DISSERTATION

submitted to the

Combined Faculty for the Natural Sciences and Mathematics

of

Heidelberg University, Germany

for the degree of
Doctor of Natural Sciences

put forward by
M.Sc. Robert Breckner né Dalitz
born in Solingen

Date of oral examination: ....

Compressed Motion Sensing and Dynamic Tomography

Advisor: Prof. Dr. Stefania Petra

Zusammenfassung

Compressed Sensing ist ein neues Abtast-Paradigma der mathematischen Signalverarbeitung, das unter bestimmten Annahmen eine Rückgewinnung des Signals aus stark unterabgetasteten Messungen ermöglicht. Die Erweiterung der mathematischen Theorie und die Analyse und Entwicklung neuer Anwendungen in vielen Bereichen sind Gegenstand zahlreicher internationaler Forschungsaktivitäten.

In dieser Arbeit wird ein industrielles Problem aus der experimentellen Fluiddynamik exemplarisch betrachtet. Nach aktuellem Stand der Technik wird das Problem in zwei unabhängigen Schritten gelöst: Erst werden Partikelbilder durch eine weniger verbreitete Art der Tomographie wiederhergestellt, woraufhin die Bewegung zwischen zwei vorgegebenen Zeitpunkten abgeschätzt wird. Dies motiviert das Problem der gleichzeitigen Signal- und Bewegungsschätzung und wirft theoretische Fragen im Bereich Compressed Sensing in Zusammenhang mit der Wiederherstellung von dünn besetzten und zeitlicher veränderlichen Signalen auf.

Insbesondere werden zwei verschiedene Ansätze zur Gewinnung eines sich über die Zeit verändernden Signals und dessen Bewegung aus unterabgetasteten, linearen Messungen zu zwei verschiedenen Zeitpunkten vorgestellt. Der erste Ansatz formuliert das vorliegendes Problem als optimalen Transport zwischen zwei indirekt beobachteten Dichteverteilungen mit physikalischen Einschränkungen. Es werden mehrere Methoden vorgeschlagen, um die Projektionsbeschränkungen in das konvexe Optimierungsframework von Benamou und Brenier zu integrieren.

Im zweiten Ansatz wird das Signal so modelliert, als ob es von einem realen Sensor erfasst wird, der durch den Versuchsaufbau festgelegt ist, und einem zusätzlichen, virtuellen Sensor, der durch die Bewegung entsteht. Die Kombination dieser beiden Sensoren wird Compressed Motion Sensor genannt, dessen Eigenschaften aus der Sichtweise von Compressed Sensing untersucht werden. Es wird gezeigt, dass in *Compressed Motion Sensing (CMS)* neben dem Grad der Dünnbesetztheit eine ausreichende Signaländerung zu Rekonstruktionsgarantien führt und dass der Compressed Motion Sensor die Performance des realen Sensors mindestens verdoppelt. Darüber hinaus kann bei bestimmten Dünnbesetztheitsgraden ebenfalls die Signalbewegung ermittelt werden.

Abstract

Compressed sensing is a new sampling paradigm of mathematical signal processing which, under certain assumptions, allows signal recovery from highly undersampled measurements. The extension of the mathematical theory and the analysis and development of new applications in many fields are the subject of numerous international research activities.

In this thesis an industrial problem from experimental fluid dynamics is consider, exemplarily. The current state of the art methodology solves the problem in two independent stages: First it recovers particle images by nonstandard tomography, and secondly it estimates the motion between two given time points. This motivates the problem of joint signal and motion estimation while raising theoretical questions in compressed sensing related to the recovery of sparse time-varying signals.

In particular, two different approaches are presented for recovering a time-varying signal and its motion from undersampled linear measurements taken at two different points in time. The first approach formulates a problem at hand as optimal transport between two indirectly observed densities with a physical constraint. Several methods are proposed to integrate the projection constraints into the convex optimization framework of Benamou and Brenier.

In the second approach, the signal is modeled as if observed by the real sensor specified by the experimental setup and an additional virtual sensor due to motion. The combination of these two sensors is called compressed motion sensor and its properties are examined from the viewpoint of compressed sensing. It is shown that in *compressed motion sensing (CMS)*, besides sparsity, a sufficient change of signal leads to recovery guarantees and it is demonstrated that the compressed motion sensor at least doubles the performance of the real sensor. Moreover, for certain sparsity levels the signal motion can be established, too.

Acknowledgments

Biggest thanks goes to Stefania Petra for her endurance, guidance and 24 hour support. The same counts for Christoph Schnörr who has given me the chance to start as a Ph. D. student in the Research Training Group (RTG 1653) “Spatio/Temporal Graphical Models and Applications in Image Analysis” and never gave up on me later on. Both are at the Faculty of Mathematics and Computer Science of Heidelberg University. It is hard to imagine better and fairer supervisors than them.

Also, thanks to my roommates Francesco, Mattia and Tobias. Our occasional on and off topic discussions made my time in the office very enjoyable. Especially, the walking encyclopedia Francesco spared me the one or other literature research. Further thanks go to my other and former colleagues and the entire group where I especially want to mention Florian who reviewed the thesis. In particular, I will miss the time with everybody which we spent every now and then on playing table soccer.

Moreover, spacial thanks goes to my family and friends for always backing me up and, especially, to my wife Tina for supporting me in every possible way.

I gratefully acknowledge the funding by the Deutsche Forschungsgemeinschaft via the Research Training Group (RTG 1653) “Spatio/Temporal Graphical Models and Applications in Image Analysis”. Without that, the current thesis would not have been possible.

Contents

1 Introduction

Estimating fluid motion by image sequence analysis is an active research field [Adr05, RWWK13] with a high industrial impact [LaV] due to its wide range of applications from calculating forces and moments on aircraft to combustion chamber design in engines.

A prevailing method for the quantitative investigation of fluids by imaging techniques is *Particle Image Velocimetry (PIV)*, that is since many years a very important and active research field in experimental fluid mechanics. New PIV methods that are applied in real-world scenarios e.g. wind-tunnels, complement numerical results from direct simulations of the Navier-Stokes equation and continuously lead to the understanding of the complex nature of turbulent flows.

A now established technique and a prominent example of PIV for imaging turbulent fluids with high speed cameras is a 3D technique called *Tomographic Particle Image Velocimetry (Tomo-PIV)* developed by Elsinga, Scarano and Wieneke [ESWvO06, Sca13, Wie08, Wie13]. The image measurement process proceeds as follows: First, the flow medium is seeded with small tracer particles that are designed such that they accurately follow the motion of the fluid. Advanced imaging devices (lasers, high-speed cameras, control logic etc.) illuminate and record fully time-resolved 2D image sequences of particle distributions at high resolutions. Next, 3D reconstructions of particle volume functions are obtained by tomographic inversion from few and simultaneous projections (2D images) of the tracer particles within the fluid. Finally, entire velocity fields are measured by taking two or more 3D particle volume functions within short time intervals, and by estimating and interpolating the displacements of individual particles from frame to frame.

To put it in a nutshell, Tomo-PIV consists of two computational steps:

- the *image recovery* or *reconstruction* problem of 3D particle volume functions from simultaneously recorded 2D images from few different angles corresponding to one time point.
- the actual *motion estimation* procedure of calculating the particle displacement based on at least two subsequent 3D particle volume functions that yields a 3D velocity field.

The essential new step of Tomo-PIV related to other PIV methods is the 3D particle reconstruction problem from 2D images. These 2D images can be interpreted as projections of the 3D particle distribution and the physical measurement process is closely related to the forward problem of optical tomography. This also explains the word “tomo” in Tomo-PIV. The reconstruction problem, can be formulated as an underdetermined system of linear equations of the form

$$Ax = b \qquad \text{with} \qquad A \in \mathbb{R}^{m\times n}, b \in \mathbb{R}^m \text{ and } m \ll n \; . \tag{1.0.1}$$

Tomo-PIV adopts a simple discretization scheme known as the *algebraic image reconstruction model (ART)* which assumes that the image consists of an array of unknowns (voxels), and sets up algebraic equations for the unknowns in terms of measured projection data. The latter are the pixel entries in the recorded 2D images. State of the art reconstruction methods for underdetermined systems above are *multiplicative ART (MART)* and *simultaneous MART (SMART)* [HK99, Her09, CZ97] which both solve a linearly constrained entropy-like problem

$$\min \sum_{i=1}^{n} x_i \log(x_i) \qquad \text{subject to} \qquad Ax = b \text{ and } x \geq 0 \; . \tag{1.0.2}$$

The negative of the objective function above $E(x) := -\sum_{i=1}^{n} x_i \log(x_i)$ is the Boltzmann-Shannon entropy measure. Entropy maximization is providing good results for sparse enough particle distributions. The reason is that for such sparse solutions the feasible set of problem (1.0.2) is a singleton [PS14]. As a consequence minimizing *any* other function would also make sense. Increasing the seeding density, however, will lead for the approach in (1.0.2) to dense solutions with strictly positive entries in the solution vector, except for those entries that can be fixed a priori to zero (conform section 2.5.4.2). But higher densities are desirable since they ease subsequent motion estimation.

Another disadvantage is that MART and SMART exhibit slow convergence after an initial phase of rapid progress towards the solution. As a consequence engineers interrupt the iteration process after few runs [ESWvO06]. This results in artifacts commonly called *ghost particles*. On the positive side, it is to be emphasized that both MART and SMART are row action methods [CZ97] and capable of dealing with very large systems as encountered in practice. In general, one has to deal with millions of unknowns n and thousands of measurements m resulting in matrices A that may be much too big to be stored in computer memory.

Commonly in Tomo-PIV, ART techniques are combined with *cross correlation* [RWWK13], a well-proven method for estimating the fluid motion from corresponding image pairs based on the correlation of local interrogation volumes in subsequent 3D reconstructions. This method heuristically splits the volume into small regions and estimates motion by searching for corresponding ones in subsequent steps in time. It also involves a local averaging process which is fast but empirically derived and error-prone. In cross correlation motion estimation is carried out regardless of spatial context. As a consequence, prior knowledge about spatial flow structures cannot be exploited during estimation, and missing motion estimates in image regions where a correlation analysis yields no reliable estimates, have to be heuristically inferred in a post-processing step. As a consequence engineers start tuning away from the classic cross correlation method and have started imposing physical priors on the particle motion [SGS16].

The main motivation for this work is the development of a well-founded approach that in contrast to solving the problem in two independent stages, i.e. image reconstruction and motion estimation, performs these two steps *jointly*. Compared to the current approach of Tomo-PIV, a joint approach has the potential to enable synergetic effects: image reconstruction benefits from simultaneously considering previous reconstructions and available correspondence information. Motion estimation on the other hand benefits from the improved reconstruction quality.

1.1 Related Work

The classic theory of compressed sensing (CS) [FR13] focuses on properties of underdetermined linear systems (1.0.1) that guarantee the accurate recovery of *sparse* solutions x from observed measurements b. Theoretical assertions are based on random ensembles of measurement matrices A and are in general not subject to design. Sensor matrices A, as they occur in Tomo-PIV, do not meet the theoretical conditions that compressed sensing relies on [PS09]. Nevertheless, the image reconstruction problem in Tomo-PIV is an instance of a compressed sensing problem. It can be guaranteed that uniformly sparse vectors x can be provably recovered with high probability from *deterministic* sensors that are based on real Tomo-PIV measurement setups [PSS13, PS14]. Motivated by the average case analysis of recovery conditions for such *static* Tomo-PIV sensors, this thesis considers the extension to more realistic *dynamic* scenarios and includes, in particular, the two-frame analysis of Tomo-PIV.

Related work in the framework of the L^2 *optimal transport (OT)* problem, also known as the L^2 Monge-Kantorovich problem [Vil08], is [BB00, AABC15, SKA15a, SKA15b]. In OT the goal is to find the most efficient way of redistributing an initial density to a target density such that the total L^2 distance is minimized. The Benamou-Brenier algorithm [BB00] is not only a popular improvement to the numerical resolution of the L^2 OT problem, but is also capable of incorporating physical constraints into OT by solving a space-time convex variational problem. The work in [SKA15a, SKA15b], with a focus on PIV methods, was published recently. The authors adopt a continuous PDE-based approach (iteratively linearized Monge-Ampère equation) that leads to a more economical problem parametrization and enables, in particular, to take into account additional physical fluid flow models. On the other hand, a performance analysis is only provided for simple 1D settings or a single particle in 2D, and additional rectifying filters are needed if the approach is not discretized on a sufficiently fine grid. In contrast to the problem considered in this thesis, the above works [BB00, SKA15a, SKA15b] do not consider distributions that are only *indirectly* observed by an undersampling operation. This however is the case studied in [AABC15]. The authors consider a variational approach based on continuous OT to object recovery with a *predefined shape* from multiple tomographic measurements. The numerical implementation of the approach seems to suffer from severe issues of numerical sensitivity and stability, and has only been applied to solid bodies of simple shapes.

The "discretize-then-optimize" strategy adopted in the second part of this thesis relates to *discrete optimal transport* [Vil08], that has already been used in image processing. In connection with color transfer between natural images, the authors of [FPPA14] study *regularized* discrete optimal transport that enforces spatially smooth displacements.

Beyond the field of mathematics, in experimental fluid dynamics, highly engineered approaches to joint particle recovery and motion estimation have starting emerging, that require parameter tuning but do not provide any recovery guarantees. This works include [NBS10, LS15, SGS16].

1.2 Contribution and Organization

In this thesis two fundamentally distinct approaches are developed, that combine the two independent steps in Tomo-PIV, image reconstruction and motion estimation, in a single step.

The first approach builds on *continuous optimal transport* between two particle distributions and incorporates into the Benamou and Brenier OT framework [BB00] the linear constraint (1.0.1) corresponding to one image pair. Three different optimization strategies are derived to handle the additional projection constraints. These are validated in experiments.

The second approach builds on *discrete optimal transport* and considers the joint problem of signal as well as signal correspondence recovery. Uniqueness of both recovery of binary sparse signals (particles) and signal correspondence (displacements between particles) is established under conditions that, besides sparsity, involve a sufficient change of signal transformation. The approach can be seen as an extension of the CS framework to a temporally changing signal which is computed in parallel to the signal reconstruction. Thus, it extends the static perspective of the CS framework in [PS14] to realistic dynamic situations and leads to compressed motion sensing (CMS). CMS is the main contribution of this thesis. Computationally the joint problem of reconstruction and transformation estimation can be addressed by a large-scale linear program. Numerical experiments validate theoretical results and illustrate the performance of CMS.

Chapter 2 introduces basic material being used in later chapters. Besides compressed sensing and different approaches for estimating motion which build the basis of subsequent methods, results from convex optimization together with useful algorithms, properties of permutations and separate smaller tools are recalled.

The aforementioned two distinct approaches combining image recovery and motion follow in separate chapters. The methods presented in chapter 3 are based on Benamou and Brenier's OT framework for motion estimation. These differ in the specific representation of the target function and algorithms used for solving. Chapter 4 includes the main contribution, namely compressed motion sensing, a theoretical extension of compressed sensing, and investigations related to make computation practical via linear programming relaxation. Finally, a conclusion is given in chapter 5.

2 Preliminaries

2.1 Linear Subspace Angles

The dimension of the intersection of two linear subspaces of the same vector space can vary between just one and the dimension of the vector space itself. Sometimes it is useful to further determine how distant two subspaces are by computing specific angles between them. The derivation starts with the following definition.

Definition 2.1 *(Stiefel manifold)*:
The *Stiefel manifold*

$$\mathcal{V}_{m,n} := \{X \in \mathbb{R}^{n \times m} \mid X^\top X = I_m\}$$

is the set of real orthogonal $n \times m$ matrices with normalized columns.

Naturally, the columns of each element of $\mathcal{V}_{m,n}$ are basis vectors of a linear subspace of $\mathbb{R}^n$. Thus, the mapping of each element of $\mathcal{V}_{m,n}$ to the set of m-dimensional subspaces is merely surjective since the elements columns have a fixed order. That subspace is the Grassmann space.

Definition 2.2 *(Grassmann space)*:
The *Grassmann space*

$$\mathcal{G}_{m,n} := \{X \subseteq \mathbb{R}^n \mid X \text{ is a linear space, } \dim(X) = m\}$$

is the manifold of m-dimensional linear subspaces of $\mathbb{R}^n$.

The aforementioned concept for measuring the distance between two elements of the Grassmann space are the principal angles.

Definition 2.3 *(principal angles)*:
Let $\mathcal{X}, \mathcal{Y} \in \mathcal{G}_{m,n}$ and let the columns of $X, Y \in \mathcal{V}_{m,n}$ form an orthogonal basis for $\mathcal{X}$ and $\mathcal{Y}$, respectively. The *i-th principal angles θ_i* between $\mathcal{X}$ and $\mathcal{Y}$ is defined as

$$\theta_i(\mathcal{X}, \mathcal{Y}) = \arccos\left(\sigma_i\left(X^\top Y\right)\right) \in [0, \tfrac{\pi}{2}]$$

where σ_i is the i-th largest singular value (see definition 2.6), and $\theta(\mathcal{X}, \mathcal{Y}) := \theta_1(\mathcal{X}, \mathcal{Y})$.

Definition 2.3 implies an order of the principal angles being

$$\theta(\mathcal{X}, \mathcal{Y}) = \theta_1(\mathcal{X}, \mathcal{Y}) \leq \ldots \leq \theta_m(\mathcal{X}, \mathcal{Y})$$

where the smallest principal angle θ is of specific importance. It determines whether the two subspaces have only zero in common as the following lemma shows.

Lemma 2.4 *([Dix49, Deu95])*:
Let $\mathcal{X},\mathcal{Y} \in \mathcal{G}_{m,n}$. Then

$$\begin{aligned} &\theta(\mathcal{X},\mathcal{Y}) > 0 \qquad &&\Leftrightarrow \qquad \mathcal{X} \cap \mathcal{Y} = \{0\} \\ \text{and} \qquad &\theta(\mathcal{X},\mathcal{Y}) = \frac{\pi}{2} \qquad &&\Leftrightarrow \qquad \mathcal{X} \perp \mathcal{Y} \ . \end{aligned}$$

In particular this implies

Corollary 2.5:
Let $\mathcal{X},\mathcal{Y} \in \mathcal{G}_{m,n}$. Then

$$\theta(\mathcal{X},\mathcal{Y}) = 0 \qquad \Leftrightarrow \qquad \dim(\mathcal{X} \cap \mathcal{Y}) \geq 1 \ .$$

2.2 Matrix Decompositions

There are many matrix decompositions which are usually used to implement efficient algorithms or to help investigating a problem analytically. This section introduces the decompositions of real matrices that are used in the following chapters.

Definition 2.6 *(Singular Value Decomposition (SVD), [TBI97, Section I.4])*:
Let $M \in \mathbb{R}^{n\times m}$. Then there are orthogonal matrices $U \in \mathbb{R}^{n\times n}$, $V \in \mathbb{R}^{m\times m}$ and a diagonal matrix $\Sigma \in \mathbb{R}^{n\times m}$ with

$$\operatorname{diag}(\Sigma) = (\sigma_1, \ldots, \sigma_k)\,, \quad \sigma_1 \geq \cdots \geq \sigma_k \geq 0, \quad k = \min(m,n)$$

such that

$$M = U\Sigma V^\top \ .$$

$\sigma_1, \ldots, \sigma_k$ are the *singular values* of M and the columns of U and V are called the *left* and *right singular vectors*, respectively. The singular values are uniquely determined whereas the singular vectors can only be uniquely determined up to a factor of ±1, if the singular values are distinct and M is a square matrix. [TBI97, Theorem 4.1].

Definition 2.7 *(Polar Decomposition, [Chi12, Section 1.5.1])*:
Let $M \in \mathbb{R}^{n\times m}$, $n \geq m$, with $\operatorname{rank}(M) = m$. Then the unique polar decomposition of M is given by

$$M = H_M T_M^{1/2} \quad \text{with} \quad H_M = M(M^\top M)^{-1/2} \in \mathcal{V}_{m,n}, \quad T_M = M^\top M, \tag{2.2.1}$$

where $\mathcal{V}_{m,n}$ is the *Stiefel manifold* (see definition 2.1).

The columns of H_M form an orthonormal basis for the range $\mathcal{R}(M) = \{Mx \mid x \in \mathbb{R}^m\}$ of M.

2.3 Permutations

The rearrangement of n elements can be seen as a bijective function mapping from the finite set X with n elements to itself and is referred to as *permutation.* No matter how often and how heavy the elements of X are permuted, the result can always be described by a single permutation function $p\colon [n] \to [n]$ which uses the set $[n] := \{1, \dots, n\}$ representatively for an arbitrary set with n elements. This makes the set of all possible permutations a group.

Definition 2.8 *(symmetric group)*:
The group consisting of the set

$$\mathcal{S}_n := \{p\colon [n] \to [n] \mid p \text{ is bijective}\}$$

with $n \in \mathbb{N}$ and the function composition as group operation is called *symmetric group.*

Each of the $|\mathcal{S}_n| = n!$ different elements of $\mathcal{S}_n$ is a *permutation.* Matrices implementing this kind of permutations on the positions of vector entries by a matrix-vector multiplication are referred to as *permutation matrices.*

Definition 2.9 *(permutation matrix)*:
Each element of the set

$$\mathcal{P}_n := \left\{P \in \{0,1\}^{n \times n} \mid (P_{ij} = 1 \Leftrightarrow p(i) = j)\,,\ p \in \mathcal{S}_n\right\}$$

forming a group together with the matrix multiplication is a *permutation matrix* of size n.

It can be deduced from the definition that each permutation matrix $P \in \mathcal{P}_n$ corresponds to exactly one bijective function $p \in \mathcal{S}_n$ and vice versa. The inverse permutation matrix P^{-1} changing vector entries of some $x \in \mathbb{R}^n$ back to the position before calculating Px is naturally

$$P^{-1} = P^\top \qquad \text{leading to} \qquad P^\top P = PP^\top = I$$

and multiplying $x \in \mathbb{R}^n$ to P from the right and left gives

$$(Px)_i = x_{p(i)} \qquad \text{and} \qquad (x^\top P)_i = \left((P^\top x)^\top\right)_i = x_{p^{-1}(i)}\ .$$

Every permutation p decomposes into k disjoint *cycles.* Those can be separated into f *fixed points* and c cycles of length at least 2, hence $k = c + f$. The notation of cycles C_j having lengths $l_j \geq 2$ for $j \in [c]$ and $l_j = 1$ for $j \in \{c+1, \dots, k\}$ so that

$$p = \Big\{\underbrace{(i_{1,1} \cdots i_{1,l_1})}_{=:C_1} \cdots \underbrace{(i_{c,1} \cdots i_{c,l_c})}_{=:C_c}\underbrace{(i_{c+1,l_{c+1}})}_{=:C_{c+1}} \cdots \underbrace{(i_{k,l_k})}_{=:C_k}\Big\} \tag{2.3.1}$$

in the sense of cyclic permutations

$$p(i_{j,t}) = i_{j,t+1} \quad \text{with} \quad i_{j,l_j+1} := i_{j,1} \quad \text{for} \quad t \in [l_j]$$

is called *cycle notation* of p. It is common practice not to list the cycles of length $l_j = 1$ for $j \in \{c+1, \dots, k\}$ since those are said fixed points of the permutation function p, i.e. $p(i_j) = i_j$, and do not change their position. A given permutation p partitions the set $[n]$ into

$$\mathcal{I}_p := \bigcup_{j \in [c]} \{i \mid i \in C_j\}$$

which is the set of elements belonging to a cycle of length 2 or greater and its complement

$$\mathcal{F}_p := [n] \setminus \mathcal{I}_p = \{i \in [n] \mid p(i) = i\}$$

being the set of fixed points. Consequently,

$$[n] = \mathcal{I}_p \mathbin{\dot\cup} \mathcal{F}_p \quad \text{and} \quad n = r + f$$

with cardinalities

$$r := |\mathcal{I}_p| \quad \text{and} \quad f := |\mathcal{F}_p| \ .$$

The following definition and the subsequent lemma deal with the number of permutations without fixed points known as *derangements*.

Definition 2.10 *(derangement)*:
A *derangement* is a permutation $p \in \mathcal{S}_n$ with cycle structure (2.3.1) without a fixed points, i.e. $f = 0$ so that $k = c$. It is said that the elements are *deranged*.

Lemma 2.11 *(counting derangements [Has03])*:
The number of derangements in $\mathcal{S}_n$ is equal to the *subfactorial*

$$!n := n! \sum_{i=0}^{n} \frac{(-1)^i}{i!} = \left\lfloor \frac{n!}{e} \right\rceil \ ,$$

where $\lfloor \cdot \rceil$ is the rounding towards the nearest integer number.

Probabilistic results for the number of cycles of a random permutation are already known for a long time.

