Total Variation Restoration of Spatially Variant Blur

Additional Project for "Mathematical Image Processing", SS 2021

Ferdinand Vanmaele
Vanmaele@stud.uni-heidelberg.de

November 14, 2021

Abstract

In this project we try to deblur images where the blur is spatially variant - that is, depending on the position of the image - and do so in a computationally effective way. In the lecture, image deblurring was formulated as a convex optimization problem, and discretized with the *discrete convolution*, a linear operator A. If the blur is spatially *invariant*, the discrete convolution A is represented by a Toeplitz or block circulant matrix, and we can use the Discrete Fourier Transform (DFT) for efficient matrix operations. If the blur is spatially *variant*, A is non-Toeplitz in general.

In the papers [1, 2], the authors focus on the *alternate direction method of multipliers* (ADMM) to solve this problem. To reduce the problem size, the paper [1] finds an approximate solution in a low-dimensional *Krylov subspace*. We verify their implementation and introduce the necessary theoretical background.

1 Motivation and Overview

In image deblurring, we seek to recover the original, sharp image by using a mathematical model of the blurring process. This process is assumed to be *linear*: if f_1 and f_2 are the blurred versions of the exact images u_1 and u_2, then $f = \alpha f_1 + \beta f_2$ is the blurred image of $u = \alpha u_1 + \beta u_2$. Linear models can describe several physical mechanisms such as relative motion between the camera and subject (motion blur), bad focusing (defocusing blur), or a number of other mechanisms [2].

The continous image acquisition model is, in particular, given by the integral: [5]

$$f(s) = \int_{\mathbb{R}^2} p(s,t)u(t)\,\mathrm{d}t, \quad s \in Q \tag{1}$$

where p describes the PSF (*point-spread function*), u the exact image, f the observed (blurred) image, and $Q \subset \mathbb{R}^2$ the *field of view*. In the usual applications in astronomy, microscopy and photographic imaging, the PSF has finite width [9], i.e.

$$p(s,t) \approx 0 \quad \text{for} \quad \|s-t\| > \delta,\ \delta > 0.$$

Thus, we can restrict the integral to the *extended field of view* $B = Q + D_\delta$, where D_δ denotes a disc of radius δ centered at the origin. (Compare Figure 1) We now divide B into n_B identical pixels of area h (assumed $\equiv 1$), with n pixels inside the field of view:

$$f(s_l) = \sum_{k=1}^{n_B} p(s_l,t_k)u(t_k), \qquad 1 \le l \le n \tag{2}$$

We limit ourselves to estimating the image u from the data within the field of view Q. The observed image contains no information of the true scenery outside of Q. We thus use one of the following (artificial) boundary conditions: [4, 4.1 Basic Structures]

- The **zero boundary condition** is a good choice when the exact image is mostly zero outside the boundary—as is the case for many astronomical images with a black background. Unfortunately, the zero boundary condition has a bad effect on reconstructions of images that are nonzero outside the border.
- The **periodic boundary condition** is frequently used in image processing. This implies that the image repeats itself (endlessly) in all directions.
- In some applications it is reasonable to use a **reflexive boundary condition**, which implies that the scene outside the image boundaries is a mirror image of the scene inside the image boundaries.

Remark 1. When the blur introduced by the image acquisition process is *spatially invariant*, the PSF p does not depend on the spatial location s, and can be specified by:

$$p(s,t) = p(s-t) \tag{3}$$

and the sum becomes the two-dimensional convolution operation [1, 4.1].

Remark 2. In this project we assume the point-spread function is a *known* quantity, for example by deriving experimentally from point sources (such as fluorescent beads in microscopy [7]). If the blurring model is not known, or not with sufficient accuracy, we speak of *blind deblurring* [8].

1.1 Matrix formulation

We can arrange the elements of the images f and u into column vectors $vec(f)$ and $\text{vec}(u)$, by stacking an $M \times N$ image into an $n = MN$ vector. The equation (2) can then be written as a linear system:

$$\text{vec}(f) = A\,\text{vec}(u). \tag{4}$$

For images of sizes $M \times N$, the sizes of vectors are $MN \times 1$, and the size of the matrix A is $MN \times MN$. The construction of this matrix is described in detail in [4, 4. Structured Matrix Computations].

If p describes spatially invariant blur, the matrix A has a specific structure depending on the boundary conditions described above. In particular, A is *BCCB* (block circulant with circulant blocks) for periodic boundary conditions, and *Toeplitz-plus-Hankel* for reflexive boundary conditions. Matrix computations can then be implemented efficiently with the Discrete Fourier transform (DFT) and Discrete Cosine Transform (DCT), respectively [4, 4.2.2, 4.3].

With *spatially variant* blur, this is no longer the case. In this case, there is not a single PSF that can be used to represent the blurring operation; point sources in different locations of the image may result in different PSFs. In the most general case, there is a different PSF for each pixel in the image. Here we consider situations where there is a continuous and smooth variation in the PSF with respect to the position of the point source.

