschule Ahrensburg*
2000-2001	Civilian service
2001-2007	Study of mathematics and computer science (minor) Faculty of Mathematics, University of Göttingen Diploma thesis: *Jacobisummen und Zertifikate im Primheitstest CPP* supervised by Prof. Dr. Preda Mihăilescu
2007-2010	Ph.D. studies in mathematics Institute for Mathematical Stochastics, University of Göttingen supervised by Prof. Dr. Axel Munk Associate member of the *DFG Graduate Program 1023*