Lemma 2.12 *(expected value and variance of number of cycles [SL66, Gon42])*:
Let $p \in \mathcal{S}_n$ be a random permutation drawn uniformly from $\mathcal{S}_n$. The expected value of the number of cycles k of p and its corresponding variance are

$$\mathbb{E}_n[k] = \sum_{i \in [n]} \frac{1}{i} = \log(n) + \mathcal{O}(1)$$

$$\text{and} \qquad \mathbb{V}_n[k] = \sum_{i \in [n]} \frac{i-1}{i^2} = \log(n) + \mathcal{O}(1) \ .$$

Permutations and permutation matrices can further be partitioned into conjugation classes. Each class consists of the same cycle structure and has the same spectrum and similar eigenvectors. These results can be found in the work of Stuart and Weaver [SW91] and is summarized in the following lemma and its corollaries.

Lemma 2.13 *(conjugate classes of permutations [SW91, section 3])*:
Let $Q \in \mathcal{P}_n$ be a permutation matrix whose corresponding permutation decomposes into disjoint cycles as in (2.3.1). This is conjugate to the block permutation matrix $P \in \mathcal{P}_n$ with

$$P = \begin{bmatrix} P_{l_1} & & \\ & \ddots & \\ & & P_{l_k} \end{bmatrix}$$

having the cycle structure

$$p = \Big\{(1 \cdots l_1)\ ((l_1+1) \cdots (l_1+l_2)) \cdots ((n-l_k+1) \cdots (n))\Big\}$$

where P_j is the circulant permutation matrix

$$P_j := \begin{bmatrix} 0 & I_{l_j-1} \\ 1 & 0 \end{bmatrix} \in \mathcal{P}_{l_j}\ . \tag{2.3.3}$$

A third permutation $R \in \mathcal{P}_n$ exists such that $Q = RPR^\top$.

Corollary 2.14 *(spectrum of permutation matrices [SW91, section 3])*:
Let $P \in \mathcal{P}_n$ be a permutation matrix with cycle structure (2.3.1). The spectrum of P is

$$\operatorname{spec}(P) = \bigcup_{j=1}^{k} \left\{ \varphi_{l_j}^{i} \;\middle|\; i \in [l_j] \right\}\ , \tag{2.3.4}$$

where $\varphi_{l_j} \in \mathbb{C}$ is a primitive l_j-th root of unity, that is $\varphi_{l_j}^{l} \neq 1$ for $l \in [l_j - 1]$ and $\varphi_{l_j}^{l_j} = 1$.

Corollary 2.15 *(eigenvectors of permutation matrices [SW91, section 3])*:
Let $P \in \mathcal{P}_n$ be a permutation matrix with cycle structure (2.3.1). Then P has n linear independent complex eigenvectors. For each $j \in [n]$, the eigenvector entries at indexes C_j can be obtained from the eigenvectors of the submatrix P_j in (2.3.3). The remaining entries at indexes $[n] \setminus C_j$ must be set to 0.

2.4 Convex Optimization

The aim of mathematical optimization is the search for a global minimum in the domain $C \subseteq \mathbb{R}^n$ of a function $f : C \to \mathbb{R}$, regardless of whether it is achievable or not. That means an optimal value x^* is desired, so that there is no other value x which yields a smaller function value, i.e. $f(x^*) \leq f(x)$ for all $x \in C$. Often there are feasibility constraints involved acting on the functions variable x which have the form of equalities and inequalities, e.g. $g(x) \geq h(x)$, or equivalently $h(x) - g(x) \leq 0$.

Every (in-)equality can be rewritten in a similar way, so that the general non-linear optimization problem can be formulated as

Problem 2.16 *(general nonlinear optimization problem)*:

$$\underset{x \in \mathcal{S}}{\text{minimize}}\, f(x)$$

where

$$\mathcal{S} := \left\{ x \in C \;\middle|\; \begin{array}{ll} f_i(x) \leq 0 & \text{for } i = 1, \ldots, p \\ f_j(x) = 0 & \text{for } j = p+1, \ldots, m \end{array} \right\} .$$

is the *feasible set.*

In this generality, a realistic goal is to find a local minimum of $f(x)$ using existing methods. Many algorithms are iterative, start at any location x and then improve the value $f(x)$ step by step with the help of derivatives, such as the *gradient descent algorithm.* In every step it computes the gradient, takes a step in the opposite direction, and repeats until either the improvement of the function value or the gradient becomes too small, which are sufficient criteria in many practical applications. If that is the case a local optimum is assumed to be found. There are lots of other methods which extend gradient descent e.g. by more sophisticated choices for computing step direction and length. More detailed information about algorithms can be found in section 2.4.3.

In this respect a *convex* setting is a big advantage. Convexity is a property of a function or a set, which is beneficial for finding a global minimum. The global optimization of a convex function reduces to the search for one minimum, since every minimum of a convex function is a global minimum. Thus, convexity is a desirable property.

Definition 2.17 *(convex set)*:
Let S be a subset of a vector space. S is called *convex* if

$$tx_1 + (1-t)x_2 \in S \qquad \text{for all} \qquad t \in [0,1]$$

and any two points $x_1, x_2 \in S$.

In other words a set S is convex if the entire connecting line between x_1 and x_2 lies in S and does not cross the boundaries of the set. Moreover, a function $f : C \to \mathbb{R}$, $C \subseteq \mathbb{R}^n$ is *convex*, if (and only if) its so called *epigraph* is a convex set.

Definition 2.18 *(epigraph)*:
The *epigraph* of a function $f : \mathbb{R}^n \to \overline{\mathbb{R}} := [-\infty, \infty]$ is the set

$$\operatorname{epi} f := \{(x, y) \in \mathbb{R}^n \times \mathbb{R} \mid f(x) \leq y\} \in \mathbb{R}^{n+1} .$$

The epigraph contains all points on or above the graph of a function illustrated in figure 2.1. Thus, if there is more than one minimum of a convex function, all of them have the same function value and the set of all minima is also a convex set.

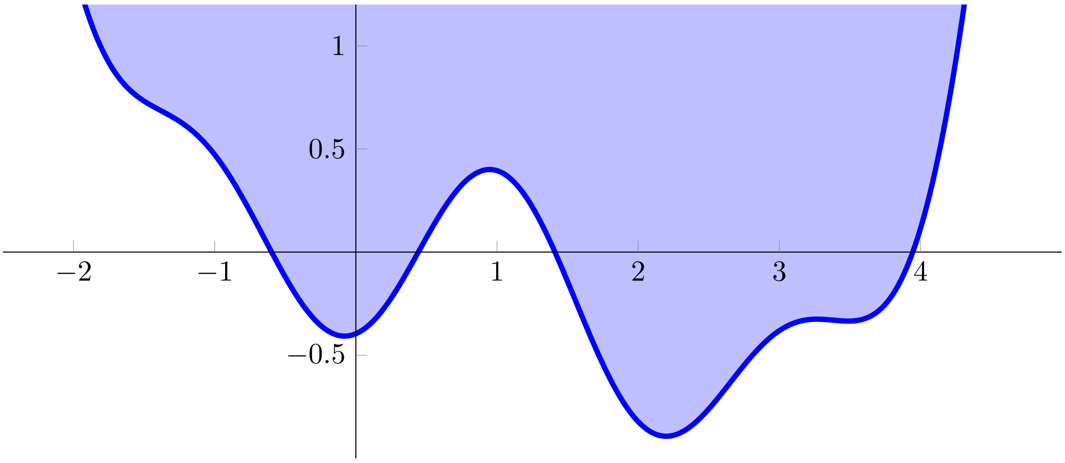


Figure 2.1 **-** The epigraph of a function is the set of all points on or above the functions graph.

The properties which make a general optimization problem 2.16 a convex optimization problem are the following.

Assumption 2.19 *(convexity assumptions for problem 2.16, [JS03, Section 8.1])*:

1. The domain $C \subseteq \mathbb{R}^n$ of f is convex and $C \subset \text{dom} f_i$ for $i = 1, \dots, m$
2. Functions f and $f_i : \mathbb{R}^n \to \mathbb{R} \cup \{\infty\}$ are convex for $i = 1, \dots, p$
3. Function f_j are affine for $j = p+1, \dots, m$

The last property means that the constraint functions f_j are nothing else than linear equations $a_j x = b_j$. Compared to a general optimization setting the additional property assumptions 2.19 allow the application of *duality theory* (see section 2.4.1) on the one hand, which leads to reliable and more sophisticated methods in order to find a global minimum of the target function f. On the other hand, existing and especially gradient based methods, like the aforementioned gradient descent algorithm, converge to a global minimum instead of possibly getting stuck in a non-optimal local one.

An important concept for dealing with convex optimization problems of the form 2.16 together with the convexity assumptions 2.19 is the *Lagrangian function.*

Definition 2.20 *(Lagrangian function [JS03, Section 8.3])*:
Let $D := \{y \in \mathbb{R}^m \mid y_i \geq 0 \text{ for } i \in [p],\ y_j \in \mathbb{R} \text{ for } j \in \{p+1, \dots, m\}\}$. Then the function $L : C \times D \to \mathbb{R}$ defined by

$$L(x, y) := f(x) + \sum_{i=1}^{m} y_i f_i(x)$$

is called *Lagrange function* of the problem 2.16. Every y_i is called *Lagrange multiplier.*

This transforms the constrained optimization problem into a so called *saddle-point problem.* In the context of the Lagrangian function a *saddle-point* is defined in the following way.

Definition 2.21 *(saddle-point [JS03, Section 8.3])*:
The point $(\bar{x}, \bar{y}) \in C \times D$ is a *saddle-point* of L on $C \times D$, if

$$L(x, \bar{y}) \leq L(\bar{x}, \bar{y}) \leq L(\bar{x}, y)$$

for all $x \in C$ and $y \in D$.

The aim of the corresponding saddle-point problem is to find $\bar{x}$ and $\bar{y}$ satisfying these inequalities which amounts in the search for the optimum of

$$\max_{y \in D} \min_{x \in C} L(x, y) \ . \tag{2.4.1}$$

The precise formulation (theorem 2.23) of the relationship between the target function $f(x)$, the Lagrangian $L(x, y)$ and the saddle-point problem goes back to Karush, Kuhn and Tucker (KKT). An important role plays the condition by Slater.

Definition 2.22 *(Slater's constraint qualification [JS03, Section 8.1])*:
The condition

$$\exists\, x \in \operatorname{int}(C) \cap \mathcal{S} : f_i(x) < 0 \text{ for all non-affine } f_i \text{ with } i \leq p$$

is called *Slater's constraint qualification* or in short *Slater condition*, where $\operatorname{int}(C)$ is the interior of C.

Together with the so called *KKT conditions* representing conditions for a point being a minimum of the target function of problem 2.16 with respect to the assumptions 2.19, they form the basis for Karush, Kuhn and Tucker's statements. The following theorem describes the relations.

Theorem 2.23 *(Karush, Kuhn & Tucker [JS03, Section 8.3])*:
Let the convexity assumptions 2.19 be fulfilled for problem 2.16. Then:

1. If $(\bar{x}, \bar{y})$ is a saddle-point of L on $C \times D$, then $\bar{x}$ is an optimal solution for the problem 2.16 and $\bar{y}_i f_i(\bar{x}) = 0$ for $i = 1, \ldots, m$, i.e.

   $$L(\bar{x}, \bar{y}) = f(\bar{x})$$

2. If $\bar{x}$ is an optimal solution of problem 2.16 and the Slater condition is fulfilled, then there exists $\bar{y} \in D$, so that $(\bar{x}, \bar{y})$ is a saddle-point of L.

3. If the optimal value α of 2.16 is finite,

   $$\alpha = \inf\{f(x) \mid x \in \mathcal{S}\} \in \mathbb{R} \ ,$$

   and the Slater condition is fulfilled, then there exists $\bar{y} \in D$, so that

   $$\alpha = \inf_{x \in C} L(x, \bar{y}) = \max_{y \in D} \inf_{x \in C} L(x, y) \ .$$

This means solving the original problem 2.16 is equivalent to finding a saddle-point for the Lagrangian function. For differentiable functions, the properties of such a saddle-point are summarized in the following definition.

Definition 2.24 *(KKT point)*:
Let $f(x)$ and $f_i(x)$ for $i \in [m]$ be differentiable. The point $(\bar{x}, \bar{y}) \in C \times D$ fulfilling the conditions

1. $f_i(\bar{x}) \leq 0,\ \bar{y}_i \geq 0 \quad \text{and} \quad f_i(\bar{x}) \cdot \bar{y}_i = 0 \quad \text{for} \quad i \in [p]$,
2. $f_j(\bar{x}) = 0 \quad \text{for} \quad j = p+1, \ldots, m \quad \text{and}$
3. $D_x L(\bar{x}, \bar{y}) = 0$

is called *Karush-Kuhn-Tucker point* or in short *KKT point* of problem 2.16.

Sophisticated algorithms often use the derivative of $f(x)$ or $L(x, y)$ for finding the desired minimum or saddle point, respectively. If $\bar{x}$ is a minimum of an unconstrained function, its derivative is naturally zero at this point. In general, this is not true for a constrained function $f(x)$ and a KKT point $(\bar{x}, \bar{y})$ of the corresponding Lagrangian in $\bar{y}$ direction, but is desired for algorithm design. The *augmented Lagrangian* is an extension to the Lagrangian function having this additional numerical advantage whereas the underlying optimization problem remains the same.

Definition 2.25 *(augmented Lagrangian [JS03, Section 11.2])*:
The function

$$L_\rho(x, y) := f(x) + \sum_{i=1}^{p} \frac{\rho_i}{2} \left(\left(f_i(x) + \frac{y_i}{\rho_i} \right)^+ \right)^2 + \sum_{j=p+1}^{m} \frac{\rho_j}{2} \left(f_j(x) + \frac{y_j}{\rho_j} \right)^2 - \frac{1}{2} \sum_{k=1}^{m} \frac{y_k^2}{\rho_k}$$

with fixed *penalty parameters* $\rho = (\rho_1, \ldots, \rho_m) > 0$ and $(\cdot)^+ := \max\{0, \cdot\}$, is called *augmented Lagrangian.*

Theorem 2.26 *(derivative of L_ρ at KKT point [JS03, Section 11.2])*:
Let $(\bar{x}, \bar{y}) \in C \times D$ be a KKT point of problem 2.16. Then

$$D_x L_\rho(\bar{x}, \bar{y}) = 0 \qquad \text{and} \qquad D_y L_\rho(\bar{x}, \bar{y}) = 0$$

for all $\rho = (\rho_1, \ldots, \rho_m) > 0$. Vice versa, if $DL_\rho(\bar{x}, \bar{y}) = 0$ for one $\rho > 0$, then $(\bar{x}, \bar{y})$ is a KKT point of problem 2.16.

It turns out that a point $(\bar{x}, \bar{y})$ with the properties of theorem 2.26 is not necessarily a saddle-point of the augmented Lagrangian. However, it can be shown that this is the case under further assumptions, which can be found together with more details in [JS03, Section 11.2].

At first sight the adjective “augmented” is not really justified, but it becomes understandable when looking at the case with no inequality constraints. When $p = 0$, then the augmented Lagrangian reduces to

$$L_\rho(x, y) = L(x, y) + \sum_{j=1}^{m} \frac{\rho_j}{2} f_j^2(x) \ ,$$

that is the standard Lagrangian function *augmented* by an additional term. In practise all penalty parameters are often set to one single value $\rho > 0$ so that

$$L_\rho(x, y) = L(x, y) + \frac{\rho}{2} \sum_{j=1}^{m} f_j^2(x) \tag{2.4.2}$$

is the form which is most often used, in cases when there are just equality constraints. This is the form which is used in the following chapters, as well.

2.4.1 Fenchel Duality

The concept of *duality* allows to investigate many mathematical objects such as functions from another points of view. This new viewpoint has the form of another object, called the *dual* object, which properties are often much easier to handle. Especially in optimization, looking at the dual function of a problem's target function, that is the dual problem, can make the search for a minimum much easier. Roughly speaking, the applied transformations "convexify" the original problem yielding the dual one.

The basic operation to define dual functions is the *Legendre-Fenchel transform*, also called *(Fenchel) conjugation*. It is a natural generalization of the *Legendre transform* which is constrained to so-called *Legendre functions*.

Definition 2.27 *(Fenchel conjugate [Roc97, §12])*:
Let $f : \mathbb{R}^n \to \overline{\mathbb{R}}$ be an arbitrary function. The *Fenchel conjugate*, or in short just *conjugate*, $f^* : \mathbb{R}^n \to \overline{\mathbb{R}}$ is

$$f^*(y) := \sup_{x \in \mathbb{R}^n} \left\{ \langle y, x \rangle - f(x) \right\} \ .$$

If $f(x)$ is *proper*, i.e. $f(x) > -\infty$ for all x and there exists at least one x with $f(x) < \infty$, then the conjugation yields a convenient function. Otherwise, the conjugate is a constant function $f^*(y) = -\infty$.

The Fenchel conjugate can be interpreted as a representation of a function in terms of supporting hyperplanes. Every hyperplane can be expressed as an affine function $\langle y, x \rangle - a$ with a vector $y \in \mathbb{R}^n$ of directional slopes and $a \in \mathbb{R}$ representing the offset from the origin. The supporting hyperplanes are those where a takes the smallest possible value depending on y, so that the hyperplanes do not intersect the graph of $f(x)$. Formally this means

$$\begin{aligned} & f(x) \geq \langle y, x \rangle - a \qquad \text{for all} \quad x \\ \Leftrightarrow \qquad & a \geq \langle y, x \rangle - f(x) \qquad \text{for all} \quad x \\ \Leftrightarrow \qquad & a \geq \sup_x \left\{ \langle y, x \rangle - f(x) \right\} \ . \end{aligned}$$

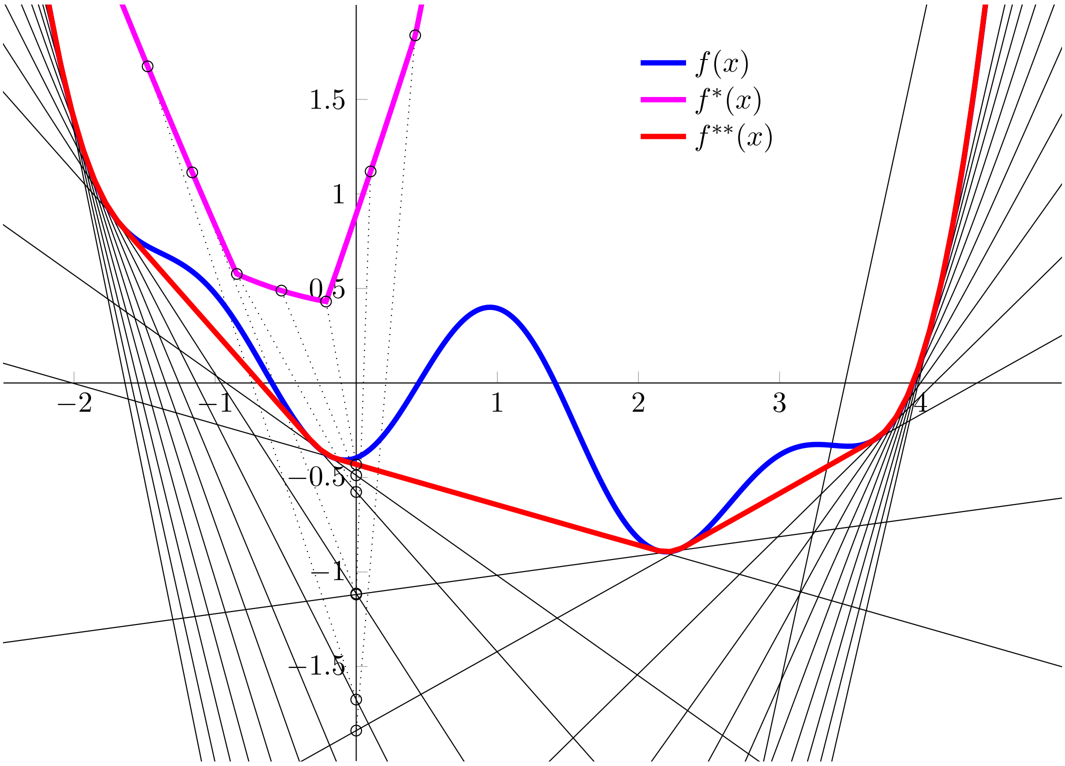


Figure 2.2 **-** A 1-dimensional function $f(x)$ and its conjugate $f^*(x)$ and biconjugate f^{**} together with a few supporting affine functions, i.e. lines. The dotted lines illustrate the connection between the hyperplanes and the conjugate $f^*(x)$.

Thus, the best choice for the smallest possible offset is $a = \sup_x \{\langle y, x\rangle - f(x)\}$ which defines the conjugate depending on given directional slopes y.

The conjugate $f^*(x)$ is always convex even for non-convex functions $f(x)$. Exciting facts reveal when considering the *biconjugate* of a function f, denoted by $f^{**} := (f^*)^*$. Of course it is a convex function, too, but more interestingly it is the greatest convex function which is majorized by $f(x)$ so that $f^{**}(x) \leq f(x)$ for all x. Thus, the unique minimum of the biconjugate of a function is also a global minimum of the function itself. Equality holds if $f(x)$ is already convex. Figure 2.2 illustrates the connection between a function and its conjugates in one dimension. The function shown is differentiable resulting in one possible supporting hyperplane for every $x \in \mathbb{R}$ with $f^{**}(x) = f(x)$ in the direction of the gradient. It might happen that there are more directions at an x_0 if f is not differentiable. In this case all possible directions of supporting hyperplanes are so called *subgradients* which are collected in the *subdifferential* at x_0.

Definition 2.28 *(subdifferential, subgradient [Roc97, §23])*:
Let $f : \mathbb{R}^n \to \overline{\mathbb{R}}$ be a function and $f(x_0) \in \mathbb{R}$ for a point $x_0 \in \mathbb{R}^n$. The set

$$\partial f(x_0) := \left\{ y \in \mathbb{R}^n \;\middle|\; f(x) - f(x_0) \geq y^\top (x - x_0) \text{ for all } x \in \mathbb{R}^n \right\}$$

is called the *subdifferential* of f at x_0. Each element is called a *subgradient* of f at x_0.

If f is convex and differentiable at x_0 then the only element of $\partial f(x_0)$ is the gradient $\nabla f(x_0)$. Closely related is the *normal cone*

$$N_C(x_0) := \left\{ y \in \mathbb{R}^n \,\middle|\, y^\top (x - x_0) \leq 0 \text{ for all } x \in C \right\}$$

containing $\{0\}$ and all normals of hyperplanes supporting the convex set $C \subset \mathbb{R}^n$ at $x_0 \in C$. It holds that

$$\partial \delta_C(x_0) = N_C(x_0) \tag{2.4.3}$$

where $\delta_{\mathcal{X}}$ is the indicator function

$$\delta_{\mathcal{X}}(x) := \begin{cases} 0 & \text{if } x \in \mathcal{X} \\ \infty & \text{if } x \notin \mathcal{X} \end{cases} \tag{2.4.4}$$

for a set $\mathcal{X}$. Further details on conjugate functions and subdifferentials can be found in [Roc97] and [BV04].

2.4.2 Lagrangian Duality

In order to formulate the dual problem a common way is to *perturb* the original problem and then consider the corresponding *inf-projection*. For a function $f : \mathbb{R}^n \to \overline{\mathbb{R}}$, the *perturbation function* is a function $\varphi(x, u) : \mathbb{R}^n \times \mathbb{R}^m \to \overline{\mathbb{R}}$ where $\varphi(x, 0) = f(x)$. If the infimum of $f(x)$ is of current interest, the *inf-projection* of φ defined by

$$\nu(u) := \inf_{x \in \mathbb{R}^n} \varphi(x, u)$$

can be evaluated at $u = 0$ instead. Given the general optimization problem 2.16, a reformulation provides great insights into the connection to the saddle-point problem 2.21. In the case of the search for a minimum in the whole $\mathbb{R}^n$, problem 2.16 can be written with the help of a perturbed function as

$$\min_x g(x, 0) \quad \text{where} \quad g(x, u) := \begin{cases} f(x) & \text{if} \quad f_i(x) + u_i \leq 0 \; \forall \, i, \;\; f_j(x) + u_j = 0 \; \forall \, j \\ \infty & \text{otherwise.} \end{cases}$$

By considering the inf-projection $\nu(u) := \inf_x g(x, u)$, the conjugate can be derived as

$$\begin{aligned}
\nu^*(y) &= \sup_u \Big\{ \langle y, u\rangle - \inf_x g(x,y) \Big\} \\
&= \sup_u \Big\{ \langle y, u\rangle - \inf_x \{ f(x) \mid f_i(x) + u_i \leq 0,\ f_j(x) + u_j = 0 \} \Big\} \\
&= \sup_u \Bigg\{ \langle y, u\rangle + \sup_x \{ -f(x) \mid f_i(x) + u_i \leq 0,\ f_j(x) + u_j = 0 \} \Bigg\} \\
&= \sup_{x,u} \{ \langle y, u\rangle - f(x) \mid f_i(x) + u_i \leq 0,\ f_j(x) + u_j = 0 \} \\
&= \sup_{x,v} \{ \langle y, v - f(x)\rangle - f(x) \mid v_i \leq 0,\ v_j = 0 \} \qquad (v \text{ substituting } f(x) + u) \\
&= \sup_{x,v} \{ \langle y_i, -v_i - f_i(x)\rangle + \langle y_j, -f_j(x)\rangle - f(x) \mid v_i \geq 0 \} \\
&= \sup_{x,v} \{ - \langle y_i, v_i\rangle - \langle y_i, f_i(x)\rangle - \langle y_j, f_j(x)\rangle - f(x) \mid v_i \geq 0 \} \\
&= -\inf_{x,v} \{ f(x) + \langle y_i, f_i(x)\rangle + \langle y_j, f_j(x)\rangle + \langle y_i, v_i\rangle \mid v_i \geq 0 \}\ .
\end{aligned}$$

This infimum does not exist if there is at least one i with $y_i < 0$, since the last term $\langle y_i, v_i\rangle$ could get arbitrary small. Thus, for $y_i \geq 0$, the best choice is trivially $v_i = 0$ for all i and the derivations continues

$$\begin{aligned}
\nu^*(y) &= \begin{cases} -\inf_x \{ f(x) + \langle y_i, f_i(x)\rangle + \langle y_j, f_j(x)\rangle \} & \text{if} \quad y_i \geq 0 \\ \infty & \text{otherwise} \end{cases} \\
&= \begin{cases} -\inf_x L(x,y) & \text{if} \quad y_i \geq 0 \\ \infty & \text{otherwise} \end{cases}
\end{aligned}$$

where L is the Lagrangian function as defined in definition 2.20. This leads to

Definition 2.29 *(Lagrangian dual function [Roc97, Section 36])*:
The function $\bar{L} : \mathbb{R}^m \to \mathbb{R}$ defined by

$$\bar{L}(y) := \inf_x L(x,y)$$

is called the *Lagrange dual function* of the problem 2.16.