2 Problem Solving Approach

For most scenarios of practical interest, the matrix A formulated in 1.1 has a very high condition number, with singular values decaying to, and clustering at 0 [4, 1.4]. This leads to the following problems:

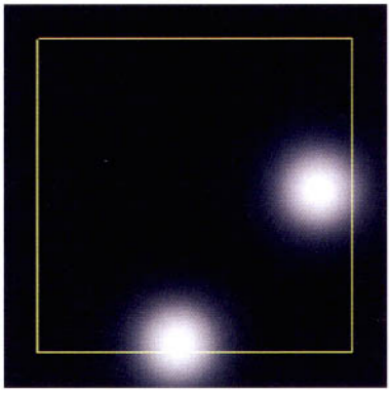

Figure 1: The PSF "spills over the edge" at the image boundary (line), so that the scene values outside the image affect what is recorded.

1. For the system $Ax = f$, noise in the data term f is amplified by the inversion of very small singular values of A (Figure 2).
2. Small perturbations in the matrix A (e.g. from differences between the estimated and true PSF) may lead to poor deblurring results.

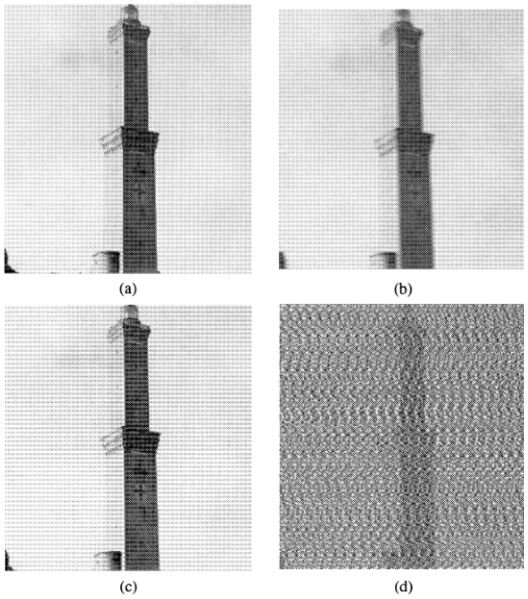

Figure 2: (a) Original image; (b) noise-free image, corrupted by linear motion blur; (c) solution obtained by inverse filtering in the noise-free case; (d) solution obtained by inverse filtering in the noisy case (white Gaussian noise added to the image in (b))

In this project, we approach the first issue by adding *regularizers*, in a constrained optimization problem of the form:

$$\min_{u \in \mathbb{R}^n} \phi(u) \quad \text{subject to} \quad \|Au - f\|_2 \leq \varepsilon \tag{5}$$

where $\phi : \mathbb{R}^n \to \overline{\mathbb{R}} = \mathbb{R} \cup \{-\infty, +\infty\}$ is a regularization function, and $\varepsilon \geq 0$ a parameter which depends upon the noise variance. This can be written as the unconstrained problem:

$$\min_{u \in \mathbb{R}^n} \phi(u) + \delta_{B(\varepsilon, f)}(Au) \tag{6}$$

where δ is the indicator function on the closed ball $B(\varepsilon, y)$ of radius ε, centered at y. To limit the dynamic range of the solution, we add the constraint $0 \leq x_i \leq 1$ for $i \in [n]$ [1, 4.2]. This results in the unconstrained problem:

$$\min_{u \in \mathbb{R}^n} J(u) := \min_{u \in \mathbb{R}^n} \phi(u) + \delta_{B(\varepsilon, f)}(Au) + \delta_{[0,1]^n}(u) \tag{7}$$

Popular choices for the regularizer are $\phi(u) = \|u\|_1$, the ℓ_1 norm, and

$$\phi(u) = TV(u) = \sum_{i=1}^{n} |u_i - u_{i-1}|,$$

the 1-dimensional discrete total variation.

Remark 3 (Existence of solution). If ϕ is proper, convex and lower semi-continuous (such as TV), then the functional J is proper, convex and lower semi-continuous (note that $B(\epsilon, f)$ is non-empty). J is also coercive, because $\delta_{[0,1]^n}$ is coercive by boundedness of the set $[0,1]^n$. It follows that problem (7) has a solution.

Remark 4. Restoration algorithms also consider the unconstrained problem:

$$\min_{u\in\mathbb{R}^n} \frac{1}{2}\|Au - f\|_2^2 + \tau\phi(u) \tag{8}$$

The formulation (5) has the benefit that the parameter ε has a clear meaning, i.e. it is proportional to the noise standard deviation [2, I-D].

2.1 Algorithms

If the regularizer is convex, problem (5) is convex, but the very high dimensions (often $\gg 10^6$) of f, and possibly u, preclude direct application of off-the-shelf optimization algorithms. Efficient restoration algorithms, such as those described in [2], make assumptions on the structure of A.

Specifically, spatially invariant blur with a periodic boundary condition results in a block circulant matrix. This matrix has a spectral decomposition $A = F^*\Lambda F$ given by the DFT matrix F (which, assuming $n = 2^k$, can be computed in $\mathcal{O}(n \log_2 n)$ time, see [4, 4.2.1].) This in turn allows many matrix computions to be performed efficiently.