This is the reason why the Lagrangian multipliers are often referred to as *dual variables*, as well. As every conjugate $\nu^*(y)$ is convex, every Lagrangian dual function is always concave where the sign is omitted for convenience. This is essential for algorithms solving constrained optimization problems (section 2.4.3).

As a prerequisite for the existence of a finite biconjugate

$$\nu^{**}(z) = \sup_y \{ \langle z, y\rangle - \nu^*(y) \} = \sup_y \Big\{ \langle z, y\rangle + \inf_x L(x,y) \ \Big|\ y_i \geq 0 \Big\}$$

of $\nu^*(y)$, it is necessary that a finite conjugate $\nu^*(y)$ exists such that

$$\nu^{**}(0) = \sup_{y} \inf_{x} \left\{ L(x,y) \mid y_i \geq 0 \right\} \ .$$

This is exactly the saddle point problem (2.4.1) and is essentially the maximization of the Lagrangian dual function, i.e.

$$\sup_{y} \left\{ \bar{L}(y) \,\middle|\, y_i \geq 0 \right\} \ .$$

This problem is known as the *dual problem* of the original problem, or in this context *primal problem* 2.16. Summing up, the saddle point problem and the dual problem are nothing but the biconjuguate of the perturbed function of the primal problem evaluated at 0.

2.4.3 Algorithms

The previous sections deal with the formulation of constraint optimization problems and only gives rather vague information about how to solve them on a computer. Algorithms for doing so are the subject of this section.

Assume a strictly convex function $f(x) : \mathbb{R}^n \to \mathbb{R}$ has to be minimized without constraints. Presuming the differentiability of the function, a standard algorithm is to start with some $x^0 \in \mathbb{R}^n$, to compute the gradient $\nabla f(x^0)$ either analytically or by finite difference approximations and to "move" in the direction of the negative gradient scaled by some step size $s^0 > 0$. This new location $x^1 := x^0 - s^0 \nabla f(x^0)$ has a smaller value than x^0, i.e. $f(x^1) < f(x^0)$ as long as the step size s^0 for this step was not chosen too large. Otherwise, the step size has to be reduced. When taking x^1 as a starting point for a new step and repeating this procedure over and over again, the sequence $x^0, x^1, x^2, \dots$ will converge to the optimal x^*. In practice the iterative process is stopped as soon as the desired accuracy necessary for the corresponding application is reached. This algorithm is known as *gradient descent* or the *method of steepest descent*.

The *dual ascent* algorithm transfers this idea to the solution of saddle-point problems (2.4.1). Assume for introductory reasons a saddle-point problem involves only linear equality constraints on the variables x, that is

$$\min_{x} \{ f(x) \mid Ax - b = 0 \} \ .$$

As the name suggests, dual ascent uses the Lagrangian dual function (definition 2.29) and takes advantage of its concavity. Since the Lagrangian dual function has to be maximized, steps are taken in "uphill" direction, that is the positive gradient direction. Due to the simple problem the gradient of the dual variables is $\nabla_y \bar{L}(y) = Ax - b$ where x minimizes the primal direction, i.e. $x = \arg\min_x L(x,y)$. The primal and dual variables are updated alternatingly [BPC+10]:

***Algorithm 2.1.* - Dual Ascent**

```
Fix $y^0 \in \mathbb{R}^m$
for $k = 0, 1, \ldots$ do
    $x^{k+1} = \arg\min_{x \in \mathbb{R}^n} L(x, y^k)$
    $y^{k+1} = y^k + s^k \left( Ax^{k+1} - b \right)$
end for
```

The step size $s^k > 0$ can be chosen arbitrary as long as $\bar{L}(y^{k+1}) < \bar{L}(y^k)$. In practice, methods aim for a maximal improvement in each step.

A very similar method to dual ascent is known as *method of multipliers*. It was developed to make dual ascent more robust and to allow less strict assumptions on the actual target function f according to [BPC+10]. Compared to dual ascent, it differs in the use of the augmented Lagrangian (2.4.2) in the minimization step of x and the step size s^k being set to the penalty parameter ρ. The price for convergence under far more general conditions is the loss of separability of the x-minimization step, so that parallel computation is not possible any more [BPC+10]. This issue is addressed by the method introduced next.

2.4.3.1 Alternating Direction Method of Multipliers

The *alternating direction method of multipliers*, or in short *ADMM*, recovers the decomposability of the method of multipliers. It can solve problems of the form

$$\min_{x,z} \{ f(x) + g(z) \mid Ax + Bz - c = 0 \}$$

with $x \in \mathbb{R}^n$, $z \in \mathbb{R}^m$, $A \in \mathbb{R}^{p \times n}$, $B \in \mathbb{R}^{p \times m}$ and $c \in \mathbb{R}^p$ given the epigraphs of f and g are nonempty closed convex sets and a saddle-point of the (unaugmented) Lagrangian function exists [BPC+10]. The corresponding augmented Lagrangian is

$$\begin{aligned} L_\rho(x, z, y) =& L(x, z, y) + \frac{\rho}{2} \|Ax + Bz - c\|_2^2 \\ =& f(x) + g(z) + y^\top (Ax + Bz - c) + \frac{\rho}{2} \|Ax + Bz - c\|_2^2 \; . \end{aligned}$$

Applying the method of multipliers to the problem, the first iterative step would be to minimize with respect to x and z at the same time. In ADMM this step is split i.e. variables are optimized consecutively.

Algorithm 2.2. - Alternating Direction Method of Multipliers (ADMM)

```
Fix $z^0 \in \mathbb{R}^m$, $y^0 \in \mathbb{R}^p$, $\rho > 0$
for $k = 0, 1, \dots$ do
    $x^{k+1} = \arg\min_{x \in \mathbb{R}^n} L_\rho(x, z^k, y^k)$
    $z^{k+1} = \arg\min_{z \in \mathbb{R}^m} L_\rho(x^{k+1}, z, y^k)$
    $y^{k+1} = y^k + \rho \left( Ax^{k+1} + Bz^{k+1} - c \right)$
end for
```

Practically more relevant is a modified form called *scaled ADMM* which combines the linear and the quadratic term of the augmented Lagrangian by introducing a new *scaled dual variable* $w := \frac{1}{\rho} y$. This can be an advantage for the inner optimization steps in x and z direction. By reformulating the last two terms of the augmented Lagrangian as

$$y^\top r + \frac{\rho}{2} \|r\|_2^2 = \frac{\rho}{2} \left\| r + \frac{1}{\rho} y \right\|_2^2 - \frac{1}{2\rho} \|y\|_2^2 = \frac{\rho}{2} \|r + w\|_2^2 - \frac{\rho}{2} \|w\|_2^2$$

with $r := Ax + Bz - c$ the equivalence between the two formulations can be shown [BPC+10] and leads to the following algorithm.

Algorithm 2.3. - Scaled ADMM

```
Fix $z^0 \in \mathbb{R}^m$, $w^0 \in \mathbb{R}^p$, $\rho > 0$
for $k = 0, 1, \dots$ do
    $x^{k+1} = \arg\min_{x \in \mathbb{R}^n} \left\{ f(x) + \frac{\rho}{2} \left\| Ax + Bz^k - c + w^k \right\|_2^2 \right\}$
    $z^{k+1} = \arg\min_{z \in \mathbb{R}^m} \left\{ g(z) + \frac{\rho}{2} \left\| Ax^{k+1} + Bz - c + w^k \right\|_2^2 \right\}$
    $w^{k+1} = w^k + Ax^{k+1} + Bz^{k+1} - c$
end for
```

In practice the convergence to high accuracy turned out to be very slow. Boyd writes further “However, it is often the case that ADMM converges to modest accuracy - sufficient for many application - within a few tens of iterations”. More details about ADMM can be found in his work [BPC+10].

2.4.3.2 Weak Coupling

Here, *weak coupling* is understood as the search for an solution to the problem

$$\min_{x,y} f(x) + g(y) + \|Ax - By\|^2 \tag{2.4.5}$$

with $x \in \mathbb{R}^n$, $y \in \mathbb{R}^m$, $A \in \mathbb{R}^{p \times n}$, $B \in \mathbb{R}^{p \times m}$ and the epigraphs of f and g are nonempty closed convex sets. The last term is called *coupling term* and induces dependencies between x and y. The probably most straight forward algorithm for solving it is the following.

Algorithm 2.4. - Weak Coupling Algorithm

```
Fix $x^0 \in \mathbb{R}^n$, $y^0 \in \mathbb{R}^m$, $\rho, \zeta, \eta > 0$
for $k = 0, 1, \ldots$ do
    $x^{k+1} = \arg\min_{x \in \mathbb{R}^n} f(x) + \frac{\rho}{2}\|Ax - By^k\|^2 + \frac{\zeta}{2}\|x - x^k\|^2$
    $y^{k+1} = \arg\min_{y \in \mathbb{R}^m} g(y) + \frac{\rho}{2}\|Ax^{k+1} - By\|^2 + \frac{\eta}{2}\|y - y^k\|^2$
end for
```

Attouch et al. [ABRS08] investigated the algorithm in detail and proved its convergence to a minimum of the target function. The additional quadratic terms $\|x - x^k\|^2$ and $\|y - y^k\|^2$ enforcing movement costs of the variables between subsequent iterations are essential for the convergence.

2.4.3.3 Chambolle and Pock's Algorithm

The first-order primal-dual algorithm investigated by Chambolle and Pock [CP11a] can solve saddle-point problems having the special form

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

with $x \in \mathbb{R}^n$, $y \in \mathbb{R}^m$ and a continuous linear operator $K : \mathbb{R}^n \to \mathbb{R}^m$, presuming that the functions f and g take values in $\mathbb{R}_+ \cup \{\infty\}$. Furthermore, their epigraphs must be nonempty closed convex sets and a saddle-point is required to exist. Chambolle and Pock show that the algorithm

Algorithm 2.5. - Chambolle's and Pock's first-order primal-dual algorithm (CP)

```
Fix $x^0 \in \mathbb{R}^n$, $y^0 \in \mathbb{R}^m$, $\theta \in [0, 1]$, $\tau > 0$, $\sigma > 0$, $\bar{x}^0 = x^0$
for $k = 0, 1, \ldots$ do
    $y^{k+1} = \arg\min_{y \in \mathbb{R}^m} \left\{ \frac{\left\|y - \left(y^k + \sigma K \bar{x}^k\right)\right\|^2}{2\sigma} + f^*(y) \right\}$
    $x^{k+1} = \arg\min_{x \in \mathbb{R}^n} \left\{ \frac{\left\|x - \left(x^k - \tau K^* y^{k+1}\right)\right\|^2}{2\tau} + g(x) \right\}$
    $\bar{x}^{k+1} = x^{k+1} + \theta\left(x^{k+1} - x^k\right)$
end for
```

converges to a saddle-point in the case $\theta = 1$ where τ, σ are chosen such that $\tau\sigma\|K\|^2 < 1$ with norm $\|K\| = \max_{x \in \mathbb{R}^n}\{\|Kx\| \mid \|x\| \leq 1\}$ [CP11a].

2.4.3.4 Parallel Proximal Algorithm

The Algorithm introduced in this section was first derived by Combettes and Pesquet [CP08, CP11b]. It is based on a special case of the Douglas-Rachford algorithm

[LM79] transformed into a product space and can solve problems of the form

$$\min_{x\in\mathbb{R}^n} \sum_{i=1}^{N} f_i(x) \tag{2.4.6}$$

with functions f_i having nonempty closed convex sets as epigraphs for all $i \in [N]$. Combettes and Pesquet's algorithm is called *parallel proximal algorithm (PPXA)* with reference to the *proximal operator* defined below and since most of the computation can be done in parallel.

Definition 2.30 *(proximal operator)*:
The function prox $f : \mathbb{R}^n \to \mathbb{R}^n$ defined by

$$\operatorname{prox} f(x) := \arg\min_{y\in\mathbb{R}^n} f(x) + \frac{1}{2}\|x-y\|_2^2$$

is called *proximal operator* or *proximity operator*.

Algorithm 2.6. - Parallel Proximal Algorithm (PPXA)

```
Fix $y_1^0, \dots, y_m^0 \in \mathbb{R}^n$, $\epsilon \in ]0,1[$, $\gamma > 0$ and $\omega \in ]0,1]^N$ such that $\sum_{i=1}^N \omega_i = 1$
$x^0 = \sum_{i=1}^N \omega_i y_i^0$
for $k = 0, 1, \dots$ do
    for $i = 1, \dots, N$ do
        $p_i^k = \operatorname{prox} \frac{\gamma}{\omega_i} f_i(y_i^k)$
    end for
    $p^k = \sum_{i=1}^N \omega_i p_i^k$
    Choose $\lambda^k \in [\epsilon, 2-\epsilon]$
    for $i = 1, \dots, N$ do
        $y_i^{k+1} = y_i^k + \lambda^k(2p^k - x^k - p_i^k)$
    end for
    $x^{k+1} = x^k + \lambda^k(p^k - x^k)$
end for
```

The parallel proximal algorithm 2.6 converges to a solution of (2.4.6) as long as

$$\bigcap_{i=1}^{N} (\operatorname{ri} \operatorname{dom} f_i) \neq \emptyset \qquad \text{and} \qquad \lim_{\|x\|\to\infty} \sum_{i=1}^{N} f_i \to \infty$$

where ri dom f_i is the relative interior of f_i's domain [CP11b].

2.5 Compressed Sensing

Compressed sensing, or in short CS, is a recent sampling theory extending classical results developed in the first half of the 20th century. The classical sampling theory is concerned with the recovery of continuous time signals from a discrete sequence of samples. The central result connects the sampling rate to the size of support (bandwidth) of the signal in some transformed domain, e.g. the Fourier domain.

The most famous names regarding sampling theory of this time are Claude Shannon and Harry Nyquist. In Shannon's work from 1949 [Sha49] that includes his famous sampling theorem, he generalized Nyquist's results from 1928 [Nyq28]. Today the two are considered parents of the classical sampling theory in the majority of the literature. However, there were others who discovered the same results in parallel or even earlier, such as Edmund Taylor Whittaker and his son John Macnaghten Whittaker. Vladimir Kotelnikov published similar results in 1933 for which he received the Eduard Rhein price late in 1999.

Nevertheless, the main result of classical sampling is the *Shannon-Nyquist sampling theorem* stating that the sampling rate is required to be greater than the bandwidth of a bandlimited signal that is a signal with a compact support in the Fourier domain. In compressed sensing, signals rather than being compactly supported have a small i.e. *sparse* support in some transformed domain. Compressed sensing was established by Candes, Tao [CT05, CRT06] and Donoho [Don06] in 2006 and connects the sampling rate to the signal sparsity.

The classical CS theory is concerned with the reconstruction of finite dimensional signals that is of vectors in $\mathbb{R}^n$ with large n. This defines also the current setting. The main constraint to the signal $x \in \mathbb{R}^n$ is the *sparsity* of x or at least of an alternative representation of x in a different domain, e.g. the Fourier domain.

Definition 2.31 *(sparsity)*:
A vector $x \in \mathbb{R}^n$ is called *s-sparse*, if it has at most $s \ll n$ non-zero entries, i.e. if

$$\|x\|_0 := |\{i \mid x_i \neq 0, i \in [n]\}| \leq s \in \mathbb{N} \ . \tag{2.5.1}$$

For convenience it is assumed that the signal itself is sparse in the current domain. The *recovery problem* can be expressed as a linear equation system

$$Ax^* = b$$

with $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$, $m < n$ which is usually highly underdetermined, i.e. $m \ll n$. Matrix A is called *sensor matrix* or just *sensor* and the right hand side b are the *observations* or *measurements* of the unknown signal x^*.

The basic recovery problem seeks for the sparsest solution of the linear system

$$\hat{x} = \arg\min_x \{\|x\|_0 \mid Ax = b\} \ . \tag{2.5.2}$$

This is a combinatorial NP-hard problem [Nat95] but it is known to yield the same sufficiently sparse solution as the ℓ_1-problem

$$\hat{x} = \arg\min_x \{\|x\|_1 \mid Ax = b\} \tag{2.5.3}$$

which is the convex relaxation of the ℓ_0-problem (2.5.2) [DH01]. In this form it can be solved by established solvers for linear programs. In more realistic scenarios when noisy observations are involved, it is reasonable to relax the strict equality constraint so that the recovery problem becomes

$$\hat{x} = \arg\min_x \{\|x\|_1 \mid \|Ax - b\|_2 \leq \eta\} \tag{2.5.4}$$

with an error $\eta > 0$. Here, the observations are assumed to be afflicted with an error $\|z\|_2 \leq \eta$ corresponding to $Ax^* = b + z$.

2.5.1 Uniform Recovery

A fundamental result of compressed sensing concerns the stable recovery of *any* s-sparse signal provided the sensor A satisfies certain conditions. These conditions ensure that any s-sparse signal can be recovered with one and the same sensor A. This kind of recovery is called *uniform recovery.* Compressed sensing can answer the question how accurate the uniform recovery of the true but unknown s-sparse signal x^* is using the formulations above.

The common way is to estimate bounds for the recovery error of the form

$$\|x - x^*\|_2 \leq c_1 \frac{\left\|x^*_{\max_s = 0}\right\|_p}{\sqrt{s}} + c_2 \eta \tag{2.5.5}$$

depending on two constants $c_1, c_2 > 0$, sparsity $s \in \mathbb{N}$, measurement error $\eta > 0$ and a term $\|x^*_{\max_s=0}\|_p$ estimating how distinct x^* is compared to an s-sparse signal. Here, $x^*_{\max_s=0}$ denotes the vector similar to x but with the s largest entries in absolute value set to 0, i.e.

$$(x_{\max_s=0})_i = \begin{cases} 0 & \text{if } i \in \arg\max_{\mathcal{S}} \left\{ \sum_{j \in \mathcal{S}} |x_j| \;\middle|\; |\mathcal{S}| \leq s \right\} \\ x_i & \text{otherwise} \end{cases} \ .$$

There is ongoing research about reducing the constants c_1 and c_2 in (2.5.5) and improving the bound tightness. An example is the result in (2.5.7) and below. In general, there are three different approaches to derive the aforementioned results, all depending on the nature of the sensor matrix A: Via the *null space property*, the *restricted isometry property* or the *mutual coherence.* Regarding subsequent notation, $S^c = [n] \setminus S$ means the complement of a set $S \subset [n]$ and $v_S \in \mathbb{R}^{|S|}$ denotes the vector containing the entries of a vector $v \in \mathbb{R}^n$ with indexes in S equally ordered.

Definition 2.32 *(null space property (NSP))*:
Let $A \in \mathbb{R}^{m\times n}$, $S \subset [n]$ and $s \in [n]$. If

$$\|v_S\|_1 < \|v_{S^c}\|_1 \qquad \text{for all} \qquad v \in \mathcal{N}(A) \setminus \{0\}$$

then A is said to have the *null space property (NSP)* relative to the set S. A is said to have the *null space property* of order s if

$$\|v_S\|_1 < \|v_{S^c}\|_1 \quad \text{for all} \quad v \in \mathcal{N}(A) \setminus \{0\} \quad \text{and all} \quad S \subset [n] \text{ with } |S| \leq s \ .$$

In practice, signals are rarely sparse but most often close to that and it is sufficient to recover an s-sparse approximation of the original signal. For handling this situation a slightly modified version of the NSP ensures *stable* recovery.

Definition 2.33 *(stable null space property)*:
Let $A \in \mathbb{R}^{m\times n}$, $S \subset [n]$, $s \in [n]$ and $0 < \rho < 1$. If

$$\|v_S\|_1 < \rho\|v_{S^c}\|_1 \qquad \text{for all} \qquad v \in \mathcal{N}(A)$$

then A is said to have the *stable null space property (stable NSP)* with constant ρ relative to the set S. A is said to have the *stable null space property* of order s with constant ρ if

$$\|v_S\|_1 < \rho\|v_{S^c}\|_1 \quad \text{for all} \quad v \in \mathcal{N}(A) \quad \text{and all} \quad S \subset [n] \text{ with } |S| \leq s \ .$$

In the case of the recovery of an arbitrary s-sparse vector $x^* \in \mathbb{R}^n$ by (2.5.3) from measurements $b = Ax^* \in \mathbb{R}^m$ the NSP is both a necessary and sufficient condition for exact recovery [FR13, section 4.1]. Finding matrices fulfilling the NSP is rather difficult. An alternative and sufficient condition for guaranteed recovery is the *RIP condition.*

Definition 2.34 *(restricted isometry constant and property)*:
Let $A \in \mathbb{R}^{m\times n}$ and $s \in [n]$. The *restricted isometry constant* $\delta_s = \delta_s(A)$ of order s is the smallest $\delta_s \geq 0$ such that

$$(1 - \delta_s)\,\|x\|_2^2 \leq \|Ax\|_2^2 \leq (1 + \delta_s)\,\|x\|_2^2 \tag{2.5.6}$$

for all s-sparse vectors $x \in \mathbb{R}^n$. If $\delta_s \in [0, 1[$ then A satisfies the *restricted isometry property (RIP condition)* of order s with constant δ_s.

Essentially the RIP condition means that every set of s or less columns of the sensor A behaves approximately like an orthonormal system [CRT06, FR13]. For a matrix with RIP condition the NSP is satisfied at the same time [CT05]. In this regard, a notable result is given by Cai [CWX10] stating that the reconstruction error for an s-sparse signal is bounded by

$$\|x - x^*\|_2 \leq \frac{\eta}{0.307 - \delta_s} \tag{2.5.7}$$

with x^* being the optimal solution to (2.5.4) under the condition

$$\delta_s < 0.307 \ .$$

In particular this means perfect recovery without an error z as long as the restricted isometry property (2.5.6) is fulfilled. If x is not s-sparse then the recovery error is bounded by

$$\|x - x^*\|_2 \leq \frac{1}{0.307 - \delta_s} \frac{\|x^*_{\max_s = 0}\|_1}{\sqrt{s}} + \frac{\eta}{0.307 - \delta_s} \ .$$

The third approach to derive recovery error bound constants is the *mutual coherence*. In general, checking whether a sensor matrix has the NSP or RIP condition requires lots of combinatorial computation. The *mutual coherence* is aimed at reducing this effort to a certain degree.

Definition 2.35 *(mutual coherence)*:
Let $A \in \mathbb{R}^{m \times n}$. The *mutual coherence* of A is the largest absolute inner product

$$\mu(A) = \max_{1 \leq i < j \leq n} \frac{|\langle A_i, A_j \rangle|}{\|A_i\|_2 \, \|A_j\|_2}$$

between any two columns A_i and A_j of A.

Roughly speaking, the smaller $\mu(A)$, the less sparsity of a signal is required for correct recovery. On the one hand $\mu(A)$ is relatively easy to compute, but on the other hand it yields larger constants in (2.5.5) compared to derivations using the NSP or the RIP condition. Good examples that fulfill above conditions with comparably large s are submatrices of the Vandermonde matrix or Gaussian random matrices which have optimal properties in this regard [FR13, Chapter 9].

2.5.2 Individual Recovery

The ℓ_1-problem (2.5.3) can have multiple solutions of the same sparsity s and the previous section mentions tool for deriving recovery error bounds. Often an efficient verification is required which checks whether a given solution is unique. The uniqueness can be verified with the help of so called *dual certificates* which can be derived from the *dual* problem to (2.5.3).