In the case of spatially variant blur (that is, the PSF changes depending on the location of pixels in an observed image), the matrix A is no longer block-circulant, Toeplitz, or even symmetric. We may instead attempt to find an approximate solution in a m-dimensional subspace of $\mathbb{R}^n$, with $m \ll n$. The paper [1] uses this approach to construct a *Krylov subspace*, and formulates the minimization problem inside this subspace to reduce computational complexity.

In both papers [1, 2], the focus is on ADMM (or *alternate direction method of multipliers)* to solve the minimization problem. ADMM is described in [3] as "an algorithm that is intended to blend the decomposability of dual ascent with the superior convergence properties of the method of multipliers", and has a close relationship with Douglas-Rachford iterative schemes. In section 3, we describe these methods in detail.

3 Theoretical Properties

3.1 Alternate method of multipliers

The problem (6) can be seen as an unconstrained optimization problem of the form

$$\min_{u\in\mathbb{R}^n} f_1(u) + f_2(Gu) \tag{9}$$

where $G \in \mathbb{R}^{d\times n}$, $f_1 : \mathbb{R}^n \to \overline{\mathbb{R}}$, and $f_2 : \mathbb{R}^d \to \overline{\mathbb{R}}$.

Variable splitting is a simple procedure that consists in creating a new variable, say v, to serve as the argument of f_2, under the constraints that $v = Gu$: [2, II.A]

$$\min_{u\in\mathbb{R}^n, v\in\mathbb{R}^d} f_1(u) + f_2(v) \quad \text{subject to} \quad Gu - v = 0. \tag{10}$$

The rationale behind variable splitting is that it may be easier to solve the constrained problem (10) than to solve its equivalent unconstrained counterpart (9) We now form the *augmented Lagrangian* for this problem:

$$L_\mu(u, v, \lambda) = f_1(u) + f_2(v) + \lambda^T(Au - v) + \frac{\mu}{2}\|Gu - v\|_2^2$$

Note that if (u, v) is a feasible solution, then

$$L_\mu(u, v, \lambda) = f_1(u) + f_2(v).$$

ADMM minimizes L_μ through the iterations [3, 3.1]

$$\begin{aligned} u_{k+1} &\in \operatorname{argmin}_u L_\mu(u, v_k, \lambda_k), \\ v_{k+1} &\in \operatorname{argmin}_v L_\mu(u_{k+1}, v, \lambda_k), \\ \lambda_{k+1} &\in \lambda^k + \mu(Au_{k+1} - v_{k+1}). \end{aligned} \tag{11}$$

By setting the residual $r := Gu - v$ and the *scaled dual variable* $d := -\frac{1}{\mu}\lambda$, we get

$$\begin{aligned} \lambda^T r + \frac{\mu}{2}\|r\|_2^2 &= (\mu/2)\|r + (\mu/2)\lambda\|_2^2 - (1/2\mu)\|\lambda\|_2^2 \\ &= (\mu/2)\|r - d\|_2^2 - (\mu/2)\|d\|_2^2. \end{aligned}$$

The removal of constant factors then results in the *scaled formulation*

$$\begin{aligned} u_{k+1} &\in \operatorname{argmin}_u f_1(u) + \frac{\mu}{2}\|Gu - v_k - d_k\|_2^2, \\ v_{k+1} &\in \operatorname{argmin}_v f_2(v) + \frac{\mu}{2}\|Gu_{k+1} - v - d_k\|_2^2, \\ d_{k+1} &= d_k - (Gu_{k+1} - v_{k+1}). \end{aligned} \tag{12}$$

If f_2 is proper, convex and lsc, then the v^k update uniquely exists, and can be written using the *proximal operator*

$$v_{k+1} = \operatorname{prox}_{f_2/\mu}(Gu_{k+1} - d_k).$$

For problem (7), we describe the proximal operators in Section 3.2. Because it contains more than 2 summands, we generalize ADMM in Section 3.3 to handle J summands.

Remark 5 (Convergence). Under the assumptions that f_1 and f_2 are both closed, proper and convex, that A has full column rank, and that both subproblems are solved exactly, then ADMM is shown to converge. As long as the sequence of errors is absolutely summable, convergence is not compromised [2, Theorem 1]. The convergence rate is $\mathcal{O}(1/k)$, that is,

$$L_\mu(u^k) - L_\mu(u^*) = \mathcal{O}(1/k)$$

with k the iteration number [11]. We do not know of a convergence result for ADMM in m-dimensional (Krylov) subspaces.

3.2 Proximal operators

The proximal operator for $\delta(\cdot|S)$ is given as the projection on the set S. Therefore:

$$\text{prox}_{\delta(\cdot|B(\epsilon,f))}(s) = \begin{cases} f + \epsilon\frac{s-f}{\|s-f\|_2}, & \|s-f\|_2 > \epsilon, \\ s, & \|s-f\|_2 \le \epsilon, \end{cases}$$

and, component-wise,

$$\text{prox}_{\delta(\cdot|[0,1]^n))}(s)_i = \begin{cases} 0, & s_i \le 0, \\ s_i, & 0 < s_i < 1, \\ \beta, & s_i \ge 1. \end{cases}$$

The proximal operator for the 1-dimensional total variation can be computed with e.g. Condat's taut string algorithm [10].