Theorem 2.36 *(dual certificates [FR13, Theorem 4.26.(b)])*:
Let $x^* \in \mathbb{R}^n$ with $Ax^* = b$, $S = \operatorname{supp}(x^*) \subset [n]$ and $s = \operatorname{sign}(x^*_S)$. Then

$$\arg\min_x \{\|x\|_1 \mid Ax = b\} = \{x^*\}$$

if and only if the following two conditions *(dual certificates)* hold:

1. There exists $y \in \mathbb{R}^m$ such that $A_S^\top y = s$ and $\|A_{S^c}^\top y\|_\infty < 1$,
2. A_S is injective.

Due to the strict inequality constraint in condition 1 it requires some reformulation to get a simple linear programming test that can be used together with standard rank computation for verifying uniqueness of a solution to the ℓ_1-problem [KP18, Theorem 2.3].

2.5.3 Probabilistic Recovery

In practice, the recovery of signals often succeeds even though no guarantee can be given a priori. While uniform recovery guarantees require very strong assumptions on the sensor matrix, guaranteeing that an s-sparse solution is most likely unique is often equally desirable and often succeeds under weaker conditions on the sensor A. For simplicity, however, the Gaussian case is considered.

In the following, let $X \in \mathbb{R}^n$ be a Gaussian random vector with independent and identical standard normally distributed entries, that is $X \sim \mathcal{N}(0, I_n)$. Further, $S^{n-1} := \{x \in \mathbb{R}^n \mid \|x\|_2 = 1\}$ is the unit spere centered at $0 \in \mathbb{R}^n$ and the *polar cone* of a cone K is denoted by $K^\circ := \left\{y \in \mathbb{R}^n \,\middle|\, y^\top x \leq 0 \text{ for all } x \in K\right\}$.

The recovery guarantees from this section are expressed in terms of two related measures, the *Gaussian width* and the *statistical dimension*.

Definition 2.37 *(conic Gaussian width [FR13, Section 9.2])*:
The *conic Gaussian width* of a cone $K \subset \mathbb{R}^n$ is defined as

$$\omega(K) = \mathbb{E}\left(\sup_{y \in K \cap S^{n-1}} X^\top y\right) .$$

Definition 2.38 *(statistical dimension [ALMT14, Section 3.1.])*:
The *statistical dimension* of a convex cone $K \subset \mathbb{R}^n$ is defined as[1]

$$\delta(K) = \mathbb{E}\left(\operatorname{dist}^2(X, K^\circ)\right)$$

where $\operatorname{dist}(x, S) := \inf_{y \in S} \|x - y\|_2$.

Both the conic Gaussian width and the statistical dimension are approximately equal as the following theorem shows and can thus be used interchangeable [Tro15].

[1] The missing subscript distinguishes the statistical dimension from the indicator function (2.4.4).

Theorem 2.39 *(relation between conic Gaussian width and statistical dimension [ALMT14, Proposition 10.2])*:
Let $K \subset \mathbb{R}^n$ be a convex cone. Then

$$\omega^2(K) \leq \delta(K) \leq \omega^2(K) + 1 \ .$$

Estimating the Gaussian width - or the statistical dimension - of the *descent cone* comprising all descent directions will turn out utterly important for signal recovery guarantees.

Definition 2.40 *(descent cone [Tro15, Definition 2.4])*:
Let $f : \mathbb{R}^n \to \overline{\mathbb{R}}$ be a proper convex function. The set

$$D(f, x) = \bigcup_{\tau > 0} \{y \in \mathbb{R}^n \,|\, f(x + \tau y) \leq f(x)\}$$

is called the *descent cone* of the function f at a point $x \in \mathbb{R}^n$.

Theorem 2.41 *(optimality condition [FR13, Theorem 4.35.])*:
Let $f : \mathbb{R}^n \to \overline{\mathbb{R}}$ be a proper convex function and $A \in \mathbb{R}^{m \times n}$. The vector $x^* \in \mathbb{R}^n$ is the unique optimal solution to the convex program

$$x^* = \arg\min_x \{f(x) \,|\, Ax = b\} \tag{2.5.8}$$

with measurements $b = Ax^* \in \mathbb{R}^m$ if and only if

$$D(f, x^*) \cap \mathcal{N}(A) = \{0\} \tag{2.5.9}$$

where $\mathcal{N}(A)$ denotes the nullspace of A.

The probability for (2.5.9) can be estimated with respect to the problem size n and the number of measurements m by utilizing the two measures above. The smaller the descent cone $D(f, x^*)$, the more unlikely an intersection with the nullspace $\mathcal{N}(A)$.

Theorem 2.42 *(phase transition for random measurements [ALMT14, Theorem II])*:
Fix a tolerance $p \in]0, 1[$ and let $f : \mathbb{R}^n \to \overline{\mathbb{R}}$ be a proper convex function. Further, let the sensor matrix $A \in \mathbb{R}^{m \times n}$ have independent standard normal entries and sample a signal $x^* \in \mathbb{R}^n$ so that the measurements are $b = Ax^* \in \mathbb{R}^m$. Then, the signal recovery via (2.5.8) succeeds with probability at least $1 - p$ if

$$m \geq \delta(D(f, x^*)) + c_p \sqrt{n}$$

and fails with probability at least $1 - p$ if

$$m \leq \delta(D(f, x^*)) - c_p \sqrt{n}$$

where $c_p := \sqrt{8 \log\left(\frac{4}{p}\right)}$.

Note that theorem 2.42 does not give a statement about the situation when

$$m \in T_p := \left]\delta(D(f,x^*)) - c_p\sqrt{n},\ \delta(D(f,x^*)) + c_p\sqrt{n}\right[\ .$$

On average, achieving a successful recovery of the signal turns from unlikely to likely at $m = \delta(D(f,x^*))$ measurements. This phenomenon is known as *phase transition*. The *transition zone* is T_p, the interval where the shift from failure to success happens. Outside the transition zone the statement about the recovery success is relatively secure. Therefore, x^* can typically be recovered if there are

$$m \geq \delta(D(f,x^*)) \tag{2.5.10}$$

measurements available [ALMT14].

Naturally, exact recovery cannot be expected in the case of measurement errors. A result for the probability of a bounded recovery error is the following.

Theorem 2.43 *(recovery from random measurements [Tro15, Corollary 3.5])*:
Let $x^* \in \mathbb{R}^n$ the desired signal and let the columns of the sensor matrix $A \in \mathbb{R}^{m\times n}$ be independent Gaussian random vectors drawn from the standard normal distribution $\mathcal{N}(0, I_n)$. Further, let the measurements $b = Ax + z \in \mathbb{R}^m$ be afflicted with an error z with $\|z\|_2 \leq \eta$ and let x_η be any solution the problem

$$\min_x \{f(x) \,|\, \|Ax - b\|_2 \leq \eta\} \ .$$

Then it holds for the recovery error

$$\mathbb{P}\left(\|x - x^*\|_2 \leq \frac{2\eta}{\left(\sqrt{m-1} - \omega(D(f,x^*)) - t\right)^+}\right) \leq 1 - e^{-\frac{t^2}{2}}$$

with $(\cdot)^+ := \max\{0, \cdot\}$.

Theorem 2.43 provides stable recovery for x^* if the number of measurements m is lower bounded by

$$m \geq \omega^2\left(D(f,x^*)\right) + c\,\omega\left(D(f,x^*)\right) \tag{2.5.11}$$

with a constant $c > 0$ [Tro15].

In general it is impossible to calculate the Gaussian width or statistical dimension exactly. However, upper bounds can be derived which are sufficient for the estimation of probabilistic recovery guarantees in terms of (2.5.10) and (2.5.11). Those bounds utilize the subdifferential (definition 2.28).

Theorem 2.44 *(upper bound for the Gaussian width and statistical dimension of a descent cone [ALMT14, Proposition 4.1][Tro15, Proposition 4.5])*:
Let $f : \mathbb{R}^n \to \overline{\mathbb{R}}$ be a proper convex function and $x \in \mathbb{R}^n$. Assume that the subdifferential $\partial f(x)$ is nonempty, compact and does not contain the origin. The

Gaussian width and the statistical dimension, respectively, are upper bounded by

$$\omega^2\left(D(f,x)\right) \leq \inf_{\tau\geq 0} J(\tau) \qquad \text{and} \qquad \delta\left(D(f,x)\right) \leq \inf_{\tau\geq 0} J(\tau)$$

where

$$J(\tau) := J(\tau, \partial f(x)) := \mathbb{E}\left(\operatorname{dist}^2\left(X, \tau\partial f(x)\right)\right) \quad \text{for} \quad \tau \geq 0.$$

The function J is strictly convex, continuous at $\tau = 0$, differentiable for $\tau \geq 0$ and thus achieves its minimum at a unique point.

2.5.3.1 Statistical Dimension Bound Estimations

When recovering an s-sparse signal x^* via (2.5.8), equation (2.5.10) gives a lower bound on the required number of measurements m for a likely success, which is in turn upper bounded by theorem 2.44. In particular, the aim is to recover an s-sparse *and* nonnegative signal or an s-sparse *and* binary signal. Hence, in this section the upper bound $\inf_{\tau\geq 0} J(\tau)$ is derived for the corresponding relaxed convex models that are the ℓ_1-recovery with nonnegative constraints

$$x^* = \arg\min_x \left\{\|x\|_1 \,|\, Ax = b, x \geq 0\right\} \tag{2.5.12}$$

and with box constraints

$$x^* = \arg\min_x \left\{\|x\|_1 \,|\, Ax = b, x \in [0,1]^n\right\} \ . \tag{2.5.13}$$

ℓ_1-Recovery of Nonnegative s-Sparse Signals

With the help of the indicator function (2.4.4) problem (2.5.12) can be reformulated as

$$x^* = \arg\min_x \left\{\|x\|_1 + \delta_{\mathbb{R}^n_+}(x) \,\middle|\, Ax = b\right\}$$

in order to agree with (2.5.8) so that $f(x) := \|x\|_1 + \delta_{\mathbb{R}^n_+}(x)$. The first step is to derive $\partial f(x)$ where the subdifferential sum rule applies [Roc97, Theorem 23.8.] leading to

$$\partial f(x) = \partial\left(\|\cdot\|_1 + \delta_{\mathbb{R}^n_+}\right)(x) = \partial\|\cdot\|_1(x) + \partial\delta_{\mathbb{R}^n_+}(x) \ .$$

Each component of $\partial\|\cdot\|_1$ equals the subdifferential of the absolute value, i.e.

$$\left(\partial\|\cdot\|_1(x)\right)_i = \partial|\cdot|(x) = \begin{cases} \operatorname{sign}(x) & \text{if } x \neq 0 \\ [-1,1] & \text{if } x = 0 \end{cases}$$

with the sign function

$$\operatorname{sign}(x) := \begin{cases} -1 & \text{if } x < 0 \\ 0 & \text{if } x = 0 \\ 1 & \text{if } x > 0 \end{cases} \ .$$

Applying the sign function componentwise to x_S leads to

$$\partial\|\cdot\|_1(x) = \{y \in \mathbb{R}^n \,|\, y_S = \operatorname{sign}(x_S), \|y_{S^c}\|_\infty \leq 1\}$$

and since x^* is assumed to be positive, it is sufficient to consider

$$\begin{aligned} \partial\|\cdot\|_1(x^*) &= \left\{y \in \mathbb{R}^n \,\middle|\, y_S = \mathbb{1}_{|S|}, \|y_{S^c}\|_\infty \leq 1\right\} \\ &= \left\{y \in \mathbb{R}^n \,\middle|\, y_S = \mathbb{1}_{|S|}, y_{S^c} \in [-1,1]^{|S^c|}\right\} \ . \end{aligned} \tag{2.5.14}$$

Because of (2.4.3) every component of $\partial\delta_{\mathbb{R}^n_+}(x^*)$ equals

$$(\partial\delta_{\mathbb{R}^n_+}(x^*))_i = N_{\mathbb{R}_+}(x_i^*) = \begin{cases} 0 & \text{if } x^* > 0 \\ ]-\infty, 0] & \text{if } x^* = 0 \end{cases}$$

so that

$$\partial\delta_{\mathbb{R}^n_+}(x^*) = \{0\}^{|S|} \times \mathbb{R}_-^{|S_c|} = \{y \in \mathbb{R}^n \,|\, y_S = 0,\, y_{S^c} \leq 0\} \ . \tag{2.5.15}$$

Summing the two subdifferentials (2.5.14) and (2.5.15) yields

$$\partial f(x^*) = \partial\|\cdot\|_1(x^*) + \partial\delta_{\mathbb{R}^n_+}(x^*) = \left\{y \in \mathbb{R}^n \,\middle|\, y_S = \mathbb{1}_{|S|}, y_{S^c} \leq \mathbb{1}_{|S^c|}\right\}$$

and its scaled version

$$\tau\partial f(x^*) = \left\{y \in \mathbb{R}^n \,\middle|\, y_S = \tau\mathbb{1}_{|S|}, y_{S^c} \leq \tau\mathbb{1}_{|S^c|}\right\} \ .$$

Further it holds

$$\operatorname{dist}^2(X, \tau\partial f(x^*)) = \inf_y \left\{\|X - y\|_2^2 \,\middle|\, y \in \tau\partial f(x^*)\right\} = \left\|X - \Pi_{\tau\partial f(x^*)}(X)\right\|_2^2 \tag{2.5.16}$$

where every component of the projection is

$$\left(\Pi_{\tau\partial f(x^*)}(X)\right)_i = \begin{cases} \tau & \text{if } i \in S \\ X_i & \text{if } i \in S^c, X_i \leq \tau \\ \tau & \text{if } i \in S^c, X_i > \tau \end{cases}$$

so that

$$\left(X - \Pi_{\tau\partial f(x^*)}(X)\right)_i = \left\{\begin{array}{ll} X_i - \tau & \text{if } i \in S \\ 0 & \text{if } i \in S^c, X_i \leq \tau \\ X_i - \tau & \text{if } i \in S^c, X_i > \tau \end{array}\right\} = \begin{cases} X_i - \tau & \text{if } i \in S \\ \max\{X_i - \tau, 0\} & \text{if } i \in S^c \end{cases}$$

are the entries of the difference. Now, it is possible to compute the expected squared

Euclidean distance of a normal vector to the scaled subdifferential of theorem 2.44 as

$$\begin{aligned} J_s(\tau) &= \mathbb{E}\left(\operatorname{dist}^2\left(X, \tau\partial f(x^*)\right)\right) \\ &= \mathbb{E}\left(\sum_{i\in S}(X_i-\tau)^2 + \sum_{i\in S^c}\max\{X_i-\tau,0\}^2\right) \\ &= \sum_{i\in S}\mathbb{E}\left((X_i-\tau)^2\right) + \sum_{i\in S^c}\mathbb{E}\left(\max\{X_i-\tau,0\}^2\right) \\ &= \sum_{i\in S}\frac{1}{\sqrt{2\pi}}\int_{-\infty}^{\infty}(x_i-\tau)^2 e^{-\frac{x_i^2}{2}}\,\mathrm{d}x_i + \sum_{i\in S^c}\frac{1}{\sqrt{2\pi}}\int_{\tau}^{\infty}(x_i-\tau)^2 e^{-\frac{x_i^2}{2}}\,\mathrm{d}x_i \\ &= \frac{s}{\sqrt{2\pi}}\int_{-\infty}^{\infty}(x-\tau)^2 e^{-\frac{x^2}{2}}\,\mathrm{d}x + \frac{n-s}{\sqrt{2\pi}}\int_{\tau}^{\infty}(x-\tau)^2 e^{-\frac{x^2}{2}}\,\mathrm{d}x \\ &= s(1+\tau^2) + \frac{n-s}{\sqrt{2\pi}}\int_{\tau}^{\infty}(x-\tau)^2 e^{-\frac{x^2}{2}}\,\mathrm{d}x\ . \end{aligned} \tag{2.5.17}$$

In order to apply theorem 2.44 the infimum $\inf_{\tau\geq 0} J_s(\tau)$ is required. However, this cannot be computed explicitly but it is possible to derive an upper bound. In this regard, integration by parts gives

$$\begin{aligned} &\int_{\tau}^{\infty}(x-\tau)^2 e^{-\frac{x^2}{2}}\,\mathrm{d}x = \int_{\tau}^{\infty}(x^2-2x\tau+\tau^2)e^{-\frac{x^2}{2}}\,\mathrm{d}x \\ =& -\int_{\tau}^{\infty} x\left(-xe^{-\frac{x^2}{2}}\right)\mathrm{d}x + 2\tau\int_{\tau}^{\infty} -xe^{-\frac{x^2}{2}}\,\mathrm{d}x + \tau^2\int_{\tau}^{\infty} e^{-\frac{x^2}{2}}\,\mathrm{d}x \\ =& -\left(\left[xe^{-\frac{x^2}{2}}\right]_{\tau}^{\infty} - \int_{\tau}^{\infty} e^{-\frac{x^2}{2}}\,\mathrm{d}x\right) + 2\tau\left[e^{-\frac{x^2}{2}}\right]_{\tau}^{\infty} + \tau^2\int_{\tau}^{\infty} e^{-\frac{x^2}{2}}\,\mathrm{d}x \\ =& -\tau e^{-\frac{\tau^2}{2}} + (1+\tau^2)\int_{\tau}^{\infty} e^{-\frac{x^2}{2}}\,\mathrm{d}x \end{aligned} \tag{2.5.18}$$

yielding

$$\begin{aligned} J_s(\tau) &= s(1+\tau^2) + \frac{n-s}{\sqrt{2\pi}}\left(-\tau e^{-\frac{\tau^2}{2}} + (1+\tau^2)\int_{\tau}^{\infty} e^{-\frac{x^2}{2}}\,\mathrm{d}x\right) \\ &= s(1+\tau^2) - \frac{\tau(n-s)}{\sqrt{2\pi}}e^{-\frac{\tau^2}{2}} + \frac{(n-s)(1+\tau^2)}{\sqrt{2\pi}}\int_{\tau}^{\infty} e^{-\left(\frac{x}{\sqrt{2}}\right)^2}\,\mathrm{d}x \\ &= s(1+\tau^2) - \frac{\tau(n-s)}{\sqrt{2\pi}}e^{-\frac{\tau^2}{2}} + \frac{(n-s)(1+\tau^2)}{2}\underbrace{\frac{2}{\sqrt{\pi}}\int_{\frac{\tau}{\sqrt{2}}}^{\infty} e^{-u^2}\,\mathrm{d}u}_{=\operatorname{erfc}\left(\frac{\tau}{\sqrt{2}}\right)} \end{aligned} \tag{2.5.19}$$

after insertion into (2.5.17) and substitution in the integral by $u := \frac{x}{\sqrt{2}}$ and, consequently, $\mathrm{d}x = \sqrt{2}\,\mathrm{d}u$. The last integral can be expressed in terms of the *complementary error function* $\operatorname{erfc}\left(\frac{\tau}{\sqrt{2}}\right) = 1 - \operatorname{erf}\left(\frac{\tau}{\sqrt{2}}\right)$ commonly available in numerical software so that $\inf_{\tau\geq 0} J_s(\tau)$ can be computed easily. Moreover, the Gaussian upper

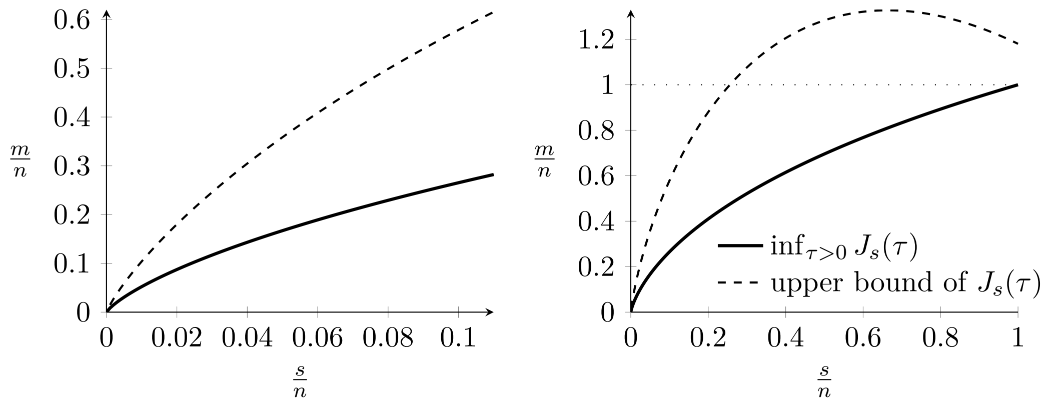


Figure 2.3 - Exact upper bound $\inf_{\tau\geq 0} J_s(\tau)$ of the Gaussian width and statistical dimension (2.5.19) together with the estimate of an upper bound according to (2.5.20) for the ℓ_1-recovery of nonnegative s-sparse signals. If the relative undersampling ration $\frac{m}{n}$ lies on the continuous curve or above then recovery from Gaussian measurements of an s-sparse nonnegative signal is guaranteed.

tail bound [FR13, Lemma C.7.] allows to estimate

$$\begin{aligned} J_s(\tau) &\leq s(1+\tau^2) + \frac{n-s}{\sqrt{2\pi}}\left(-\tau e^{-\frac{\tau^2}{2}} + \frac{1+\tau^2}{\tau} e^{-\frac{\tau^2}{2}}\right) \\ &= s(1+\tau^2) + \frac{n-s}{\tau\sqrt{2\pi}}\, e^{-\frac{\tau^2}{2}} \ . \end{aligned}$$

Even tough theorem 2.44 requires a compact subdifferential which is not the case here, [KP18] show that it still holds. By exploiting the fact that $\delta\left(D(f,x^*)\right) \leq \inf_{\tau\geq 0} J(\tau)$ in theorem 2.44 implies $\delta\left(D(f,x^*)\right) \leq J(\tau)$ for all $\tau \geq 0$, the value of τ can be chosen arbitrarily. Here, a bound with respect to the relative sparsity $\frac{n}{s}$ is desired justifying the choice $\tau = \sqrt{2\log\left(\frac{n}{s}\right)}$. Thus,

$$\begin{aligned} \delta\left(D(f,x^*)\right) &\leq s\left(1 + 2\log\left(\frac{n}{s}\right)\right) + \frac{n-s}{2\sqrt{\pi}\sqrt{\log\left(\frac{n}{s}\right)}}\,\frac{s}{n} \\ &= s\left(1 + 2\log\left(\frac{n}{s}\right)\right) + s\frac{1-\frac{s}{n}}{2\sqrt{\pi\log\left(\frac{n}{s}\right)}} \end{aligned}$$

and by (2.5.10) and since $\frac{1-\frac{s}{n}}{2\sqrt{\pi\log\left(\frac{n}{s}\right)}} \leq 0.1801$ for $0 \leq s \leq n$, the recovery via (2.5.12) is likely to succeed if there are

$$m \geq 2s\log\left(\frac{n}{s}\right) + 1.1801\, s \tag{2.5.20}$$

measurements available. Both $\inf_{\tau\geq 0} J_s(\tau)$ and the upper bound (2.5.20) of $J_s(\tau)$ are illustrated in figure 2.3.

ℓ_1-Recovery of Binary s-Sparse Signals

In the case of binary constraints on the signal entries it is appropriate to consider the convex relaxation

$$x^* = \arg\min_x \{\|x\|_1 \,|\, Ax = b, x \in [0,1]^n\}$$

for recovering a binary signal $x^* \in \{0,1\}^n$. The derivation is analogous to the positive case. It starts with the derivation of the subdifferential

$$\partial f(x^*) = \partial\|\cdot\|_1(x^*) + \partial\delta_{[0,1]^n}(x^*) \ .$$

The first term is already given by (2.5.14) whereas each component of the second term equals

$$(\partial\delta_{[0,1]^n}(x^*))_i = N_{[0,1]}(x_i^*) = \begin{cases} 0 & \text{if } x^* \in]0,1[\\ ]-\infty,0] & \text{if } x^* = 0 \\ [0,\infty[& \text{if } x^* = 1 \end{cases} \ .$$

Consequently,

$$\partial\delta_{[0,1]^n}(x^*) = \mathbb{R}_+^{|S|} \times \mathbb{R}_-^{|S_c|} = \{y \in \mathbb{R}^n \,|\, y_S \geq 0,\, y_{S^c} \leq 0\} \tag{2.5.21}$$

and summing the two subdifferentials (2.5.14) and (2.5.21) yields

$$\partial f(x^*) = \partial\|\cdot\|_1(x^*) + \partial\delta_{[0,1]^n}(x^*) = \left\{y \in \mathbb{R}^n \,\middle|\, y_S \geq \mathbb{1}_{|S|}, y_{S^c} \leq \mathbb{1}_{|S^c|}\right\}$$

and the corresponding scaled version

$$\tau\partial f(x^*) = \left\{y \in \mathbb{R}^n \,\middle|\, y_S \geq \tau\mathbb{1}_{|S|}, y_{S^c} \leq \tau\mathbb{1}_{|S^c|}\right\} \ .$$

In contrast to the projection (2.5.16) of X onto $\tau\partial f(x^*)$ of the positive case this results in an additional case for each component of the difference

$$\left(X - \Pi_{\tau\partial f(x^*)}(X)\right)_i = \begin{Bmatrix} 0 & \text{if } i \in S, X_i \geq 0 \\ \tau - X_i & \text{if } i \in S, X_i < 0 \\ 0 & \text{if } i \in S^c, X_i \leq \tau \\ X_i - \tau & \text{if } i \in S^c, X_i > \tau \end{Bmatrix} = \begin{cases} \max\{\tau - X_i, 0\} & \text{if } i \in S \\ \max\{X_i - \tau, 0\} & \text{if } i \in S^c \end{cases}$$

so that

$$\begin{aligned}J_s(\tau) &= \mathbb{E}\left(\sum_{i\in S}\max\{\tau - X_i, 0\}^2 + \sum_{i\in S^c}\max\{X_i-\tau,0\}^2\right)\\ &= \sum_{i\in S}\mathbb{E}\left(\max\{\tau-X_i,0\}^2\right) + \sum_{i\in S^c}\mathbb{E}\left(\max\{X_i-\tau,0\}^2\right)\\ &= \sum_{i\in S}\frac{1}{\sqrt{2\pi}}\int_{-\infty}^{\tau}(x_i-\tau)^2 e^{-\frac{x_i^2}{2}}\,\mathrm{d}x_i + \sum_{i\in S^c}\frac{1}{\sqrt{2\pi}}\int_{\tau}^{\infty}(x_i-\tau)^2 e^{-\frac{x_i^2}{2}}\,\mathrm{d}x_i\\ &= \frac{s}{\sqrt{2\pi}}\int_{-\infty}^{\tau}(x-\tau)^2 e^{-\frac{x^2}{2}}\,\mathrm{d}x + \frac{n-s}{\sqrt{2\pi}}\int_{\tau}^{\infty}(x-\tau)^2e^{-\frac{x^2}{2}}\,\mathrm{d}x\ .\end{aligned} \tag{2.5.22}$$

At this point it can already be seen that $\inf_{\tau\geq 0} J_s(\tau) \leq \frac{n}{2}$ since $X_i \sim \mathcal{N}(0,1)$ and inserting $\tau = 0$ yields

$$J_s(0) = \frac{s}{\sqrt{2\pi}}\int_{-\infty}^{0} x^2 e^{-\frac{x^2}{2}}\,\mathrm{d}x + \frac{n-s}{\sqrt{2\pi}}\int_{0}^{\infty}x^2 e^{-\frac{x^2}{2}}\,\mathrm{d}x = \frac{s}{2}+\frac{n-s}{2} = \frac{n}{2}\ . \tag{2.5.23}$$

Further reformulation of (2.5.22) leads to

$$\begin{aligned}J_s(\tau) &= \frac{s}{\sqrt{2\pi}}\int_{-\infty}^{\infty}(x-\tau)^2e^{-\frac{x^2}{2}}\,\mathrm{d}x + \frac{n-2s}{\sqrt{2\pi}}\int_{\tau}^{\infty}(x-\tau)^2 e^{-\frac{x^2}{2}}\,\mathrm{d}x\\ &= s(1+\tau^2) + \frac{n-2s}{\sqrt{2\pi}}\int_{\tau}^{\infty}(x-\tau)^2e^{-\frac{x^2}{2}}\,\mathrm{d}x\\ &\overset{(2.5.18)}{=} s(1+\tau^2) + \frac{n-2s}{\sqrt{2\pi}}\left(-\tau e^{-\frac{\tau^2}{2}} + (1+\tau^2)\int_{\tau}^{\infty}e^{-\frac{x^2}{2}}\,\mathrm{d}x\right)\\ &= s(1+\tau^2) - \frac{\tau(n-2s)}{\sqrt{2\pi}}e^{-\frac{\tau^2}{2}} + \frac{(n-2s)(1+\tau^2)}{\sqrt{2\pi}}\int_{\tau}^{\infty}e^{-\left(\frac{x}{\sqrt{2}}\right)^2}\,\mathrm{d}x\\ &= s(1+\tau^2) - \frac{\tau(n-2s)}{\sqrt{2\pi}}e^{-\frac{\tau^2}{2}} + \frac{(n-2s)(1+\tau^2)}{2}\underbrace{\frac{2}{\sqrt{\pi}}\int_{\frac{\tau}{\sqrt{2}}}^{\infty}e^{-u^2}\,\mathrm{d}u}_{=\mathrm{erfc}\left(\frac{\tau}{\sqrt{2}}\right)}\ .\end{aligned} \tag{2.5.24}$$

Again, $J_s(\tau)$ can be estimated with the help of the Gaussian upper tail bound [FR13, Lemma C.7.] as

$$\begin{aligned}J_s(\tau) &\leq s(1+\tau^2) + \frac{n-2s}{\sqrt{2\pi}}\left(-\tau e^{-\frac{\tau^2}{2}} + \frac{1+\tau^2}{\tau}e^{-\frac{\tau^2}{2}}\right)\\ &= s(1+\tau^2) + \frac{n-2s}{\tau\sqrt{2\pi}}\,e^{-\frac{\tau^2}{2}}\end{aligned}$$

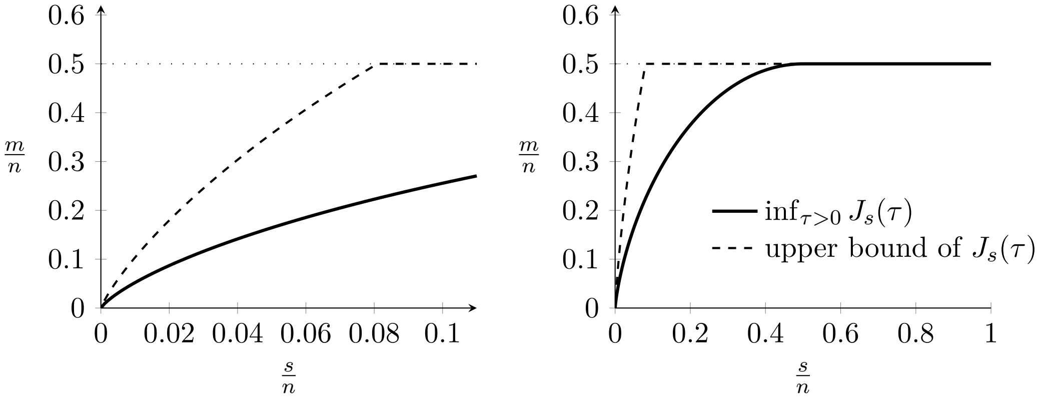


Figure 2.4 - Exact upper bound $\inf_{\tau\geq 0} J_s(\tau)$ of the Gaussian width and statistical dimension (2.5.24) together with the estimate of an upper bound according to (2.5.25) for the relaxed ℓ_1-recovery of binary s-sparse signals. If the relative undersampling ration $\frac{m}{n}$ lies on the continuous curve or above then recovery from Gaussian measurements of an s-sparse binary signal is guaranteed. In this case there are never more than $\frac{m}{n} = 0.5$ measurements required to achieve perfect recovery for any sparsity s.

and similarly choosing $\tau = \sqrt{2\log\left(\frac{n}{s}\right)}$ yields

$$\begin{aligned}\delta\left(D(f,x^*)\right) &\leq s\left(1 + 2\log\left(\frac{n}{s}\right)\right) + \frac{n-2s}{2\sqrt{\pi}\sqrt{\log\left(\frac{n}{s}\right)}}\,\frac{s}{n} \\ &= s\left(1 + 2\log\left(\frac{n}{s}\right)\right) + s\frac{\frac{1}{2}-\frac{s}{n}}{\sqrt{\pi\log\left(\frac{n}{s}\right)}}\end{aligned}$$

Estimating $\frac{\frac{1}{2}-\frac{s}{n}}{\sqrt{\pi\log\left(\frac{n}{s}\right)}} \leq 0.1492$ for $0 \leq s \leq n$ leads to a tighter bound for small sparsity s than (2.5.23) so that the lower bound of measurements for a likely recovery via (2.5.13) is given by

$$m \geq \min\left\{2s\log\left(\frac{n}{s}\right) + 1.1492\,s,\ \frac{n}{2}\right\}\ . \tag{2.5.25}$$

The exact value $\inf_{\tau\geq 0} J_s(\tau)$ and its upper bound (2.5.25) of $J_s(\tau)$ are illustrated in figure 2.4. In this case a binary signal can always be recovered from Gaussian measurements if $m \geq \frac{n}{2}$.

2.5.4 Nonstandard Tomography and Compressed Sensing

The pillars of the classical CS theory are sparsity of the signal in some basis or after a transformation and nonadaptive random measurements. The later does not apply to tomographic sensing since other kinds of sensors are used. Those are introduced in section 2.5.4.1 and allow to reduce the dimension of the sensing system as described in section 2.5.4.2. However, tomographic sensors of this kind do not confirm with the classical CS theory, but the interpretation as expander graphs presented in section 2.5.4.3 and further results in section 2.5.4.4 helps to close the gap.

2.5.4.1 Tomographic Sensors

In tomography the main concern is to reconstruct a d-dimensional volume from projections of dimension $d-1$. In general, the volume is subdivided into cells by a regular grid. Given the projections as measurements, the aim of a tomographic reconstruction is to compute a single value for each cell representing a physical property, e.g. substance density. In Tomo-PIV (chapter 1), for example, these projections are images recorded by a few cameras arranged around a volume of interest.

Speaking in terms of compressed sensing these images are the observations arranged in vector b and the desired cell values of the volume are represented by the signal x. The sensor A comprises the experimental setup, i.e. contains the information about the formation process of every single image pixel being captured. Each entry of a tomographic projection matrices typically equals the length of the intersection segment of each projection ray with each pixel.

For simplicity binary matrices are considered below that arise by certain geometries, e.g. orthogonal projecting directions as shown in figure 2.5 for dimension $D=3$. Such sensors have the form

$$A_d^D := \begin{bmatrix} \mathbb{1}_d^\top \otimes I_{d^{D-1}} \\ I_d \otimes \mathbb{1}_d^\top \otimes I_{d^{D-2}} \\ \vdots \\ I_{d^{i-1}} \otimes \mathbb{1}_d^\top \otimes I_{d^{D-i}} \\ \vdots \\ I_{d^{D-2}} \otimes \mathbb{1}_d^\top \otimes I_d \\ I_{d^{D-1}} \otimes \mathbb{1}_d^\top \end{bmatrix} \in \{0,1\}^{Dd^{D-1} \times d^D} = \{0,1\}^{m \times n} \tag{2.5.26}$$

where $\otimes$ denotes the Kronecker product, which are capable of acquiring measurements from a D-dimensional volume discretized into a grid with side length d. In this geometric interpretation of the sensor, the samples are obtained from D orthogonal projections onto $(D-1)$-dimensional grids having side length d and being parallel to the grid axes of the D-volume. If one of the $m = Dd^{D-1}$ projection rays intersects one of the $n = d^D$ voxels the corresponding entry in A_d^D is 1 and otherwise 0.

In these tomographic scenarios signals and sensor matrices are commonly nonnegative leading to the recovery problem

$$\hat{x} = \arg\min_x \{\|x\|_1 \mid Ax = b, x \geq 0\} \tag{2.5.27}$$

with $A_{i,j} \geq 0$ and $b_i \geq 0$.

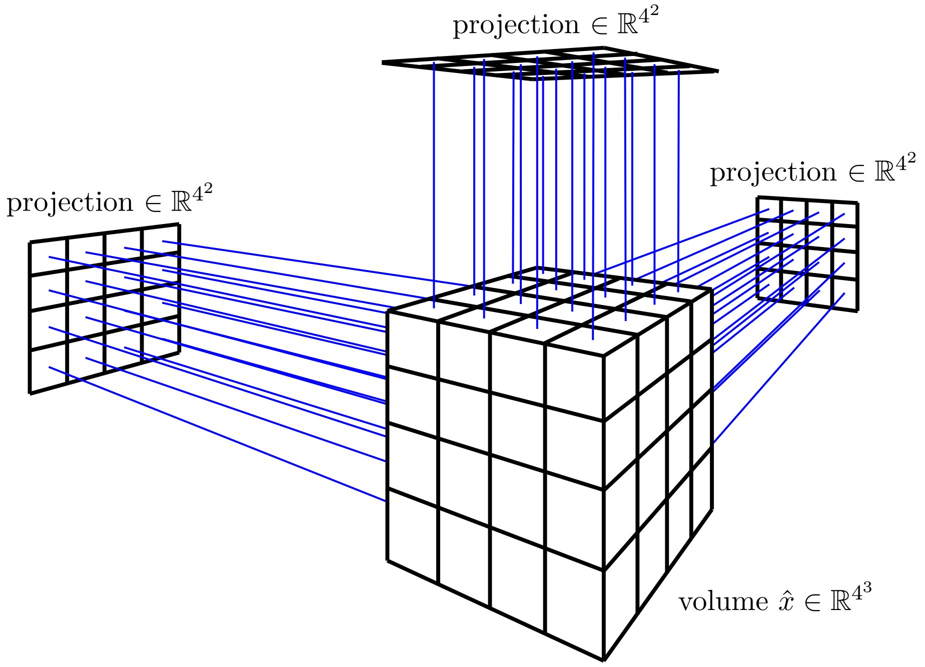

projection $\in \mathbb{R}^{4^2}$
projection $\in \mathbb{R}^{4^2}$
projection $\in \mathbb{R}^{4^2}$
volume $\hat{x} \in \mathbb{R}^{4^3}$

Figure 2.5 **-** Illustration of a $d \times d \times d$ volume measured by the sensor $A_d^D \in \{0,1\}^{Dd^{D-1} \times d^D}$ with $D = 3$ and $d = 4$ in (2.5.26). The projections are concatenated forming the observations $b \in \mathbb{R}^{Dd^{D-1}}$. $A_{i,j} = 1$ if the j-th ray is incident to the i-th voxel, otherwise $A_{i,j} = 0$. This results in observation b_j being the sum of voxels incident to the corresponding ray so that $A\hat{x} = b$.

2.5.4.2 Reduced Systems

A solution to a recovery problem known to be nonnegative in combination with a sensor matrix A without negative entries is a huge advantage. On the one hand it simplifies the investigation of recovery guarantees, and on the other hand it can help to reduce computational costs, significantly. It is natural when looking at the equation system $Ax = b$ with $A \in \mathbb{R}_+^{m \times n}$, $x \in \mathbb{R}_+^n$ and $b \in \mathbb{R}_+^m$ that, if there is a zero observation $b_i = 0$, all signal entries that lead as part to this observation must consequently be zero, as well. Formally, it is known a priori that

$$x_j = 0 \qquad \text{for all} \qquad j \in J := \{j \mid A_{i,j} > 0,\ i \in I\} \quad \text{with} \quad I := \{i \mid b_i = 0\}\ . \tag{2.5.28}$$

Let $I^c := [m] \setminus I$ and $J^c := [n] \setminus J$ be the set complements. The recovery is then executed for the remaining entries of the signal x by removing redundant rows and columns from the system. What remains is the *reduced system.*

Definition 2.45 *(reduced system)*:
The *reduced system* of the equation system $Ax = b$ with $A \in \mathbb{R}_+^{m \times n}$, $x \in \mathbb{R}_+^n$ and

$b \in \mathbb{R}_+^m$ is

$$A_{\text{red}} x_{\text{red}} = b_{\text{red}} \qquad \text{with} \qquad \begin{aligned} & A_{\text{red}} \in \mathbb{R}_+^{m_{\text{red}} \times n_{\text{red}}}, \quad x_{\text{red}} \in \mathbb{R}_+^{n_{\text{red}}}, \quad b_{\text{red}} \in \mathbb{R}_+^{m_{\text{red}}}, \\ & m_{\text{red}} = |I^c| = |\{i \mid b_i > 0\}| \quad \text{and} \\ & n_{\text{red}} = |J^c| = |\{j \mid A_{i,j} = 0,\ i \in I\}| \end{aligned}$$

where A_{red} is composed of the rows and columns of A which indexes are in I^c and J^c, respectively, ordered the same as in A. Correspondingly, this applies to b_{red}, too.

Obviously, the solution to the original system $Ax = b$ is given by $x_{J^c} = x_{\text{red}}$ while all other entries are zero, $x_J = 0$.

When observing sparse signals, the size of the reduced system is usually much smaller compared to the original one leading to only fractions of computational recovery costs. In addition, this a priori knowledge of zero signal entries helps analyzing the recovery guarantee. If the reduced system is overdetermined, i.e. $m_{\text{red}} \geq n_{\text{red}}$, and full rank, the original signal can be recovered perfectly. For adjacency matrices these condition can be guaranteed under mild assumptions [PS14].

2.5.4.3 Unbalanced Expander Graphs

Unfortunately, the tomographic sensors described in section 2.5.4.1 do not satisfy the usual CS conditions like the NSP (definition 2.32) and the RIP condition (definition 2.34) and standard results can not be applied [PS09, PS14]. For few tomographic projections and high sparsity that is the highly underdetermined case [PS14] provides provable guarantees. Therefore, sensor matrices implying orthogonal projections are interpreted as adjacency matrices of *unbalanced expander graphs* that are the topic of this section.

Definition 2.46 *((ν, δ)-unbalanced expander graph [FR13, Chapter 13])*:
A (ν, δ)-*unbalanced expander graph* is a bipartite simple graph $G = (L, R; E)$ having constant left degree ℓ such that for any $X \subseteq L$ with $|X| \leq \nu$, the set of neighbors of X denoted by $\mathcal{N}(X)$ has $\delta\ell|X|$ or more elements and $\mathcal{N}(X) \subseteq R$.

Thus, an adjacency matrix $A \in \{0,1\}^{m \times n}$ for an unbalanced expander graph is binary (as for any graph) and has ℓ nonzero entries in each column. Related to a recovery problem $Ax = b$ it is enough to just consider a subgraph depending on the given observations b. The corresponding right nodes of the subgraph are denoted by $R_b := \operatorname{supp}(b) \subseteq R$. Consequently, those left nodes which are neighbors of zero observations can be disregarded so that the left nodes of the subgraph are $L_b := L \setminus \mathcal{N}(R_b^c)$. The resulting subgraph perfectly corresponds to the matrix A_{red} of the reduced system which means $m_{\text{red}} = |R_b|$ and $n_{\text{red}} = |L_b|$.

For specific unbalanced expander graphs and sufficient sparse signals the following theorem originating from [WXT11] holds.

Theorem 2.47 *(solution set for given sparsity [PS14, Theorem 3.1.])*:
Let A be the adjacency matrix of a (ν, δ)-unbalanced expander graph with $1 \geq \delta > \frac{\sqrt{5}-1}{2} \approx 0.618$. Then for any s-sparse nonnegative vector x^* with $s \leq \frac{\nu}{1+\delta}$, the solution set $\{x \mid Ax = Ax^*, x \geq 0\}$ is a singleton.

Roughly speaking, this means that the diversity of the neighbors of the left node must be sufficiently high. The conditions on matrix A required by theorem 2.47 are generally met by tomographic sensors. Besides the sparsity, theorem 2.47 does not take the given observations into account, however, this is done by the next theorem.

Theorem 2.48 *(solution set for given observations [PS14, Theorem 3.5.])*:
Let A be the adjacency matrix of a (ν, δ)-unbalanced expander graph $G = (L, R; E)$ and let $X \subset L$ be a random subset of left nodes with $R_b = \mathcal{N}(X)$. If

$$|\mathcal{N}(Y)| \geq \delta\ell|Y| \qquad \text{with} \qquad \delta = \frac{\sqrt{5}-1}{2} \qquad \text{for any} \qquad Y \subset L_b \; ,$$

then the solution set $\{x \mid A_{\text{red}}x = A_{\text{red}}x^*, x \geq 0\}$ is a singleton for any $\frac{|L_b|}{1+\delta}$-sparse nonnegative vector x^*.

In particular, that means if $|X| \leq \frac{|L_b|}{1+\delta}$ for $X \subset L_b$ then a recovery supported on X is successful. Furthermore, theorem 2.48 guarantees that the adjacency matrix satisfying the condition $|\mathcal{N}(Y)| \geq \delta\ell|Y|$ has full rank. There was nothing said about general recovery on $\mathbb{R}^n$, yet. The following theorem is a combination of the two theorems 2.47 and 2.48 and does exactly that for the specific case $D = 3$.

Theorem 2.49 *(recovery with high probability for $D = 3$ [PS14, Proposition 5.9.])*:
Let $D = 3$ and let A be a sensor matrix of the form (2.5.26). If

$$s \leq \frac{N_L(s_\delta)}{1+\delta} = \frac{N_R(s_\delta)}{\ell} \qquad \text{where } s_\delta \text{ solves} \qquad N_R(s_\delta) = \ell\delta N_L(s_\delta)$$

then the system $Ax = b$ admits unique recovery of s-sparse nonnegative vectors x with high probability. Here, $\delta = \frac{\sqrt{5}-1}{2}$,

$$N_R(s) := Dd^{D-1}\left(1 - \left(1 - \frac{1}{d^{D-1}}\right)^s\right) \tag{2.5.29}$$

and

$$N_L(s) := d^3\left(1 - 3\left(1 - \frac{1}{d^2}\right)^s + 3\left(1 - \frac{2d-1}{d^3}\right)^s - \left(1 - \frac{3d-2}{d^3}\right)^s\right) \; . \tag{2.5.30}$$

The knowledge of the solution to a recovery problem being unique makes the search for a solution with special properties obsolete. It is not necessary any more to compute the sparsest solution via ℓ_0/ℓ_1-minimization. Instead it is sufficient to search for any solution by minimize the distance

$$\arg\min_x \{\|Ax - b\| \mid x \geq 0\} \tag{2.5.31}$$

in some norm or to solve $A_{\text{red}}x = b_{\text{red}}$ directly in the case of no involved errors.

2.5.4.4 Perturbed Expander

By *Perturbation* the full rank of a reduced system can be ensured and it helps to improve the results from the previous section, significantly.

Definition 2.50 *(perturbed matrix)*:
Let A be a matrix. A *perturbed* matrix $\tilde{A}$ of A is obtained by first generating a non-normalized perturbed sensor $\tilde{\tilde{A}}$ from A as

$$\tilde{\tilde{A}}_{i,j} := \begin{cases} A_{i,j} + \varepsilon_{i,j} & \text{if } A_{i,j} \neq 0 \\ A_{i,j} & \text{if } A_{i,j} = 0 \end{cases}$$

where $\varepsilon_{i,j} \in [-\varepsilon, \varepsilon]$ is uniformly random with a small $0 < \varepsilon < \min_{i,j} \{A_{i,j} \mid A_{i,j} > 0\}$. The perturbed matrix $\tilde{A}$ is gained by normalizing the columns of $\tilde{\tilde{A}}$ so that they sum up to the same constant (e.g. D).

That means a perturbed matrix $\tilde{A}$ has the same structure as A that is its sparsity and the constant left degree ℓ of the corresponding expander graph are preserved. A more strict rank definition plays a central role in this approach.

Definition 2.51 *(complete (Kruskal) rank)*:
The *complete (Kruskal) rank* $r_0 = r_0(A)$ of a matrix A is the maximum integer r_0 such that every subset of columns of A having size r_0 is linearly independent.

The complete rank of a perturbed sensor matrix and the corresponding recovery performance can be improved by investigating the connection to specific properties of the related expander graph. The basis is formed by the next lemma.

Lemma 2.52 *(existence of perturbed matrix and complete rank [PS14, Lemma 3.3.])*:
Let A be a nonnegative matrix with ℓ nonzero entries in each column. If for any submatrix formed by $\tilde{r}_0$ columns of A holds $|\mathcal{N}(X)| \geq |X|$ for each subset $X \subset L$ of columns with $|L| \leq \tilde{r}_0$, then there exists a perturbed matrix $\tilde{A}$ with the same structure as A so that its complete rank satisfies $r_0(\tilde{A}) \geq \tilde{r}_0$.

This can be adapted to the previously considered scenario.

Theorem 2.53 *(solution set for the perturbed case [PS14, Theorem 3.6.])*:
Let A be the adjacency matrix of a (ν, δ)-unbalanced expander graph $G = (L, R; E)$ and let $X \subset L$ be a random subset of left nodes with $R_b = \mathcal{N}(X)$. If

$$|\mathcal{N}(Y)| \geq |Y| \qquad \text{for any} \qquad Y \subset L_b \ ,$$

then there exists a perturbation $\tilde{A}_{\text{red}}$ of A_{red} such that the solution set

$\left\{x \;\middle|\; \tilde{A}_{\text{red}}x = \tilde{A}_{\text{red}}x^*, x \geq 0\right\}$ is a singleton for any $|L_b|$-sparse nonnegative vector x^*.

Theorem 2.53 is the equivalent of theorem 2.48 for the perturbed case. Again, the above results can be combined to a statement about the recovery in the specific case $D = 3$.

Theorem 2.54 *(recovery for the perturbed case, $D = 3$ [PS14, Proposition 5.10.])*: Let $D = 3$ and let A be a sensor matrix of the form (2.5.26). If

$$s \leq s_{\text{crit}} \qquad \text{where } s_{\text{crit}} \text{ solves} \qquad N_R(s_{\text{crit}}) = N_L(s_{\text{crit}})$$

where N_R and N_L are given by (2.5.29) and (2.5.30) then the perturbed system $\tilde{A}x = \tilde{b}$ admits unique recovery of s-sparse nonnegative vectors x with high probability.