3.3 Generalized ADMM

To approach problem (7), and eliminate the f_1 term in the u-update of (12), we consider a generalization of problem (9). Instead of two functions, we have J functions, that is [2, II.D]

$$\min_{u\in\mathbb{R}^d} \sum_{j=1}^{J} g_j(G^{(j)}u), \tag{13}$$

where $g_j : \mathbb{R}^{p_j} \to \overline{\mathbb{R}}$ are closed, proper and convex functions, $u \in \mathbb{R}^d$, and $G^{(j)} \in \mathbb{R}^{p_j\times d}$ are arbitrary matrices.

The minimization problem (13) can be written as the problem of two summands (9), using the following correspondences:

$$f_1 = 0, \quad G = \begin{bmatrix} G^{(1)} \\ \vdots \\ G^{(J)} \end{bmatrix} \in \mathbb{R}^{p\times d}$$

with $p = p_1 + \cdots + p_J$, and $f_2 : \mathbb{R}^{p\times d} \to \overline{\mathbb{R}}$ given by:

$$f_2(v) = \sum g_j(v^{(j)}), \quad v^{(j)} \in \mathbb{R}^{p_j}$$

Algorithm 1 Generalized ADMM for problem (13).

Choose $\mu > 0$, $v_0^{(1)}, \ldots, v_0^{(J)}$, and $d_0^{(1)}, \ldots, d_0^{(J)}$.
Repeat:
  $u_{k+1} \leftarrow \text{argmin}_u \|Gu - v_k - d_k\|_2^2$
  For $j = 1, \ldots, J$ **Do**:
    $v_{k+1}^{(j)} \leftarrow \text{prox}_{g_j/\mu}(G^{(j)}u_{k+1} - d_k^{(j)})$
    $d_{k+1}^{(i)} \leftarrow d_k^{(i)} - (G^{(j)}u_{k+1} - v_{k+1}^{(j)})$
  EndDo
  $k \leftarrow k + 1$
Until some stopping criterion is satisfied.

and $v = [(v^{(1)})^T, \ldots, (v^{(J)})^T]^T$. We now apply ADMM with

$$d_k = \begin{bmatrix} d_k^{(1)} \\ \vdots \\ d_k^{(J)} \end{bmatrix}, \quad v_k = \begin{bmatrix} v_k^{(1)} \\ \vdots \\ v_k^{(J)} \end{bmatrix}.$$

Note that:

- $f_1 = 0$ turns the u-update in ADMM into a least-squares problem, which has a unique solution if A has full column rank.
- The way of mapping problem (13) into problem (9) allows decoupling the minimization step in the v-update of ADMM into a set of J independent ones. That is,

$$v_{k+1} = \text{argmin}_v\, f_2(v) + \frac{\mu}{2}\|Gu_{k+1} - d_k\|_2^2$$

can be written as

$$\begin{bmatrix} v_{k+1}^{(1)} \\ \vdots \\ v_{k+1}^{(J)} \end{bmatrix} = \arg\min_{v^{(1)},\ldots,v^{(J)}} g_1(v^{(1)}) + \cdots + g_J(v^{(J)}) + \frac{\mu}{2}\left\| \begin{bmatrix} G^{(1)} \\ \vdots \\ G^{(J)} \end{bmatrix} u_{k+1} - \begin{bmatrix} v^{(1)} \\ \vdots \\ v^{(J)} \end{bmatrix} - \begin{bmatrix} d_k^{(1)} \\ \vdots \\ d_k^{(J)} \end{bmatrix} \right\|_2^2$$

This is summarized in Algorithm 1.

3.4 Krylov subspaces

In the optimization problem (7), the very high dimension n of the observed image u precludes a direct application of ADMM, as explained in Section 2.1. We thus wish to apply optimization problems in a *subspace* of dimensionality $m \ll n$.

Definition 6 ([6, 6.2])**.** The **Krylov subspace** of order m, denoted by $\mathcal{K}_m$, is the vector space spanned by $v \in \mathbb{R}^n$ and the first $m-1$ powers of $A \in \mathbb{R}^{n\times n}$:

$$\mathcal{K}_m(A, v) = \text{span}\{v, Av, A^2v, \ldots, A^{m-1}v\}.$$

The Krylov subspaces form an increasing family of subspaces, with dimension bounded by n. It is the subspace of all vectors in $\mathbb{R}^n$ which can be written as $x = p(A)v$, where p is a polynomial of degree not exceeding $m-1$. In practice, constructing the Krylov subspaces amounts to determining their basis. The power basis $(v, Av, A^2v, \dots, A^{m-1}v)$ converges to (a multiple of) an eigenvector that corresponds to the largest eigenvalue of A, in absolute value. Therefore we construct an orthonormal basis $\{v_1, v_2, \dots, v_m\}$ as follows.

Arnoldi's method applies Gram-Schmidt orthonormalization procedure to the vectors obtained by successive products of the matrix A. Algorithm 2 [6, 6.3.2] gives a practical implementation, using the modified Gram-Schmidt procedure for improved numerical stability.

Algorithm 2 Arnoldi(A, f, m, u_0)

$r_0 \leftarrow f - Au_0$
$q_1 \leftarrow r_0/\|r\|_2$
For $i = 1, 2, \dots, m$ **Do**:
  $v \leftarrow Aq_i$
  For $j = 1, \cdots, i$ **Do**:
    $h_{j,i} \leftarrow q_j^T v$
    $v \leftarrow v - h_{j,i} q_j$
  EndDo
  $h_{i+1,i} \leftarrow \|v\|_2$. **If** $h_{i+1,i} = 0$ **Stop**
  $q_{i+1} \leftarrow v/h_{i+1,i}$
EndDo

Proposition 7 (*[6, 6.5]*)**.** The following relation holds for the Arnoldi basis:

$$AQ_m = Q_{m+1} H_m, \tag{14}$$

where Q_m denotes the $n \times m$ matrix with orthonormal column vectors $v_1, \dots, v_m$, and H_m the $(m+1) \times m$ upper Hessenberg matrix with nonzero entries h_{ij}.

For a linear system $Au = f$, Krylov subspace methods extract an approximate solution u_m from the affine subspace $u_0 + \mathcal{K}_m(A, r_0)$ of dimension m, by imposing the condition $b - Au_m \perp \mathcal{L}_m$. Here, $\mathcal{L}_m$ is another subspace of dimension m, u_0 represents an arbitrary initial guess to the solution, and $r_0 = b - Au_0$ is the initial residual.

For example, if $\mathcal{L}_m = A\mathcal{K}_m$, the method minimizes the 2-norm of the residual $b - Ax$ over $x \in x_0 + \mathcal{K}_m$ [6, Proposition 5.3]. Such a method (e.g. GMRES) can work with general non-symmetric matrices A, provided they are non-singular.

We can now state the following proposition.

Proposition 8 ([1, Proposition 3])**.** Under the assumption $u \in u_0 + \mathcal{K}_m(A, r_0)$, the least squares problem on line 3 of Algorithm 1 is reduced to

$$\min_{\alpha \in \mathbb{R}^m} \left\| \begin{bmatrix} H_m \\ \overline{G}_2 \\ \vdots \\ \overline{G}_J \end{bmatrix} \alpha - \begin{bmatrix} Q_{m+1}^T(v_1 + d_1) \\ Q_m^T(v_2 + d_2) \\ \vdots \\ Q_m^T(v_J + d_J) \end{bmatrix} \right\|_2^2 \tag{15}$$

where $\overline{G}_j := Q_m^T G_j Q_{+m} \in \mathbb{R}^{m \times m}$ for $j = 2, \dots, J$.

Proof. Follows from relation (14) and orthogonality of Q_m. □

Algorithm 3 KADMM for TV restoration

Choose $\mu > 0$, $v_1, \dots, v_3 \in \mathbb{R}^{MN}$, $d_1, \dots, d_3 \in \mathbb{R}^{MN}$, $k = 0$.
$Q_{m+1}, H_m \leftarrow$ Arnoldi$(A, f, m, 0)$
Repeat:
  $\alpha \leftarrow \operatorname{argmin}_\alpha \left\| \begin{bmatrix} H_m \\ I_m \\ I_m \end{bmatrix} \alpha - \begin{bmatrix} Q_{m+1}^T(v_1 + d_1) \\ Q_m^T(v_2 + d_2) \\ Q_m^T(v_3 + d_3) \end{bmatrix} \right\|_2^2$
  $v_1 \leftarrow \operatorname{prox}_{\delta(\cdot | B(\epsilon, f))}(Q_{m+1} H_m \alpha - d_1)$
  $d_1 \leftarrow d_k^{(1)} - (Q_{m+1} H_m \alpha - v_1)$
  $v_2 \leftarrow \operatorname{prox}_{TV/\mu}(Q_m \alpha - d_2)$
  $d_2 \leftarrow d_2 - (Q_m \alpha - v_2)$
  $v_3 \leftarrow \operatorname{prox}_{\delta(\cdot | [0,1]^n)}(Q_m \alpha - d_3)$
  $d_3 \leftarrow d_3 - (Q_m \alpha - v_3)$
Until some stopping criterion is satisfied.

3.5 TV restoration in Krylov subspace

We now combine the results of the previous sections. Consider again the unconstrained problem (7) with $\phi = TV$:

$$\min_{u \in \mathbb{R}^n} TV(u) + \delta_{B(\varepsilon, f)}(Au) + \delta_{[0,1]^n}(u)$$

Formulated in Krylov subspace $\mathcal{K}_m(A, f)$, with initial solution $u_0 = 0$ and matrix basis Q_m, the problem becomes:

$$\min_{\alpha \in \mathbb{R}^m} \delta_{B(\varepsilon, f)}(Q_{m+1} H_m \alpha) + TV(Q_m \alpha) + \delta_{[0,1]^m}(Q_m \alpha) \tag{16}$$

using the identity $AQ_m = Q_{m+1} H_m$ for the first summand. Algorithm 1 is applied to this problem by setting

$$G^{(1)} = Q_{m+1} H_m, \quad G^{(2)} = G^{(3)} = Q_m,$$

and applying Proposition 8. Q_{m+1} and H_m are constructed with Arnoldi (Algorithm 2) from the blur matrix $A \in \mathbb{R}^{MN \times MN}$. This results in Algorithm 3.