Taking a look back at theorem 2.49, theorem 2.54 shows that perturbation improves the recovery performance of a system, significantly. Further results [PS14, Propositions 5.2. and 5.8.] prove that the dimensions of the reduced system concentrates on their expected values. Thus, the size of the reduced system does not deviate for given sparsity.

2.6 Motion Estimation by Optimal Transport

Motion estimation is a term of wide comprehension, typically used in connection with velocity estimation or object tracking. This section specifically concerns the displacement estimation of density distributions over a period of time. Such problems of mass transport at minimal costs belong to the research area called *optimal transport* [Vil03, Vil08]. The majority of approaches only considers two subsequent points in time and calculates a motion between those two. The same procedure can be repeated for each pair of subsequent time points yielding time dependent motion information.

The *optimal transport problem* was introduced by Monge in 1781 in terms of two measures supported on $\mathbb{R}^D$. The measure η_1 is supposed to be transported to η_2 by a mapping $T : \mathbb{R}^D \to \mathbb{R}^D$ called *transport plan* which is optimal in the sense that it solves

$$\inf_T \left\{ \int_{\mathbb{R}^D} c(x, T(x)) \, \mathrm{d}\eta_1(x) \;\middle|\; T\#\eta_1 = \eta_2 \right\} . \tag{2.6.1}$$

Here T is assumed to be η_1-measurable and $\#$ is the push-forward operator for measures. The $\eta_1 \otimes \eta_2$-measurable cost function $c : \mathbb{R}^D \times \mathbb{R}^D \to \mathbb{R}_+$ determines the cost of transporting mass from x to $T(x)$. Solving the *Monge problem* (2.6.1) over all 1-to-1 mappings as feasible set constitutes a highly nonlinear problem, that is difficult to analyze and to implement numerically. Therefore, one commonly resorts to a convex relaxation, introduced by Kantorovich [Kan58] and known as

Monge-Kantorovich problem, which reads

$$\inf_{\eta} \left\{ \int_{\mathbb{R}^D \times \mathbb{R}^D} c(x,y) \, \mathrm{d}\eta(x,y) \;\middle|\; \eta \in \mathcal{M}(\eta_1, \eta_2) \right\} \tag{2.6.2}$$

where $\mathcal{M}$ is the set of probability measures on $\mathbb{R}^D \times \mathbb{R}^D$ with marginal distributions η_1 and η_2, that is

$$\begin{aligned} &\int_{\mathbb{R}^D} \eta(x,y) \mathrm{d}y = \eta_1(x) \text{ for (almost) all } x \in \mathbb{R}^D \text{ and} \\ &\int_{\mathbb{R}^D} \eta(x,y) \mathrm{d}x = \eta_2(y) \text{ for (almost) all } y \in \mathbb{R}^D \ . \end{aligned} \tag{2.6.3}$$

The following sections present different approaches for solving either problem (2.6.1) or (2.6.2), numerically.

2.6.1 Discrete Optimal Transport

Discrete optimal transport concerns the optimal transport of discrete measures of the form

$$\eta_1 = \sum_{i=1}^{n} \delta_{x_i} \qquad \text{and} \qquad \eta_2 = \sum_{j=1}^{n} \delta_{y_j} \ ,$$

where δ_x is the Dirac measure located at $x \in \mathbb{R}^D$. That means the support of the measures η_1 and η_2 are composed of n points $x_i \in \mathbb{R}^d$ for $i \in [n]$ and $y_j \in \mathbb{R}^D$ for $j \in [n]$, respectively, which all have the same unit mass 1.

Of particular interest is the scenario where the points are located on a D-dimensional regular grid with grid positions indexed by $x_i, y_j \in \mathbb{R}^D$. With the help of a permutation $p \in \mathcal{S}_n$, both the continuous coupling measure η and transport mapping T can be represented by a permutation matrix $P \in \mathcal{P}_n$ (section 2.3) with

$$P_{i,j} = \begin{cases} 1 & \text{if} \quad j = p(i) \\ 0 & \text{otherwise} \end{cases}$$

and $P(x_i) = y_{p(i)}$. This leads to an *1-to-1 assignment* between grid points which has to be determined.

The cost function c is realized by a cost matrix $C_n \in \mathbb{R}_+^{n \times n}$ with $(C_n)_{i,j} = c(x_i, y_j)$ for all $i, j \in [n]$. Then, the permutation matrix P_n is the solution of the optimization problem

$$\min_{P \in \mathcal{P}_n} \operatorname{tr}\left(C_n^\top P\right) = \sum_{i=1}^{n} \sum_{j=1}^{n} c(x_i, y_j) P_{i,j} \ . \tag{2.6.4}$$

This is a non-convex NP-hard problem, but the binary constraints $P \in \{0,1\}^{n \times n}$ of the feasible set $\mathcal{P}$ can be relaxed to the *Birkhoff polytope* of doubly stochastic matrices

$$\mathcal{B}_n := \left\{ P \in \mathbb{R}_+^{n\times n} \colon P\mathbb{1} = \mathbb{1}, P^\top \mathbb{1} = \mathbb{1} \right\} \ ,$$

which is also known as *assignment polytope*. The well-known Birkhoff-von-Neumann theorem [KV08, Corollary 11.3] states that the assignment polytope is the convex hull of all assignments and thus the vertices of the assignment polytope correspond uniquely to permutation matrices. Consequently, the solution of the linear program

$$\min_{P\in\mathcal{B}_n} \operatorname{tr}\left(C_n^\top P \right) \tag{2.6.5}$$

is binary and a solution to the discrete problem (2.6.4). In a broader sense (2.6.5) is a discrete version of (2.6.2) enabling the determination of the transport mapping T by means of the permutation matrix P. Further details can be found in [Vil03, Vil08] and [FPPA14].

2.6.2 The fluid mechanics framework of Benamou and Brenier

Benamou and Brenier [BB00] study the Monge-Kantorovich problem (2.6.2) together with further constraints in order to connect optimal transport and physically valid flows. First, the variational approach is sketched followed by studying algorithmic realizations using convex analysis.

2.6.2.1 Variational Approach

The available data is given in the form of intensity fields, mass fields or, respectively, images $U^0(x) \geq 0$ and $U^T(x) \geq 0$ with $U^0, U^T : \mathbb{R}^D \to \mathbb{R}_0^+$ for the first and second point in time 0 and T. A first assumption is that a *mass preserving mapping* $M : \mathbb{R}^D \to \mathbb{R}^D$ transports the mass from U^0 to U^T, i.e.

$$\int_{M(x)\in S} U^0(x)\,\mathrm{d}x = \int_{x\in S} U^T(x)\,\mathrm{d}x$$

for all bounded subsets $S \subseteq \mathbb{R}^D$. This constraint is analogous to the constraint (2.6.3) of the Monge-Kantorovich problem (2.6.2) where η_1 corresponds to U^0 and η_2 to U^T. If such a mass preserving mapping M minimizes the so called *Wasserstein distance*

$$W(U^0, U^T) := \sqrt{\inf_M \int_{\mathbb{R}^D} \|M(x) - x\|_2^2 U^0(x)\,\mathrm{d}x} \tag{2.6.6}$$

it realizes an optimal transport plan from U^0 to U^T and forms a solution to the Monge-Kantorovich problem in the sense that $W^2(U^0, U^T)$ is equal to the infimum (2.6.2).

Benamou and Brenier studied time-variant realizations of the mass preserving mapping $M(x)$ in terms of a velocity field $v(t, x) : \mathbb{R}^D \to \mathbb{R}^D$, $0 \leq t \leq T$ that transports the underlying domain in a physically plausible way. Specifically, they considered the continuity equation

$$\partial_t u + \nabla_x \cdot (uv) = 0 \tag{2.6.7}$$

with the boundary conditions

$$u(0,x) = U^0(x) \quad \text{and} \quad u(T,x) = U^T(x) \quad \forall\, x \in \mathbb{R}^D\ . \tag{2.6.8}$$

This introduces an additional time dependency of the density function $u(t,x) \geq 0$ with $0 \leq t \leq T$.

The most important result proven by Benamou and Brenier is the equation

$$\Big(W(U^0,U^T)\Big)^2 = T\ \inf_{u,v} \left\{ \int_{\mathbb{R}^D} \int_0^T u(t,x) \|v(t,x)\|_2^2 \,\mathrm{d}t\,\mathrm{d}x \ \middle|\ (2.6.7) \text{ and } (2.6.8) \text{ hold} \right\} \tag{2.6.9}$$

converting the problem from the evaluation of the Wasserstein distance (2.6.6) into a *continuum mechanics formulation* with physical constraints and boundary conditions [BB00, Proposition 1.1]. It is the basis of the next section.

2.6.2.2 Dynamic Optimal Transport by Convex Programming

The computational domain is chosen to be periodic, that is $X := \mathbb{R}^D/\mathbb{Z}^D$. In order to obtain a formulation suitable for the application of algorithms, Benamou and Brenier introduce the momentum $w(t,x) := u(t,x)v(t,x)$ and reformulate (2.6.9) as a (generalized[2]) saddle-point problem (2.4.1) with Lagrangian multipliers $\phi(t,x)$, that is

$$\inf_{u,v} \sup_{\phi}\ L(u,v,\phi) \tag{2.6.10}$$

with Lagrangian

$$L(u,v,\phi) := \int_X \int_0^T \frac{1}{2} u\|v\|^2 \,\mathrm{d}t\,\mathrm{d}x + \int_X \int_0^T \phi\left(\partial_t u + \nabla_x \cdot (uv)\right) \mathrm{d}t\,\mathrm{d}x \tag{2.6.11}$$

where the factor T is replaced by $\frac{1}{2}$ for later convenience.

With the help of *Green's first identity* for two scalar functions f and g on $\mathbb{R}^D$

$$\int_X \nabla g \cdot \nabla f + g\nabla^2 f \,\mathrm{d}x = \oint_{\partial X} g(\nabla f \cdot \overrightarrow{n}) \,\mathrm{d}s \tag{2.6.12}$$

[2] The initial problem is formulated in a continuous space, so that an extension to the ordinary saddle point problem as described in chapter 2.4 is necessary, cf. [Roc97, Section 29]. Since the problem is discretized later on anyway and the principle of Lagrangian functions and multipliers is the same, the introduction to generalized saddle point problems is omitted.

and the *Hamilton-Jacobi equation*

$$\begin{aligned} &\partial_t \phi + \frac{\|\nabla \phi\|^2}{2} = 0 \;\text{ with }\; v = \nabla_x \phi \\ \Leftrightarrow\quad &\partial_t \phi + \frac{\|w\|^2}{2u} = 0 \;\text{ with }\; w = u \nabla_x \phi \end{aligned} \tag{2.6.13}$$

which is an optimality condition following from (2.6.9), the Lagrangian (2.6.11) can be reformulated. By defining $\nabla g := v$ and $\nabla f := w$ and the assumption of homogeneous Neumann boundary conditions in space, i.e. $\nabla_x \phi \cdot \overrightarrow{n} = 0$, insertion of (2.6.13) into (2.6.12) yields

$$\begin{aligned} \int_X (\nabla_x \phi) \cdot w + \phi (\nabla_x \cdot w) \,\mathrm{d}x &= \oint_{\partial X} \phi (u \underbrace{\nabla_x \phi \cdot \overrightarrow{n}}_{=0}) \,\mathrm{d}s \\ \Leftrightarrow \int_X \phi (\nabla_x \cdot w) \,\mathrm{d}x &= - \int_X (\nabla_x \phi) \cdot w \,\mathrm{d}x \;. \end{aligned} \tag{2.6.14}$$

Thus, the Lagrangian (2.6.11) can be expressed as

$$\begin{aligned} L(u, v, \phi) &= \int_X \int_0^T \frac{\|w\|^2}{2u} \,\mathrm{d}t \,\mathrm{d}x + \int_X \int_0^T \phi \,(\nabla_x \cdot w) \,\mathrm{d}t \,\mathrm{d}x + \int_X \int_0^T \phi \,(\partial_t u) \,\mathrm{d}t \,\mathrm{d}x \\ &\overset{(2.6.14)}{=} \int_X \int_0^T \frac{\|w\|^2}{2u} \,\mathrm{d}t \,\mathrm{d}x - \int_X \int_0^T (\nabla_x \phi) \cdot w \,\mathrm{d}t \,\mathrm{d}x + \int_X \int_0^T \phi \,(\partial_t u) \,\mathrm{d}t \,\mathrm{d}x \\ &= \int_X \int_0^T \frac{\|w\|^2}{2u} - (\nabla_x \phi) \cdot w \,\mathrm{d}t \,\mathrm{d}x + \int_X \left([\phi u]_0^T - \int_0^T (\partial_t \phi) \, u \,\mathrm{d}t \right) \mathrm{d}x \\ &= \int_X \int_0^T \frac{\|w\|^2}{2u} - (\partial_t \phi) u - (\nabla_x \phi) \cdot w \,\mathrm{d}t \,\mathrm{d}x - \int_X \phi(0, x) U^0 - \phi(T, x) U^T \,\mathrm{d}x \end{aligned}$$

and it can be further transformed as follows in order to apply convex programming.

Consider the indicator function

$$\delta_K(x) = \begin{cases} 0 & \text{if } x \in K \\ \infty & \text{if } x \notin K \end{cases}$$

of the convex set

$$K := \left\{ \begin{bmatrix} \alpha(t, x) \\ \beta(t, x) \end{bmatrix} : \mathbb{R} \times \mathbb{R}^D \to \mathbb{R} \times \mathbb{R}^D \;\middle|\; \alpha + \frac{\|\beta\|^2}{2} \leq 0 \quad \forall \, t, x \right\} \;, \tag{2.6.15}$$

and its Legendre-Fenchel conjugate (definition 2.27)

$$\delta_K^* \left(\begin{bmatrix} u \\ w \end{bmatrix} \right) = \sup_{\alpha \in \mathbb{R}, \beta \in \mathbb{R}^D} \left\{ \begin{bmatrix} \alpha \\ \beta \end{bmatrix} \cdot \begin{bmatrix} u \\ w \end{bmatrix} - \delta_K \left(\begin{bmatrix} \alpha \\ \beta \end{bmatrix} \right) \right\} \;. \tag{2.6.16}$$

Rewriting this in Lagrangian form using the inequality that defines K by (2.6.15), yields

$$\delta_K^*\left(\begin{bmatrix} u \\ w \end{bmatrix}\right) = \inf_{\alpha,\beta} \sup_{\tau\geq 0} \left\{-\alpha u - \beta \cdot w + \tau \left(\alpha + \frac{\|\beta\|^2}{2}\right)\right\}$$

with Lagrange multiplier $\tau \geq 0$. This infimum can be obtained directly by setting the gradient to zero:

$$\begin{aligned} \partial_\alpha : &\qquad -u + \tau = 0 \Leftrightarrow \tau = u \\ \nabla_\beta : &\qquad -w + \tau\beta = 0 \Leftrightarrow \beta = \frac{w}{u} \\ \partial_\tau : &\qquad \alpha + \frac{\|\beta\|^2}{2} = 0 \Leftrightarrow \alpha = -\frac{\|w\|^2}{2u^2} \ . \end{aligned} \tag{2.6.17}$$

Inserting the optimal values $\bar{\alpha}$ and $\bar{\beta}$ into (2.6.16) yields

$$\begin{aligned} &\sup_{\alpha,\beta}\left\{\begin{bmatrix} \alpha \\ \beta \end{bmatrix} \cdot \begin{bmatrix} u \\ w \end{bmatrix} - \delta_K\left(\begin{bmatrix} \alpha \\ \beta \end{bmatrix}\right)\right\} = \begin{bmatrix} \bar{\alpha} \\ \bar{\beta} \end{bmatrix} \cdot \begin{bmatrix} u \\ w \end{bmatrix} \\ = \bar{\alpha}u + \bar{\beta}\cdot w &\overset{(2.6.17)}{=} -\frac{\|w\|^2}{2u^2}u + \frac{w}{u}\cdot w = -\frac{\|w\|^2}{2u} + \frac{\|w\|^2}{u} = \frac{\|w\|^2}{2u} \ . \end{aligned} \tag{2.6.18}$$

Now, using the compound variables $\mu := \begin{bmatrix} u \\ w \end{bmatrix}$ and $q := \begin{bmatrix} \alpha \\ \beta \end{bmatrix}$ and the function

$$F_1(\phi) := \int_X \phi(0,x)U^0 - \phi(T,x)U^T \,\mathrm{d}x \ ,$$

the saddle-point problem (2.6.10) can be reformulated with the help of identity (2.6.18) to obtain

$$\begin{aligned} -\inf_{u,w}\sup_{\phi} \ L(u,w,\phi) \ &= \ \sup_{u,w}\inf_{\phi} \ -L(u,w,\phi) \\ = &\sup_{\mu}\inf_{\phi} \int_0^T \int_X \nabla_{t,x}\phi \cdot \mu - \sup_{q\in K}\{\mu \cdot q\} \,\mathrm{d}x\,\mathrm{d}t + F_1(\phi) \\ = &\sup_{\mu}\inf_{\phi,q} \ \delta_K(q) + \int_0^T \int_X \nabla_{t,x}\phi \cdot \mu - \mu \cdot q \,\mathrm{d}x\,\mathrm{d}t + F_1(\phi) \\ = &\sup_{\mu}\inf_{\phi,q} \ \delta_K(q) + F_1(\phi) + \int_0^T \int_X \mu \cdot (\nabla_{t,x}\phi - q)\,\mathrm{d}x\,\mathrm{d}t \ . \end{aligned} \tag{2.6.19}$$

Here μ can be considered as a Lagrange multiplier of the new constraint $\nabla_{t,x}\phi - q = 0$. This saddle-point problem has to be solved numerically and, for this purpose, provides a more convenient problem formulation than the original continuous formulation. An established method for solving this saddle-point problem is to utilize the ADMM algorithm 2.2 together with the augmented Lagrangian

$$\begin{aligned} L_\rho(\phi, q, \mu) :=&\delta_K(q) + F_1(\phi) + \int_0^T \int_X \mu \cdot (\nabla_{t,x}\phi - q)\,\mathrm{d}x\,\mathrm{d}t \\ &+ \frac{\rho}{2}\int_0^T \int_X (\nabla_{t,x}\phi - q)\cdot(\nabla_{t,x}\phi - q)\,\mathrm{d}x\,\mathrm{d}t \end{aligned} \tag{2.6.20}$$

with penalty parameter $\rho(t, x) > 0$ for every $t \in [0, T]$ and every $x \in D$. Assuming all variables are discretized in space and time on a regular grid with n vertices, this leads to the following algorithm.

Algorithm 2.7. **-** ADMM for the Benamou and Brenier formulation

```
   Fix $q^0 \in \mathbb{R}^{n(D+1)}$, $\mu^0 \in \mathbb{R}^{n(D+1)}$, $\rho > 0$
1: for $k = 0, 1, \dots$ do
2:     $\phi^{k+1} = \arg\min_{\phi\in\mathbb{R}^n} L_\rho(\phi, q^k, \mu^k)$
3:     $q^{k+1} = \arg\min_{q\in\mathbb{R}^{n(D+1)}} L_\rho(\phi^{k+1}, q, \mu^k)$
4:     $\mu^{k+1} = \mu^k + \rho\left(\nabla\phi^{n+1} - q^{n+1}\right)$
5: end for
```

In the following two sections, solutions to the subproblems of line 2 and line 3 are elaborated.

2.6.2.3 First ADMM step

In this section a partial differential equation is derived that solutions $\phi(t, x)$ of line 2 of algorithm 2.7 has to satisfy. The derivation starts with computing the first variation of the functional L_ρ with respect to ϕ as necessary optimality condition. Let φ be a function with $\varphi : \mathbb{R} \times \mathbb{R}^D \to \mathbb{R}$ and $h > 0$. The Gâteaux derivative of (2.6.20) is

$$\begin{aligned} l(h) &:= L_\rho(\phi + h\varphi) \\ &= \delta_K(q) + F_1(\phi + h\varphi) + \int_0^T \int_X \mu \cdot (\nabla_{t,x}(\phi + h\varphi) - q)\,\mathrm{d}x\,\mathrm{d}t \\ &\quad + \frac{\rho}{2}\int_0^T \int_X (\nabla_{t,x}(\phi + h\varphi) - q)\cdot(\phi + h\varphi) - q)\,\mathrm{d}x\,\mathrm{d}t \\ &= \delta_K(q) + F_1(\phi) + hF_1(\varphi) + \int_0^T \int_X \mu \cdot (\nabla_{t,x}\phi + h\nabla_{t,x}\varphi - q)\,\mathrm{d}x\,\mathrm{d}t \\ &\quad + \frac{\rho}{2}\int_0^T \int_X (\nabla_{t,x}\phi + h\nabla_{t,x}\varphi - q)\cdot(\nabla_{t,x}\phi + h\nabla_{t,x}\varphi - q)\,\mathrm{d}x\,\mathrm{d}t \end{aligned}$$

leading to

$$l'(h) = \; F_1(\varphi) + \int_0^T \int_X \mu \cdot \nabla_{t,x}\varphi\,\mathrm{d}x\,\mathrm{d}t + \rho\int_0^T \int_X (\nabla_{t,x}\phi + h\nabla_{t,x}\varphi - q)\cdot\nabla_{t,x}\varphi\,\mathrm{d}x\,\mathrm{d}t$$

and searching for the roots of $l'(0)$ gives

$$l'(0) = F_1(\varphi) + \int_0^T \int_X (\mu + \rho \nabla_{t,x}\phi - \rho q) \cdot \nabla_{t,x}\varphi \, \mathrm{d}x \, \mathrm{d}t = 0 \ .$$

Now, splitting the compound variables $\mu(x,t) = \begin{bmatrix} u(x,t) \in \mathbb{R} \\ w(x,t) \in \mathbb{R}^D \end{bmatrix}$ and $q(x,t) = \begin{bmatrix} \alpha(x,t) \in \mathbb{R} \\ \beta(x,t) \in \mathbb{R}^D \end{bmatrix}$ into their time and space components, again, yields by partial integration

$$\begin{aligned} F_1(\varphi) + \int_X \int_0^T (u + \rho \partial_t \phi - \rho\alpha)\ \partial_t \varphi \, \mathrm{d}t \, \mathrm{d}x \\ + \int_0^T \int_X (w + \rho \nabla_x \phi - \rho\beta) \cdot \nabla_x \varphi \, \mathrm{d}x \, \mathrm{d}t = 0 \end{aligned}$$

$$\begin{aligned} \Leftrightarrow F_1(\varphi) \underbrace{+ \int_X [(u + \rho\partial_t\phi - \rho\alpha)\varphi]_0^T \ \mathrm{d}x}_{=:\, F_2(\phi,\varphi)} \underbrace{- \int_X \int_0^T \partial_t(u + \rho\partial_t\phi - \rho\alpha)\varphi \, \mathrm{d}t \, \mathrm{d}x}_{=:\, F_3(\phi,\varphi)} \\ \underbrace{+ \int_0^T [(w + \rho\nabla_x\phi - \rho\beta)\varphi]_X \ \mathrm{d}t}_{=:\, F_4(\phi,\varphi)} \underbrace{- \int_0^T \int_X \nabla_x \cdot (w + \rho\nabla_x\phi - \rho\beta)\varphi \, \mathrm{d}x \, \mathrm{d}t}_{=:\, F_5(\phi,\varphi)} = 0. \end{aligned}$$

This holds for arbitrary φ if $F_1 + F_2 = 0$, $F_3 + F_5 = 0$ and $F_4 = 0$. Reformulating these conditions gives

$$\begin{aligned} F_1 + F_2 =& 0 \\ \Rightarrow \quad U^T\varphi(T,x) - U^0\varphi(0,x) =& [(u + \rho\partial_t\phi - \rho\alpha)\varphi]_0^T \qquad \forall\, x \in D \\ \Leftrightarrow \quad U^T\varphi(T,x) - U^0\varphi(0,x) =& (u(T,x) + \rho\partial_t\phi(T,x) - \rho\alpha(T,x))\varphi(T,x) \\ & - (u(0,x) + \rho\partial_t\phi(0,x) - \rho\alpha(0,x))\varphi(0,x) \\ \Rightarrow \quad u_t\varphi(t,x) =& (u(t,x) + \rho\partial_t\phi(t,x) - \rho\alpha(t,x))\varphi(t,x) \quad \text{for } t \in \{0,T\} \\ \Rightarrow \quad \rho\partial_t\phi(t,x) =& U^t - u(t,x) + \rho\alpha(t,x) \qquad \forall\, t \in \{0,T\}, x \in D\ , \end{aligned} \tag{2.6.21}$$

$$\begin{aligned} F_3 + F_5 =& 0 \\ \Rightarrow (-\nabla_x \cdot w - \rho\Delta_x\phi + \rho\nabla_x \cdot \beta)\varphi =& (\partial_t u + \rho\partial_t^2\phi - \rho\partial_t\alpha)\varphi \qquad \forall\, t \in \,]0,T[\,, x \in X \\ \Rightarrow -\rho\partial_t^2\phi - \rho\Delta_x\phi =& \partial_t u + \nabla_x \cdot w - \rho\partial_t\alpha - \rho\nabla_x \cdot \beta \\ \Leftrightarrow -\rho\Delta_{t,x}\phi =& \nabla_{t,x} \cdot (\mu - \rho q) \qquad \forall\, t \in \,]0,T[\,, x \in X \end{aligned} \tag{2.6.22}$$

and

$$\begin{aligned}
&& F_4 =&0\\
&\Rightarrow & 0 =& [(w + \rho\nabla_x\phi - \rho\beta)\varphi]_X \qquad \forall\, t \in\,]0, T[\\
&\Leftrightarrow & 0 =& \sum_{i=1}^{D} [(w_i + \rho\partial_{x_i}\phi - \rho\beta_i)\varphi]_0^{X_i}\\
&\Rightarrow & 0 =&[(w_i + \rho\partial_{x_i}\phi - \rho\beta_i)\varphi]_0^{X_i} \qquad \text{for } i \in [D]\\
&\Leftrightarrow & 0 =&(w_i(t, X_i) + \rho\partial_{x_i}\phi(t, X_i) - \rho\beta_i(t, X_i))\varphi(t, X_i)\\
&&& - (w_i(t, 0) + \rho\partial_{x_i}\phi(t, 0) - \rho\beta_i(t, 0))\varphi(t, 0)\\
&\Rightarrow & 0 =&(w_i(t, x) + \rho\partial_{x_i}\phi(t, x) - \rho\beta_i(t, x))\varphi(t, x) \qquad \text{for } x \in \{0, X_i\}\\
\Rightarrow\quad & -\rho\partial_{x_i}\phi(t, x) =&&w_i(t, x) - \rho\beta_i(t, x)\\
\Leftrightarrow\quad & -\rho\partial_{x_i}^2\phi(t, x) =&&\partial_{x_i}(w_i(t, x) - \rho\beta_i(t, x)) \qquad \forall\, t \in\,]0, T[\,, x \in \{0, X_i\}, i \in [D]
\end{aligned} \tag{2.6.23}$$

where the last equivalence after applying the ∂_{x_i} operator on both sides of the equation holds up to constants.