3.6 Complexity analysis

Let $n = MN$ be the dimension of the observed image, and m the order of the Krylov subspace. Let k be the number of iterations. We observe the following for Algorithm 3:

n	Elapsed time	$n_{\max}/n$	$t_{\max}/t$
1617	0.0807s	95.4861	95.2307
9801	0.5342	15.7536	14.3783
38801	1.9732	3.9793	3.8928
87001	4.9271	1.7747	1.5590
154401	7.6811	1	1

Table 1: Runtimes (n) for 100 (k) iterations, using parameters in Section 4. The table shows the elapsed time to be proportional to the problem size n.

- Constructing the Arnoldi basis via Gram-Schmidt orthogonalization takes $\mathcal{O}(m^2 n)$ operations.
- In each iteration, a least squares problem of size $(3m+1) \times m$ is solved, with $\mathcal{O}((3m+1)^2 m)$ operations (using QR decomposition).

  This is much smaller than the least squares problem from a direct application of ADMM, with size $3n \times n$ and complexity $\mathcal{O}(9n^3)$.
- The product $Q_m \alpha$ takes mn operations. In each iteration, the result can be stored and reused.
- The product $Q_{m+1} H_m \alpha$ requires $\mathcal{O}(2m^2 + 2mn)$ operations, compared to $\mathcal{O}(2mn + 2n^2)$ for the product $AQ_m\alpha$.
- The projection onto $B(\epsilon, f)$ and $[0,1]^n$ takes $\mathcal{O}(n)$ operations.
- Condat's taut string algorithm has worst-case complexity $\mathcal{O}(n^2)$, with $\mathcal{O}(n)$ "in all practical situations".

In short, we have $\mathcal{O}(m^2 n)$ for the Arnoldi basis, and $\mathcal{O}(km^3 + kn)$ for the ADMM iterations. Evidently, for k sufficiently large, the complexity of ADMM dominates. For $m \ll n$, the algorithm then has complexity $\mathcal{O}(kn)$. For an experimental verification, see Table 1.

4 Numerical Experiments

All methods were implemented using MATLAB on an Intel Core i5-2520 CPU. Images from the BSDS500 data set are blurred with spatially variant blur (see below), with periodic boundary conditions. Gaussian noise is added, with noise variance σ^2 determined by the BSNR (*blurred signal-to-noise ratio*), defined in [1] as:

$$\mathrm{BSNR} = 10 \log \frac{\mathrm{var}(Au)}{\sigma^2} \quad \Leftrightarrow \quad \frac{\mathrm{var}(Au)}{\exp\left(\frac{1}{10}\mathrm{BSNR}\right)} = \sigma^2$$

4.1 Spatially variant Gaussian blur

The spatially variant Gaussian blur model in [12] is used for the restoration experiments. The blur kernel at the pixel location $s = (s_1, s_2)$ in the image acquisition model is obtained from the two-dimensional separable Gaussian function

$$p(s,t) = \frac{1}{\xi_1 \xi_2} \exp\left(-\frac{1}{2} \frac{(t_1 - s_1)^2}{\sigma_x^2(s_1)} + \frac{(t_2 - s_2)^2}{\sigma_y^2(s_2)} \right),$$

with normalizing constants

$$\xi_1 = \sqrt{2\pi}\sigma_x(s_1), \quad \xi_2 = \sqrt{2\pi}\sigma_y(s_2).$$

The blur kernel changes depending on the location x, and hence the blur is spatially variant.

Remark 9. $p(x,y)$ is seperable, that is,

$$p(s,t) = p_1(s_1,t_1)\, p_2(s_2,t_2)$$

holds with

$$p_1(s_1,t_1) = \frac{1}{\xi_1} \exp\left\{ -\frac{1}{2} \frac{(t_1 - s_1)^2}{\sigma_x^2(s_1)} \right\}$$
$$p_2(s_2,t_2) = \frac{1}{\xi_2} \exp\left\{ -\frac{1}{2} \frac{(t_2 - s_2)^2}{\sigma_y^2(s_2)} \right\}$$

This means we can represent the matrix A as a Kroneker product $A_1 \otimes A_2$, with A_1 and A_2 the 1-dimensional blur matrices [4, 4.1.1] for a given boundary condition [12, 2.2.5].