Equation (2.6.22) is a partial differential equation for ϕ, given the quantities q, μ and ρ. Its Neumann time boundary conditions (2.6.21) and those regarding the space boundary (2.6.23) are the natural boundary conditions when extending the differential equation to the space boundary. In summary the differential equation system which has to be solved in order to obtain ϕ^{n+1} for the first ADMM step of algorithm 2.7 is

$$\begin{aligned}
-\rho\Delta_{t,x}\phi &= \nabla_{t,x}\cdot(\mu - \rho q) && \forall\, t \in\,]0, T[\,, x \in X\\
-\rho\partial_{x_i}^2\phi &= \partial_{x_i}(w_i - \rho\beta_i) && \forall\, t \in\,]0, T[\,, x \in \{0, X_i\}, i \in [D]\\
\rho\partial_t\phi &= U^t - u + \rho\alpha && \forall\, t \in \{0, T\}, x \in X\ .
\end{aligned}$$

The system is well posed since the mass U^0 and U^T are equal at both time points and it can be solved numerically by approximating the differential operators by finite differences.

2.6.2.4 Second ADMM step

The goal is to minimize (2.6.20) with respect to $q(t,x)$ whereas $\phi(t,x)$ and $\mu(t,x)$ are fixed. Thus, (2.6.20) can be reformulated as

$$\begin{aligned}
q = \operatorname*{arg\,min}_{q\in\mathbb{R}^{D+1}} &\delta_K(q) + \int_0^T \int_X \mu\cdot(\nabla_{t,x}\phi - q)\,\mathrm{d}x\,\mathrm{d}t \\
&\qquad + \frac{\rho}{2}\int_0^T \int_X (\nabla_{t,x}\phi - q)\cdot(\nabla_{t,x}\phi - q)\,\mathrm{d}x\,\mathrm{d}t \\
= \operatorname*{arg\,min}_{q\in K} &\int_0^T \int_X \mu\cdot(\nabla_{t,x}\phi - q)\,\mathrm{d}x\,\mathrm{d}t + \frac{\rho}{2}\int_0^T \int_X (\nabla_{t,x}\phi - q)\cdot(\nabla_{t,x}\phi - q)\,\mathrm{d}x\,\mathrm{d}t \\
= \operatorname*{arg\,min}_{q\in K} &\frac{1}{2\rho}\int_0^T \int_X \mu\cdot\mu\,\mathrm{d}x\,\mathrm{d}t + \int_0^T \int_X \mu\cdot(\nabla_{t,x}\phi - q)\,\mathrm{d}x\,\mathrm{d}t \\
&\qquad + \frac{\rho}{2}\int_0^T \int_X (\nabla_{t,x}\phi - q)\cdot(\nabla_{t,x}\phi - q)\,\mathrm{d}x\,\mathrm{d}t \\
= \operatorname*{arg\,min}_{q\in K} &\int_0^T \int_X \frac{\mu}{\rho}\cdot\frac{\mu}{\rho} + 2\,\frac{\mu}{\rho}\cdot(\nabla_{t,x}\phi - q) + (\nabla_{t,x}\phi - q)\cdot(\nabla_{t,x}\phi - q)\,\mathrm{d}x\,\mathrm{d}t \\
= \operatorname*{arg\,min}_{q\in K} &\int_0^T \int_X \left(\frac{\mu}{\rho} + \nabla_{t,x}\phi - q\right)\cdot\left(\frac{\mu}{\rho} + \nabla_{t,x}\phi - q\right)\,\mathrm{d}x\,\mathrm{d}t\ .
\end{aligned} \tag{2.6.24}$$

with set K defined as in (2.6.15). Since $q(x,t) = \begin{bmatrix}\alpha(x,t)\in\mathbb{R}\\ \beta(x,t)\in\mathbb{R}^D\end{bmatrix}$ is a vector function dependent on t and x the minimization problem (2.6.24) can be solved pointwise, i.e. separately for every $t\in[0,T]$ and every $x\in X$. By splitting the compound variables $\mu(x,t) = \begin{bmatrix}u(x,t)\in\mathbb{R}\\ w(x,t)\in\mathbb{R}^D\end{bmatrix}$ into their time and space components, as well, this yields the least square problems

$$\min_{\alpha\in\mathbb{R},\beta\in\mathbb{R}^D}\left\{\left(\frac{u}{\rho}+\partial_t\phi-\alpha\right)^2+\left\|\frac{w}{\rho}+\nabla_x\phi-\beta\right\|_2^2\ \middle|\ \alpha+\frac{\|\beta\|_2^2}{2}\le 0\right\}\ \forall\, t\in[0,T], x\in X$$

which solutions constitute q^{n+1} in the second ADMM step of algorithm 2.7.

3 Joint Tomographic Reconstruction and Motion Recovery Using Dynamic Optimal Transport

Consider an optimization problem which can be reformulated such that a solution can be found by solving two smaller optimization problems where the second one depends on the result of the first. A straightforward approach is to solve the two problems consecutively such that the solution to the first problem defines the second one. Depending on the accuracy of the first solution, however, the computed solution to the overall problem might be poor, since the second problem is affected by errors of the first one. In order to circumvent this error source a better approach is to optimize both problems jointly.

This chapter describes an attempt to tackle the joint problem by combining tomographic reconstruction and dynamic optimal transport. More precisely, recovery of nonnegative signals (section 2.5.4) is combined with the fluid mechanics framework of Benamou and Brenier (section 2.6.2). These two concepts are chosen to replace the inaccurate and empirically derived reconstruction and motion estimation routines that are executed subsequently in Tomo-PIV (chapter 1). Three different approaches for jointly solving them are presented and evaluated: weak coupling (section 2.4.3.2), scaled ADMM (section 2.4.3.1) and the parallel proximal algorithm (section 2.4.3.4).

The weak coupling and scaled ADMM approaches are quite similar and both consists of two nested optimization loops where the inner one is identical in terms of structure. In contrast to that the PPXA approach just contains one single iteration. Section 3.1 gives details about the discretization used and introduces additional required notation whereas section 3.2 states the reconstruction and transport problem supposed to be combined. The three subsequent sections 3.3, 3.4 and 3.5 are devoted to problem splitting methods from convex programming and derive the aforementioned approaches with corresponding algorithms. This is followed by experiments in section 3.6 and a conclusion of this chapter in section 3.7.

3.1 Discretization and Notation

Let $U^0 \in \mathbb{R}^n$ and $U^T \in \mathbb{R}^n$ be two D-dimensional images rearranged as vectors. Both are unknown a priori and build the ground truth of the joint problem. The two are the connecting parts between the recovery of nonnegative signals (section 2.5.4) and the fluid mechanics framework of Benamou and Brenier (section 2.6.2). The task of the former is the recovery of the two images and the latter aims for the estimation of motion in between. In order to apply the images as time boundary conditions for the motion estimation, the fluid mechanics framework has to be discretized in space and time first.

Therefore, a grid is introduced with X_i points in the i-th dimension in space and X_0 points in time such that the total number of grid points is $N := X_0 n$ with $n := \prod_{i=1}^{D} X_i$. Hereafter, single grid values are addressed either using an index $1, \ldots, N$ whenever the ordering of grid nodes does not matter, or by a D-tupel $(i_1, \ldots, i_D)$ or $(1+D)$-tupel $(i_0, i_1, \ldots, i_D)$ where $i_j \in [X_j]$ for every $j \in \{0, \ldots, D\}$ if a distinction between different dimensions is necessary. Vector entries corresponding to a grid node can be distinguished by an additional index placed in front of the grid index or the tupel, e.g. $\mu_{k,i}$ or $\mu_{k,(i_0,\ldots,i_D)}$, respectively.

Subsequent sections contain the derivation of the joint problem and will use the following notation, frequently.

- X_j as the number of grid discretization steps of the j-th dimension for $j \in \{0, \ldots, D\}$ where X_0 corresponds to the time dimension
- $U^0 \in \mathbb{R}^n$ and $U^T \in \mathbb{R}^n$ as time boundary images
- $\mu = \begin{bmatrix} \mu_1 \\ \vdots \\ \mu_N \end{bmatrix} \in \mathbb{R}^{N(1+D)}$ with $\mu_i = \begin{bmatrix} u_i \\ w_i \end{bmatrix} \in \mathbb{R}^{1+D}$ and $u_i \in \mathbb{R}_+$ being a mass or pixel brightness value at grid position i and the corresponding momentum $w_i \in \mathbb{R}^D$ as primal variables
- $\bar{u} := \begin{bmatrix} u_{(1,\cdot,\ldots,\cdot)} \\ u_{(X_0,\cdot,\ldots,\cdot)} \end{bmatrix} = \begin{bmatrix} \mu_{0,(1,\cdot,\ldots,\cdot)} \\ \mu_{0,(X_0,\cdot,\ldots,\cdot)} \end{bmatrix} \in \mathbb{R}^{2n}$ as simplified notation for the entries of μ corresponding to the images of the first and last point in time, that are the estimates for U^0 and U^T.
- $q = \begin{bmatrix} q_1 \\ \vdots \\ q_N \end{bmatrix} \in \mathbb{R}^{N(1+D)}$ with $q_i = \begin{bmatrix} \alpha_i \\ \beta_i \end{bmatrix} \in \mathbb{R}^{1+D}$, $\alpha_i \in \mathbb{R}$ and $\beta_i \in \mathbb{R}^D$ at grid position i as dual variables
- $\phi \in \mathbb{R}^N$ as former Lagrangian multipliers
- $\bar{b} := \begin{bmatrix} b_1 \\ b_{X_0} \end{bmatrix} \in \mathbb{R}^{2m}$ with $b_1, b_{X_0} \in \mathbb{R}^m$ being the given m observations at the first and last point in time
- $\bar{A} = \begin{bmatrix} A & 0 \\ 0 & A \end{bmatrix} \in \mathbb{R}^{2m \times 2n}$ with the sensor matrix $A \in \mathbb{R}^{m \times n}$ used to obtain the observations

3.2 The Two Subproblems

This section recalls the basic problems supposed to be combined in order to solve tomographic reconstruction and dynamic optimal transport jointly.

Basically, for recovering U^0 and U^T the underdetermined linear system

$$\bar{A}\bar{u} = \bar{b} \tag{3.2.1}$$

has to be solved. This system has a unique solution if the sparsity of U^0 and U^T is sufficiently small and the observations on the right hand side were acquired by a sensors with special properties (section 2.5.4). Tomographic sensors usually fulfill these requirements and A is chosen later as such, as well as U^0 and U^T will be appropriate. If the aforementioned requirements are met it is sufficient to search for any solution e.g. through

$$\bar{u} = \arg\min_x \left\{ \|\bar{A}x - \bar{b}\|_2 \;\middle|\; x \geq 0 \right\} \ . \tag{3.2.2}$$

The estimation of the transport from U^0 to U^T is based on Benamou and Brenier's continuum mechanics formulation of the Wasserstein distance problem (section 2.6.2). It was shown that the solution $\mu(t,x)$ can be obtained from solving

$$\arg\min_\mu \ \max_\phi \ \int_X \int_0^T \frac{\|w\|^2}{2u} \,\mathrm{d}t\,\mathrm{d}x + \int_X \int_0^T \phi\left(\partial_t u + \nabla_x \cdot w\right) \mathrm{d}t\,\mathrm{d}x \tag{3.2.3}$$

which is equivalent to (compare (2.6.19))

$$\arg\max_\mu \ \min_{\phi,q} \ \delta_K(q) + F_1(\phi) + \int_0^T \int_X \mu \cdot (\nabla_{t,x}\phi - q)\,\mathrm{d}x\,\mathrm{d}t \tag{3.2.4}$$

with K as defined in (2.6.15) and

$$F_1(\phi) = \int_X \phi(0,x)U^0 - \phi(T,x)U^T \,\mathrm{d}x \ .$$

The weak coupling and scaled ADMM approaches both combine (3.2.2) and (3.2.4), whereas the PPXA approach uses (3.2.1) and (3.2.3). Each of the subsequent three sections derives one of the mentioned approaches.

3.3 Weak Coupling Approach

The first optimization approach for solving the reconstruction and motion joint problem is to utilize the weak coupling algorithm 2.4 as the main method. After discretization and fixing an order of grid nodes the saddle-point problem (3.2.4) reads

$$\max_{\mu \in \mathbb{R}^{N(1+D)}} \min_{\substack{q \in \mathbb{R}^{N(1+D)} \\ \phi \in \mathbb{R}^{N}}} \delta_K(q) + F_1(\phi) + \sum_{i=1}^{N} \mu_i^\top (\nabla_{t,x}\phi_i - q_i) \tag{3.3.1}$$

where K and $F_1(\phi)$ are reused for their discrete versions

$$K = \left\{ q \in \mathbb{R}^{N(1+D)} \;\middle|\; q_i = \begin{bmatrix} \alpha_i \\ \beta_i \end{bmatrix} \in \mathbb{R}^{1+D},\ \alpha_i + \frac{\|\beta_i\|^2}{2} \leq 0 \quad \text{for all } i \in [N] \right\} \tag{3.3.2}$$

and

$$F_1(\phi) = \sum_{i_1=1}^{X_1} \cdots \sum_{i_D=1}^{X_D} \phi_{(1,i_1,\dots,i_D)} U^0_{(i_1,\dots,i_D)} - \phi_{(X_0,i_1,\dots,i_D)} U^T_{(i_1,\dots,i_D)} \,,$$

respectively. Now, a preliminary form of the joint problem can be formulated that is the saddle-point problem

$$\min_{\mu} \max_{\phi,q} \frac{\gamma}{2} \left\| \bar{A}\bar{u} - \bar{b} \right\|_2^2 + \delta_{\mathbb{R}^{2n}_+}(\bar{u}) - \delta_K(q) - F_1(\phi,\mu) - \sum_{i=1}^{N} \mu_i^\top (\nabla_{t,x}\phi_i - q_i) \tag{3.3.3}$$

achieved by summing the discretized target functions of the reconstruction problem (3.2.2), the negative motion estimation problem (3.3.1) and additional constraints by using indicator functions. Since the image reconstruction is supposed to be solved jointly with the motion, the images U^0 and U^T representing the time boundary conditions regarding the motion are not known a priori and thus must be substituted by the corresponding variables $\mu_{0,(1,i_1,\dots,i_D)}$ and $\mu_{0,(X_0,i_1,\dots,i_D)}$, respectively, leading to

$$F_1(\phi,\mu) = \sum_{i_1=1}^{X_1} \cdots \sum_{i_D=1}^{X_D} \phi_{(1,i_1,\dots,i_D)} \mu_{0,(1,i_1,\dots,i_D)} - \phi_{(X_0,i_1,\dots,i_D)} \mu_{0,(X_0,i_1,\dots,i_D)} \,.$$

Furthermore, a parameter $\gamma > 0$ is introduced weighting the first term responsible for the reconstruction relative to the motion estimation terms. Now, the target function in (3.3.3) is divided into the sum of

$$\begin{aligned} F(\mu) &:= \max_{\phi,q} \left\{ -\delta_K(q) - F_1(\phi,\mu) - \sum_{i=1}^{N} \mu_i^\top (\nabla_{t,x}\phi_i - q_i) \right\} \\ \text{and} \qquad G(\bar{u}) &:= \frac{\gamma}{2} \left\| \bar{A}\bar{u} - \bar{b} \right\|_2^2 + \delta_{\mathbb{R}^{2n}_+}(\bar{u}) \end{aligned}$$

where $\bar{u}$ consists of specific entries of μ, namely $\bar{u} = \begin{bmatrix} \mu_{1,(1,\cdot,\dots,\cdot)} \\ \mu_{1,(T,\cdot,\dots,\cdot)} \end{bmatrix}$. Moreover, let $M \in \{0,1\}^{2n \times (D+1)N}$ be the matrix which selects the corresponding entries from μ to form $\bar{u}$ so that $M\mu = \bar{u}$. This natural dependency is incorporated in the coupling term of the weak coupling problem (2.4.5) such that algorithm 2.4 can be applied to

$$\min_{\mu,\bar{u}} F(\mu) + G(\bar{u}) + \|M\mu - \bar{u}\|_2^2 \tag{3.3.4}$$

leading to the preliminary algorithm 3.1.

Algorithm 3.1. **-** Prelininary Weak Coupling Algorithm for Joint Problem (3.3.4)

```
   Fix $\mu^0 \in \mathbb{R}^{(1+D)N}$, $\bar{u}^0 = M\mu^0 \in \mathbb{R}^{2n}$, $\rho, \gamma, \zeta, \eta > 0$
1: for $k = 0, 1, \dots$ do
2:     $\mu^{k+1} = \arg\min_{\mu \in \mathbb{R}^{(1+D)N}} F(\mu) + \frac{\rho}{2}\|M\mu - \bar{u}^k\|_2^2 + \frac{\zeta}{2}\|\mu - \mu^k\|_2^2$
3:     $\bar{u}^{k+1} = \arg\min_{\bar{u} \in \mathbb{R}^{2n}} \left\{ \frac{\gamma}{2} \left\|\bar{A}\bar{u} - \bar{b}\right\|_2^2 + \frac{\rho}{2} \left\|\bar{u} - M\mu^{k+1}\right\|_2^2 + \frac{\eta}{2} \left\|\bar{u} - \bar{u}^k\right\|_2^2 \middle| \bar{u} \geq 0 \right\}$
4: end for
```

The second iterative step in line 3 is a simple least square problem with constant bound constraints and can be solved right away. But, since $F(\mu)$ is a maximization problem itself, line 2 still is a saddle-point problem. Its solution is the topic of the next section.

3.3.1 Inner Saddle-Point Problem

Algorithm 3.1 contains an inner saddle-point problem in line 2 which has to be solved in every iteration. This is achieved by applying Chambolle and Pock's first-order primal-dual algorithm 2.5 (section 2.4.3.3) which requires some reformulation first. After defining

$$\hat{f}(\mu) := \frac{\rho}{2}\left\|M\mu - \bar{u}^k\right\|_2^2 + \frac{\zeta}{2}\left\|\mu - \mu^k\right\|_2^2 \tag{3.3.5}$$

line 2 turns into

$$\begin{aligned} \mu^{k+1} &= \arg\min_{\mu} \; F(\mu) + \hat{f}(\mu) \\ &= \arg\max_{\mu} \; \min_{\phi,q} \; \delta_K(q) + F_1(\phi,\mu) + \sum_{i=1}^{N} \mu_i^\top(\nabla\phi_i - q_i) - \hat{f}(\mu) \\ &= \arg\max_{\mu} \; \min_{\phi,q} \; \delta_K(q) + F_1(\phi,\mu) + \sum_{j=0}^{D}\sum_{i=1}^{N} \mu_{j,i}(\partial_j\phi_i - q_{j,i}) - \hat{f}(\mu) \; . \end{aligned} \tag{3.3.6}$$

The gradient operator ∇ is approximated by using first order central finite differences with appropriate one-sided differences at grid boundaries in each dimension, i.e.

$$\sum_{i=1}^{N} \mu_{j,i}(\partial_j\phi_i - q_{j,i}) = \sum_{i_0=1}^{X_0} \cdots \sum_{i_D=1}^{X_D} \mu_{j,(i_0,\dots,i_D)}(\partial_j\phi_{(i_0,\dots,i_D)} - q_{j,(i_0,\dots,i_D)})$$

with

$$\partial_j \phi_{(\dots,i_j,\dots)} \approx \begin{cases} \frac{1}{h_j}\left(\phi_{(\dots,2,\dots)} - \phi_{(\dots,1,\dots)}\right) & \text{for } i_j = 1 \\ \frac{1}{2h_j}\left(\phi_{(\dots,i_j+1,\dots)} - \phi_{(\dots,i_j-1,\dots)}\right) & \text{for } i_j \in \{2,\dots,X_j-1\} \\ \frac{1}{h_j}\left(\phi_{(\dots,X_j,\dots)} - \phi_{(\dots,X_j-1,\dots)}\right) & \text{for } i_j = X_j \end{cases}$$

for every $j \in \{0,\dots,D\}$ leading to

$$\sum_{i=1}^{N} \mu_{j,i}(\partial_j \phi_i - q_{j,i}) = \mu_j^\top \left(\left[\begin{array}{c|c} I_{\frac{N}{X_j}} \otimes \Phi_j & -I_N \end{array}\right] \begin{bmatrix} \phi \\ q_j \end{bmatrix} \right)$$

where

$$\Phi_j := \begin{bmatrix} -\frac{1}{h_j} & \frac{1}{h_j} & 0 & \cdots & 0 \\ -\frac{1}{2h_j} & 0 & \frac{1}{2h_j} & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & -\frac{1}{2h_j} & 0 & \frac{1}{2h_j} \\ 0 & \cdots & 0 & -\frac{1}{h_j} & \frac{1}{h_j} \end{bmatrix} \in \mathbb{R}^{X_j \times X_j} .$$

Here a reordering of the grid nodes takes place so that the new order of μ and q is

$$\mu = \begin{bmatrix} \mu_0 \\ \vdots \\ \mu_D \end{bmatrix} \quad \text{with} \quad \mu_j = \mu_{j,i} = \mu_{j,(\cdot,\dots,\cdot)} \quad \text{for} \quad j \in \{0,\dots,D\} \quad \text{and} \quad i \in [N]$$

and q correspondingly. Consequently, these approximations applied to (3.3.6) lead to the saddle-point problem

$$\begin{aligned} &\max_{\mu} \min_{\phi,q} \ \delta_K(q) + F_1(\phi,\mu) + \sum_{j=0}^{D}\sum_{i=1}^{N} \mu_{j,i}(\partial_j \phi_i - q_{j,i}) - \hat{f}(\mu) \\ \approx &\max_{\mu} \min_{\psi} \ \langle \Phi\psi, \mu \rangle + \delta_{\bar{K}}(\psi) - \hat{f}(\mu) \end{aligned} \tag{3.3.7}$$

with

$$\Phi := \left[\begin{array}{c|c} I_{\frac{N}{X_0}} \otimes \bar{\Phi}_0 & \\ I_{\frac{N}{X_1}} \otimes \Phi_1 & \\ \vdots & -I_{DN} \\ I_{\frac{N}{X_D}} \otimes \Phi_D & \end{array}\right] \in \mathbb{R}^{(1+D)N \times (2+D)N} ,$$

$$\psi := \begin{bmatrix} \phi \\ q \end{bmatrix} \in \mathbb{R}^{(2+D)N} \qquad \text{and} \qquad \bar{K} := \left\{ \begin{bmatrix} \phi \\ q \end{bmatrix} \,\middle|\, q \in K \right\}$$

where

$$\bar{\Phi}_0 := \Phi_0 + \begin{bmatrix} 1 & 0 & \cdots & 0 & 0 \\ 0 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & \cdots & 0 & -1 \end{bmatrix}$$

obtained by combining $F_1(\phi, \mu)$ with $\sum_{i=1}^{N} \mu_{0,i}(\partial_0 \phi_i - q_{0,i})$.

Finally, the form of (3.3.7) is the requirement for the application of Chambolle and Pock's first-order primal-dual algorithm 2.5 (CP). Thus, μ^{k+1} in line 2 of the preliminary weak coupling algorithm 3.1 can be computed through solving (3.3.7) by algorithm 3.2.