In [1], three types of spatially variant blur are considered. The first type has a smaller amount of blur at the center and a larger amount of blur at the corners of an image. The variances are given by:

$$\sigma_x(s_1) = \gamma |0.5 - s_1/M| + 0.5,$$
$$\sigma_y(s_2) = \gamma |0.5 - s_2/N| + 0.5.$$

The second type is the opposite of the first case, with more severe blur at the center and milder blur at the corners. The variances are given by

$$\sigma_x(s_1) = -\gamma |0.5 - s_1/M| + 2.5,$$
$$\sigma_y(s_2) = -\gamma |0.5 - s_2/N| + 2.5.$$

The third type has smaller amount of blur at the lower right corner and larger amount of blur at the upper left part of an image. The variances are given by

$$\sigma_x(s_1) = \gamma (0.5 - s_1/M)/2 + 1.25,$$
$$\sigma_y(s_2) = \gamma (0.5 - s_2/N)/2 + 1.25.$$

The scalar γ is set fixed to 4, and 21×21 kernel size is used for discretization of p.

4.2 Image restoration

For the three types of blur described in 4.1, we applied KADDM to images with varying degrees of noise.

To measure the quality of the restoration, we used the mean squared error between restored image $\hat{u}$ and original image u,

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(u_i - \hat{u}_i).$$

Parameters for Algorithm 3 were chosen as follows.

- The initial values for v_i and d_i are sampled from the uniform distribution $U[0, 1]$.
- The order m of the Krylov subspace is set to 16, to limit memory requirements.
- Relative change of residuals was used for the termination criterium, i.e. $\|r^i - r^{i-1}\|/\|r\| < \text{TOL}$, with a tolerance of 10^{-6} and a maximum amount of iterations of 10,000.
- The authors in [1] suggest $\epsilon = \sqrt{\tau MN}\sigma$ using Morozov's discrepancy principle, with $\tau = 1$. Following [13], we used a fixed $\tau < 1$ to avoid an oversmoothed image.

A comparison of degraded and restored images for each blur type is given in Figures 3, 4, 5. KADMM managed to deblur the images to a certain degree, with an MSE of order of magnitude 10^{-3}. Increasing the maximum iteration count (to 100,000) did not lead to a significant decrease in MSE, nor did setting ϵ to a low ($\approx$TOL) value; in the presence of noise, latter lead to a "speckle pattern" in the restored images.

ID	Blur	Iterations	Time	MSE
1	1	7086	382.5s	$1.199 \cdot 10^{-3}$
	2	7171	395s	$5.23 \cdot 10^{-4}$
	3	5490	296s	$7.53 \cdot 10^{-4}$
50	1	5186	365s	$3.370 \cdot 10^{-3}$
	2	4991	347.5s	$2.594 \cdot 10^{-3}$
	3	3458	248s	$2.695 \cdot 10^{-3}$
103	1	5509	378s	$2.223 \cdot 10^{-3}$
	2	4895	326s	$2.492 \cdot 10^{-3}$
	3	4532	272s	$1.855 \cdot 10^{-3}$

Table 2: Numerical results for images 1, 50 and 103 from BSDS500, with spatially variant (types 1-3) and invariant (IV) blur, and Gaussian noise (60dB BSNR).

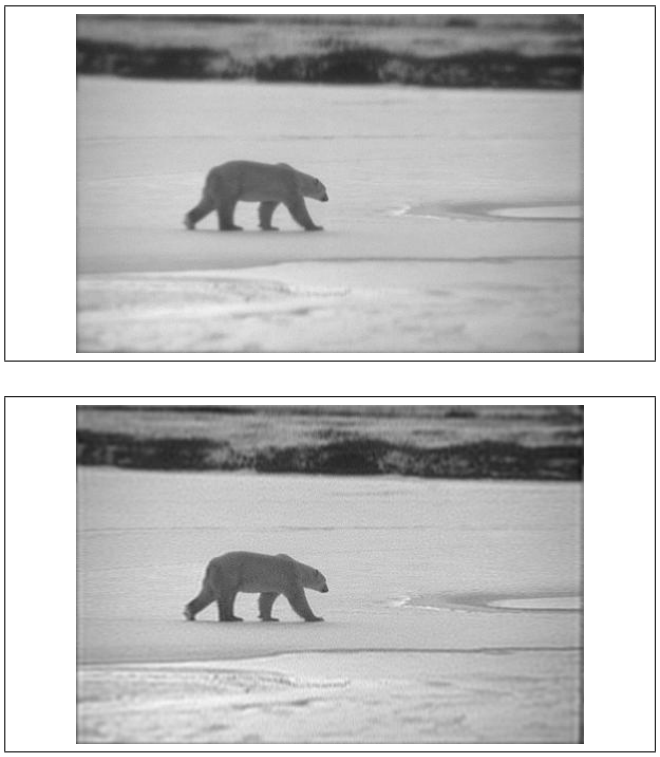

Figure 3: Degraded (top) vs. restored (bottom) image after 7086 KADMM iterations (blur type 1).

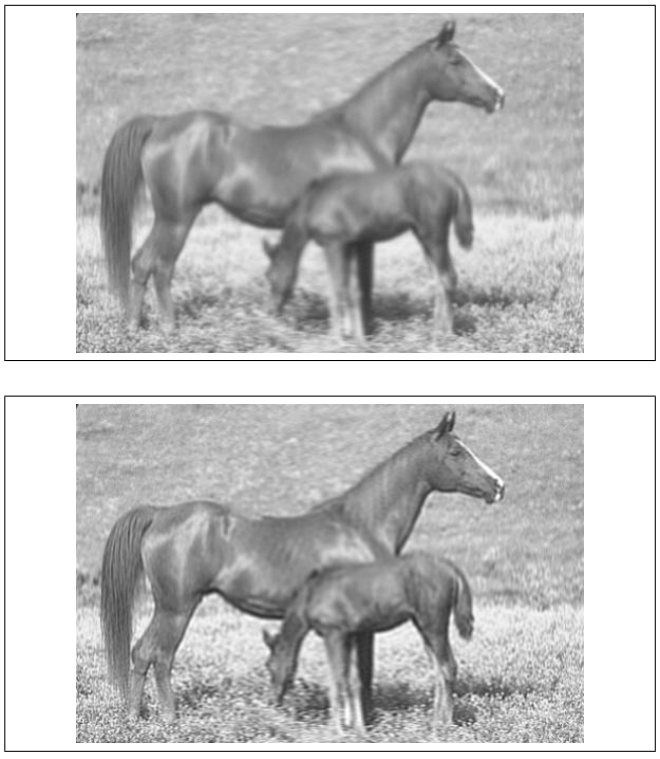

Figure 4: Degraded (top) vs. restored (bottom) image after 4991 KADMM iterations (blur type 2).

5 Conclusion

This project presents a convex optimization framework based on ADMM and Krylov subspace methods. The proposed KADMM finds an inexact solution by restricting the solution space to a Krylov subspace of small order. The basis of the Krylov subspace is found by the Arnoldi process, and ADMM is applied in the Krylov

Figure 5: Degraded (top) vs. restored (bottom) image after 4532 KADMM iterations (blur type 3).

subspace. The computational complexity of the algorithm is then linear in the image size and number of iterations. Unlike restoration methods for spatially invariant blur, which rely on Fourier transforms for computational efficiency, the method is applicable to general, non-symmetric matrices A.

Experiments on restoration of spatially variant blurred and noised images show that, assuming full knowledge of the blurring operator and noise variance, the image can be restored to a certain degree. Improved results can likely be achieved with a careful hyperparameter study, e.g. on the regularization parameter μ for the TV proximal operator [14].

References

[1] J.D. Yun, S. Yang. *ADMM in Krylov Subspace and Its Application to Total Variation Restoration of Spatially Variant Blur*, SIAM J. Imaging Sciences, Vol. 10, No. 2, pp. 484-507, 2017.

[2] M.V. Afonso, J.M. Bioucas-Dias, M. A. T. Figueiredo. *An Augmented Lagrangian Approach to the Constrainted Optimization Formulation of Imaging Inverse Problems*, IEEE Transactions on Image Processing, Vol. 20, No. 3, March 2011.

[3] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein. *Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers*, Foundations and Trends in Machine Learning, Vol. 3, No. 1, 2010.

[4] P. C. Hansen, J. G. Nagy, D. P. O'Leary. *Deblurring Images – Matrices, Spectra and Filtering*, SIAM, 2006.

[5] J. G. Nagy, D. P. O'Leary. *Restoring images degraded by spatially variant blur*, SIAM J. Sci. Comput., Vol. 19, No. 4, pp. 1063-1082, July 1998.

[6] Y. Saad. *Iterative Methods for Sparse Linear Systems*, SIAM, 2003.

[7] N. Stuurman. *Measuring a Point Spread Function*, iBiology, 2012. `https://www.ibiology.org/talks/measuring-a-point-spread-function/`

[8] M.S.C. Almeida, L.B. Almeida. *Blind and Semi-Blind Deblurring of Natural Images*, IEEE Transactions on Image Processing, Vol. 19, No. 1, January 2010.

[9] D. Calvetti, E. Somersalo. *Bayesian image deblurring and boundary effects*, Advanced Signal Processing Algorithms, Architectures and Implementations, 2005.

[10] L. Condat. *A Direct Algorithm for 1D Total Variation Denoising*, IEEE Signal Proc. Letters, Vol. 20, No. 11, pp. 1054-1057, 2013.

[11] B. He, X. Yuan. *On the $O(1/n)$ convergence rate of the Douglas-Rachford alternating direction method*, SIAM J. Numer. Anal., Vol. 50, No. 2, pp. 700-709, 2012.

[12] S. Berisha, J.G. Nagy. *Iterative methods for image restoration*, Academic Press Library in Signal Processing, Volume 4, pp. 193-247, 2014.

[13] H. Chuan, H. Chuang-Hua, Z. Wei, S. Biao. *Box-constrained Total-variation Image Restoration with Automatic Parameter Estimation*, Acta Automatica Sinica, Vol. 40 No.8, August 2014.

[14] C. Deledalle, S. Vaiter, G. Peyré, J.M. Fadili, C. Dossal. *Unbiased Risk Estimation for Sparse Analysis Regularization*, Proc. ICIP'12, Sep 2012.