Algorithm 3.2. - CP Algorithm Applied to Joint Problem (3.3.7), CP($\mu^0, \hat{f}$)

```
Input $\mu^0 \in \mathbb{R}^{N(D+1)}$
Fix $\psi^0 \in \mathbb{R}^{N(D+2)}$, $\bar{\psi}^0 = \psi^0$, $\theta \in [0,1]$, $\tau > 0$, $\sigma > 0$
for $k = 0, 1, \dots$ do
    $\mu^{k+1} = \arg\min_{\mu \in \mathbb{R}^{N(1+D)}} \left\{ \frac{\|\mu - \mu^k - \sigma \Phi \bar{\psi}^k)\|_2^2}{2\sigma} + \hat{f}(\mu) \right\}$
    $\psi^{k+1} = \arg\min_{\psi \in \mathbb{R}^{N(D+2)}} \left\{ \frac{\|\psi - \psi^k + \tau \Phi^\top \mu^{k+1}\|_2^2}{2\tau} \,\middle|\, \psi \in \bar{K} \right\}$
    $\bar{\psi}^{k+1} = \psi^{k+1} + \theta \left( \psi^{k+1} - \psi^k \right)$
end for
```

The final weak coupling algorithm 3.3 that is supposed to reconstruct images and compute the transport jointly thus uses algorithm 3.2 as an inner loop in the first iterative step.

Algorithm 3.3. - Weak Coupling Algorithm for Joint Problem (3.3.4)

```
Fix $\mu^0 \in \mathbb{R}^{(1+D)N}$, $\bar{u}^0 = M\mu^0 \in \mathbb{R}^{2n}$, $\rho, \gamma, \zeta, \eta > 0$
for $k = 0, 1, \dots$ do
    $\mu^{k+1} = \mathrm{CP}\left( \mu^k, \mu \to \frac{\rho}{2} \left\| M\mu - \bar{u}^k \right\|_2^2 + \frac{\zeta}{2} \left\| \mu - \mu^k \right\|_2^2 \right)$        (algorithm 3.2)
    $\bar{u}^{k+1} = \arg\min_{\bar{u} \in \mathbb{R}^{2n}} \left\{ \frac{\gamma}{2} \left\| \bar{A}\bar{u} - \bar{b} \right\|_2^2 + \frac{\rho}{2} \left\| \bar{u} - M\mu^{k+1} \right\|_2^2 + \frac{\eta}{2} \left\| \bar{u} - \bar{u}^k \right\|_2^2 \middle| \bar{u} \geq 0 \right\}$
end for
```

3.3.1.1 Implementation Details

This section describes the implementation details of algorithm 3.2. The function $\hat{f}(\mu)$ in (3.3.5) is a least square term and can be rearranged as

$$\hat{f}(\mu) = \frac{\lambda}{2} \|L\mu - l\|_2^2$$

with $L \in \mathbb{R}^{p \times N(D+1)}$, $l \in \mathbb{R}^p$ and $\lambda > 0$. By merging both terms in line 2 of algorithm 3.2, this turns the minimization with respect to μ into the simple unconstrained linear least squares problem

$$\mu^{k+1} = \operatorname*{arg\,min}_{\mu \in \mathbb{R}^{N(1+D)}} \frac{1}{2} \left\| \begin{bmatrix} \frac{1}{\sqrt{\sigma}} I_{N(1+D)} \\ \sqrt{\lambda} L \end{bmatrix} \mu - \begin{bmatrix} \frac{1}{\sqrt{\sigma}} \left(\mu^k + \sigma \Phi \bar{\psi}^k\right) \\ \sqrt{\lambda} l \end{bmatrix} \right\|_2^2 .$$

The minimization with respect to ψ in line 3 is a least square problem, as well, but it requires reformulation due to the nonlinear constraint $\psi \in \bar{K}$. The set

$$\bar{K} = \left\{ \psi \in \mathbb{R}^{(2+D)N} \;\middle|\; \psi_i = \begin{bmatrix} \phi_i \\ \alpha_i \\ \beta_i \end{bmatrix} \in \mathbb{R}^{2+D},\ \alpha_i + \frac{\|\beta_i\|_2^2}{2} \le 0 \quad \text{for all } i \in [N] \right\}$$

induces a natural separation of the problem. Since the inequality constraint is required for every $i \in [N]$, it is equivalent to the pointwise problem

$$\begin{bmatrix} \phi_i^{k+1} \\ \alpha_i^{k+1} \\ \beta_i^{k+1} \end{bmatrix} = \operatorname*{arg\,min}_{\phi_i \in \mathbb{R},\, \alpha_i \in \mathbb{R},\, \beta_i \in \mathbb{R}^D} \left\{ \frac{1}{2} \left\| \begin{bmatrix} \phi_i \\ \alpha_i \\ \beta_i \end{bmatrix} - \left(\psi^k - \tau \Phi^\top \mu^{k+1}\right)_i \right\|_2^2 \;\middle|\; \alpha_i + \frac{\|\beta_i\|_2^2}{2} \le 0 \right\}$$

which can be solved for each $i \in [N]$, separately. Because ϕ_i is not affected by the constraint, the optimal solution is the corresponding entry of $\left(\psi^k - \tau \Phi^\top \mu^{k+1}\right)_i$. The remaining minimization problems have the form

$$\min_{x \in \mathbb{R}^{1+D}} \left\{ \frac{1}{2} \|x - z\|_2^2 \;\middle|\; x_1 + \frac{1}{2} \sum_{i=2}^{1+D} x_i^2 \le 0 \right\} \tag{3.3.8}$$

where $x := \begin{bmatrix} \alpha_i \\ \beta_i \end{bmatrix} \in \mathbb{R}^{1+D}$ and $z \in \mathbb{R}^{1+D}$ is the given vector with corresponding entries of $\left(\psi^k - \tau \Phi^\top \mu^{k+1}\right)_i$. Hereafter, each of those problems is solved analytically. If $x = z$ does not violate the constraints, then it is the optimal solution and the problem is solved. If it does, the optimal solution is located on the boundary of the feasible set, because the target function $\|x - z\|^2$ is convex. In this case the problem changes to (3.3.8) with equality constraint and the corresponding Lagrangian

$$L(x, y) = \frac{1}{2} \sum_{j=1}^{1+D} \left((x_j - z_j)^2\right) + y \left(x_1 + \frac{1}{2} \sum_{i=2}^{1+D} x_i^2\right)$$

minimized by the same variable values is used to search for this minimum (section 2.4).

Requiring the derivatives

$$\nabla_x L(x,y) = \begin{bmatrix} x_1 - z_1 + y \\ x_2 - z_2 + yx_2 \\ \vdots \\ x_{1+D} - z_{1+D} + yx_{1+D} \end{bmatrix} \in \mathbb{R}^{1+D} \quad \text{and} \quad \partial_y L(x,y) = x_1 + \frac{1}{2}\sum_{i=2}^{1+D} x_i^2$$

to vanish leads to

$$x_1 = z_1 - y \ , \quad x_j = \frac{z_j}{1+y} \ \text{ for } j \in \{2,\dots,1+D\} \quad \text{and} \quad x_1 = -\frac{1}{2}\sum_{i=2}^{1+D} x_i^2 \tag{3.3.9}$$

and the elimination of x_i to

$$y^3 + (2-z_1)y^2 + (1-2z_1)y - z_1 - \frac{1}{2}\sum_{i=2}^{1+D} z_i^2 = 0 \ .$$

In order to obtain x from (3.3.9) the roots of this cubic polynomial in y are required. Those are calculated analytically with the help of the classical formulas of Cardano's method. A real solution $\hat{y}$ is desired here as of which there is at least one and three at most. If there is only one real solution, there is nothing more to do. If there are more, $\hat{y}_1, \hat{y}_2, \hat{y}_3$, then the one which yields the smallest value of the target function in (3.3.8) is chosen, i.e.

$$\hat{y} = \operatorname*{arg\,min}_{y \in \{\hat{y}_1,\hat{y}_2,\hat{y}_3\}} \frac{1}{2} \left\| \begin{bmatrix} z_1 - y \\ \frac{z_2}{1+y} \\ \vdots \\ \frac{z_{1+D}}{1+y} \end{bmatrix} - z \right\|_2^2 \ .$$

3.4 Scaled ADMM Approach

The basis of the second approach for solving the reconstruction and motion joint problem is scaled ADMM (section 2.4.3.1). In contrast to plain ADMM (algorithm 2.2) having two appended terms originating from the constraint, there is a linear part only in scaled ADMM (algorithm 2.3) which is an advantage here. In order to meet the requirements, the target function of the saddle-point problem (3.3.3) is divided in the same way as the weak coupling approach (section 3.3) into the sum of

$$\begin{aligned} & F(\mu) := \max_{\phi,q} \left\{ -\delta_K(q) - F_1(\phi,\mu) - \sum_{i=1}^{N} \mu_i^\top (\nabla_{t,x}\phi_i - q_i) \right\} \\ \text{and} \qquad & G(\bar{u}) := \frac{\gamma}{2} \left\| \bar{A}\bar{u} - \bar{b} \right\|_2^2 + \delta_{\mathbb{R}_+^{2n}}(\bar{u}) \ . \end{aligned}$$

Again, let $M \in \{0,1\}^{2n\times(D+1)N}$ be the matrix with $M\mu = \bar{u}$ leading to the problem

$$\min_{\mu,\bar{u}} \{F(\mu) + G(\bar{u}) \mid M\mu - \bar{u} = 0\} \tag{3.4.1}$$

which is the form scaled ADMM can handle. Thus, scaled ADMM applied to problem (3.4.1) yields algorithm 3.4.

Algorithm 3.4. - Scaled ADMM for Joint Problem (3.3.3)

```
   Fix $\bar{u}^0 \in \mathbb{R}^{2n}$, $r^0 \in \mathbb{R}^{2n}$, $\rho, \gamma > 0$
1: for $k = 0, 1, \dots$ do
2:     $\mu^{k+1} = \mathrm{CP}\left(\mu^k, \mu \to \frac{\rho}{2}\left\|M\mu - \bar{u}^k + r^k\right\|_2^2\right)$     (algorithm 3.2)
3:     $\bar{u}^{k+1} = \arg\min_{\bar{u}\in\mathbb{R}^{2n}} \left\{\frac{\gamma}{2}\left\|\bar{A}\bar{u} - \bar{b}\right\|_2^2 + \frac{\rho}{2}\left\|M\mu^{k+1} - \bar{u} + r^k\right\|_2^2 \;\middle|\; \bar{u} \geq 0\right\}$
4:     $r^{k+1} = r^k + M\mu^{k+1} - \bar{u}^{k+1}$
5: end for
```

The first iterative step in line 2 is obtained from (3.3.6), as well, by just setting

$$\hat{f}(\mu) := \frac{\rho}{2}\left\|M\mu - \bar{u}^k + r^k\right\|_2^2$$

such that μ^{k+1} can be computed by algorithm 3.2. Since $\hat{f}(\mu)$ is a least squares term the implementation details (section 3.3.1.1) are valid, too. As in the weak coupling approach, the second iterative step in line 3 is a least square problem with constant bound constraints not requiring any reformulation effort.

3.5 Parallel Proximal Algorithm Approach

The basis for the third approach to solve the reconstruction and motion estimation joint problem is the parallel proximal algorithm 2.6. The motion part of the objective function to which the algorithm is applied to, originates from Benamou and Brenier's continuum mechanics formulation (3.2.3) slightly rewritten as

$$\min_{\mu} \int_X \int_0^T \frac{\|w\|^2}{2u}\,\mathrm{d}t\,\mathrm{d}x + \int_X \int_0^T \delta_{\{\nabla\cdot\mu=0\}}(\mu)\,\mathrm{d}t\,\mathrm{d}x \tag{3.5.1}$$

where the Lagrangian multiplier ϕ is replaced by an indicator function in order to add the constraint. As shown in (2.6.18) the first term can be written as a supremum equal to the support function, i.e. $\frac{\|w\|^2}{2u} = \sigma_K(\mu)$ with the same set K as in (2.6.15). Thus, after discretization of (3.5.1) and K to (3.3.2), another version of the joint problem

$$\min_{\mu\in\mathbb{R}^{N(1+D)}} F_1(\mu) + F_2(\mu) + F_3(\mu) + F_4(\mu) \tag{3.5.2}$$

with

$$F_1(\mu) := \sigma_K(\mu), \qquad F_2(\mu) := \sum_{i=1}^{N} \delta_{\{\nabla\cdot\mu_i=0\}}(\mu_i),$$
$$F_3(\mu) := \delta_{\{\bar{A}\bar{u}=\bar{b}\}}(\bar{u}) \quad \text{and} \quad F_4(\mu) := \delta_{\mathbb{R}^{2n}_+}(\bar{u})$$

is formed by adding the constraint of the recovery system and the positivity constraint as indicator function in form of $F_3(\mu)$ and $F_4(\mu)$ in order to bring in the reconstruction (3.2.1). In this form the PPXA is immediately applicable yielding algorithm 3.5 which, however, requires the proximal operators (definition 2.30) for F_1, F_2, F_3 and F_4 to be available. Those are derived next.

Algorithm 3.5. **-** PPXA for Joint Problem (3.5.2)

```
    Fix $y_1^0, y_2^0, y_3^0, y_4^0 \in \mathbb{R}^{(1+D)N}$, $\epsilon \in ]0,1[$, $\gamma > 0$
    and $\omega_1, \omega_2, \omega_3, \omega_4 \in ]0,1]$ such that $\omega_1 + \omega_2 + \omega_3 + \omega_4 = 1$
    $\mu^0 = \omega_1 y_1^0 + \omega_2 y_2^0 + \omega_3 y_3^0 + \omega_4 y_4^0$
    for $k = 0, 1, \dots$ do
        $p_1^k = \text{prox}\ \frac{\gamma}{\omega_1} F_1(y_1^k)$
        $p_2^k = \text{prox}\ \frac{\gamma}{\omega_2} F_2(y_2^k)$
        $p_3^k = \text{prox}\ \frac{\gamma}{\omega_3} F_3(y_3^k)$
        $p_4^k = \text{prox}\ \frac{\gamma}{\omega_4} F_4(y_4^k)$
        $p^k = \omega_1 p_1^k + \omega_2 p_2^k + \omega_3 p_3^k + \omega_4 p_4^k$
        Choose $\lambda^k \in [\epsilon, 2-\epsilon]$
        $y_1^{k+1} = y_1^k + \lambda^k(2p^k - \mu^k - p_1^k)$
        $y_2^{k+1} = y_2^k + \lambda^k(2p^k - \mu^k - p_2^k)$
        $y_3^{k+1} = y_3^k + \lambda^k(2p^k - \mu^k - p_3^k)$
        $y_4^{k+1} = y_4^k + \lambda^k(2p^k - \mu^k - p_4^k)$
        $\mu^{k+1} = \mu^k + \lambda^k(p^k - \mu^k)$
    end for
```

3.5.1 Derivation of the Proximal Operators

Following [CP11b], the proximal operator of the scaled support function $c\sigma_K(\mu)$ with $c > 0$ is $\mu - cP_K(\frac{\mu}{c})$ where P_K is the projection onto set K such that

$$\text{prox}\ \frac{\gamma}{\omega_1} F_1(\mu) = \mu_1 - \frac{\gamma}{\omega_1} P_K\left(\frac{\omega_1}{\gamma}\mu\right) = \mu - \frac{\gamma}{\omega_1} \operatorname*{arg\,min}_{q\in K} \left\{ \frac{1}{2} \left\| q - \frac{\omega_1}{\gamma}\mu \right\|_2^2 \right\} .$$

The minimization is the same as in (3.3.8) and is solved identically.

Both F_2 and F_3 have the form $\delta_{\{Mx=\nu\}}$ yielding the proximal operator

$$\begin{aligned} \text{prox}\ c\,\delta_{\{Mx=\nu\}}(x_0) &= \operatorname*{arg\,min}_{x} \left\{ c\,\delta_{\{Mx=\nu\}}(x) + \frac{1}{2}\|x - x_0\|_2^2 \right\} \\ &= \operatorname*{arg\,min}_{x} \left\{ \frac{1}{2}\|x - x_0\|_2^2 \,\middle|\, Mx = \nu \right\} \end{aligned} \qquad (3.5.3)$$

with any factor $c > 0$. The saddle-point formulation with a Lagrangian target function of (3.5.3) is, consequently,

$$\max_y \min_x \frac{1}{2}\|x - x_0\|_2^2 + \langle y, Mx - \nu\rangle = \max_y \left(\min_x \frac{1}{2}\|x - x_0\|_2^2 + \langle y, Mx\rangle\right) - \langle y, \nu\rangle \ . \tag{3.5.4}$$

The solution to the inner minimization problem is obtained by calculating the roots of the derivatives with respect to x leading to

$$x = x_0 - M^\top y \tag{3.5.5}$$

which can be substituted into (3.5.4) yielding

$$\begin{aligned} &\max_x \ \frac{1}{2}\left\|x_0 - M^\top y - x_0\right\|_2^2 + \left\langle y, M(x_0 - M^\top y)\right\rangle - \langle y, \nu\rangle \\ =&\max_x \ \frac{1}{2}\left\|M^\top y\right\|_2^2 - \left\langle y, MM^\top y\right\rangle + \langle y, Mx_0\rangle - \langle y, \nu\rangle \\ =&\max_x \ -\frac{1}{2}\left\|M^\top y\right\|_2^2 + \langle y, Mx_0\rangle - \langle y, \nu\rangle \ . \end{aligned}$$

Again, solving for y such that the derivatives in y vanish gives

$$y = \left(MM^\top\right)^{-1}(Mx_0 - \nu) \ .$$

This can either be inserted into (3.5.5) directly by using the pseudo inverse matrix of $\left(MM^T\right)^{-1}$ or the linear equation system

$$\left(MM^\top\right) y = Mx_0 - \nu$$

can be solved first. It may be necessary to regularize MM^T by $MM^T + \epsilon I$ with a small $\epsilon > 0$ in order to compute the pseudo inverse matrix or to solve the equation system. Either way, the optimal solution to (3.5.3) and thus the desired value of the proximal operator is

$$\operatorname{prox} c\,\delta_{\{Mx=\nu\}}(x_0) = x_0 - M^\top \left(MM^\top\right)^{-1}(Mx_0 - \nu) \ . \tag{3.5.6}$$

This is based on the constraint $Mx = \nu$ being fulfilled strictly. Relaxing it together with its indicator function to $\frac{c}{2}\|Mx - \nu\|_2^2$ yields the proximal operator

$$\begin{aligned} \operatorname{prox} \frac{c}{2}\|Mx - \nu\|_2^2\,(x_0) &= \arg\min_x \left\{\frac{c}{2}\|Mx - \nu\|_2^2 + \frac{1}{2}\|x - x_0\|_2^2\right\} \\ &= \arg\min_x \left\{\frac{1}{2}\left\|\begin{bmatrix}\sqrt{c}M \\ I\end{bmatrix} x - \begin{bmatrix}\sqrt{c}\nu \\ x_0\end{bmatrix}\right\|_2^2\right\} \end{aligned} \tag{3.5.7}$$

which is a simple unconstrained least squares problem. After these derivations the proximal operators for $\frac{\gamma}{\omega_2}F_2$ and $\frac{\gamma}{\omega_3}F_3$ can be formulated.

In $F_2(\mu) = \sum_{i=1}^{N} \delta_{\{\nabla\cdot\mu_i=0\}}(\mu_i)$ the scalar products with the nabla operator for each μ_i are discretized and represented as a matrix vector multiplication of a matrix R with μ taking the sum into account, as well, i.e.

$$F_2(\mu) = \delta_{\{R\mu=0\}}(\mu) \ .$$

Matrix R is chosen to impose one-sided first order differences along each dimension and symmetric boundaries in the space dimensions such that

$$\partial_j \mu_{j,(\dots,i_j,\dots)} \approx \begin{cases} \frac{1}{h_j}\left(\mu_{j,(\dots,i_j+1,\dots)} - \mu_{j,(\dots,i_j,\dots)}\right) & \text{for } i_j \in \{1,\dots,X_j-1\} \\ \frac{1}{h_j}\left(\mu_{j,(\dots,1,\dots)} - \mu_{j,(\dots,X_j,\dots)}\right) & \text{for } i_j = X_j \text{ and } j \in [D] \\ 0 & \text{for } i_0 = X_0 \end{cases} \tag{3.5.8}$$

for every $j \in \{0,\dots,D\}$. The finite differences are intentionally chosen this simple, as it turns out that a more complicated R strongly increases computation time. With $M = R$ and $\nu = 0$ in (3.5.6) this leads to the proximal operator

$$\operatorname{prox} \frac{\gamma}{\omega_2} F_2(\mu) = \mu - R^\top \left(RR^\top\right)^{-1} R\mu \ .$$

The indicator function $F_3(\mu) = \delta_{\{\bar{A}\bar{u}=\bar{b}\}}(\bar{u})$ only acts on specific entries of μ namely $u_{(1,\dots)} = \mu_{0,(1,\dots)}$ and $u_{(X_0,\dots)} = \mu_{0,(X_0,\dots)}$. Since $\bar{A}$ is a block diagonal matrix with blocks of A corresponding to $u_{(1,\dots)}$ and $u_{(X_0,\dots)}$, respectively, the computation can be split. Together with the fact that the proximal operator of the zero function is the identity, applied to the remaining entries of μ, the desired proximal operator is

$$\operatorname{prox} \frac{\gamma}{w_3} \delta_{\{\bar{A}\bar{u}=\bar{b}\}}(\mu) = \begin{bmatrix} \mu_{0,(1,\dots)} - A^\top \left(AA^\top\right)^{-1} \left(A\mu_{0,(1,\dots)} - b_0\right) \\ \mu_{0,(2,\dots)} \\ \vdots \\ \mu_{0,(X_0-1,\dots)} \\ \mu_{0,(X_0,\dots)} - A^\top \left(AA^\top\right)^{-1} \left(A\mu_{0,(X_0,\dots)} - b_T\right) \\ \mu_{1,(\dots)} \\ \vdots \\ \mu_{D,(\dots)} \end{bmatrix} .$$

Likewise, $F_4(\mu) = \delta_{\mathbb{R}_+^{2n}}(\bar{u})$ only acts on the same entries of μ as F_3. But, since positive pixel intensities are favored at intermediate frames in between the start and end images anyway, the positivity constraint is extended to all entries of $\mu_{0,(\dots)}$ so that the required proximal operator is

$$\operatorname{prox}\frac{\gamma}{w_4}\delta_{\{\bar{u}\in\mathbb{R}^{2n}_{+}\}}(\mu) = \begin{bmatrix} \max\left\{0, \mu_{0,(\dots)}\right\} \\ \mu_{1,(\dots)} \\ \vdots \\ \mu_{D,(\dots)} \end{bmatrix}$$

where the maximum is meant pointwise.

3.6 Experiments

In order to evaluate the performance of the weak coupling, the scaled ADMM and the PPXA algorithm derived in sections 3.3, 3.4 and 3.5 experiments are carried out in this section. Those are set up with start and end images which are easy to recover and a primitive transport in between so that the solution to image reconstruction or transport estimation performed separately is simple and the focus lies on the joint computation.

The experimental scenario is chosen to be images with $d = X_1 = X_2 = 64$ pixels in each of the $D = 2$ dimensions. The time is discretized in $X_0 = 32$ steps and the start image U^0 shows $s = 40$ nonzero pixels with value 1 making it s-sparse. Referencing to a real application, these nonzero pixels are called *particles* here. The particles are randomly distributed inside a square shaped region, tendentially located in the upper left corner. A constant shift of all particles towards the bottom right generates the end image U^T. Both images are shown in figure 3.1.

A tomographic sensor matrix $A \in \{0,1\}^{d^2\times(6d-2)}$ is chosen to generate the observations $b_1, b_T \in \mathbb{R}^{6d-2}$ from the time boundary images $U^0, U^T \in \mathbb{R}^{d^2}$ by $b_1 = AU^0$ and $b_T = AU^T$. The sensor utilizes 4 projections onto 1-dimensional subspaces at different angles. It is positive and can be interpreted as adjacency matrix of an unbalanced expander graph. Finally, the time boundary images are nonnegative which allows to apply theorem 2.47. Thus, the solutions U^0 and U^T of $AU^0 = b_1$ and $AU^T = b_T$, respectively, are unique meaning it is enough to search for positive solutions like in (2.5.31) in order to reconstruct. Details about the sensor are illustrated in figure 3.2 whereas figure 3.1 shows the corresponding projections used as input for the following experiments.

In order to obtain a reference solution for the joint reconstruction and motion problem, each task is computed separately. The image recoveries via (2.5.31)

$$\min_u \{\|Au - b_1\| \mid u \geq 0\} \qquad \text{and} \qquad \min_u \{\|Au - b_T\| \mid u \geq 0\}$$

yield perfect reconstructions of the original images U^0 and U^T already shown in figure 3.1, whereas the separate execution of algorithm 2.7 based on Benamou and Brenier's fluid mechanics formulation without reconstruction yields the result shown

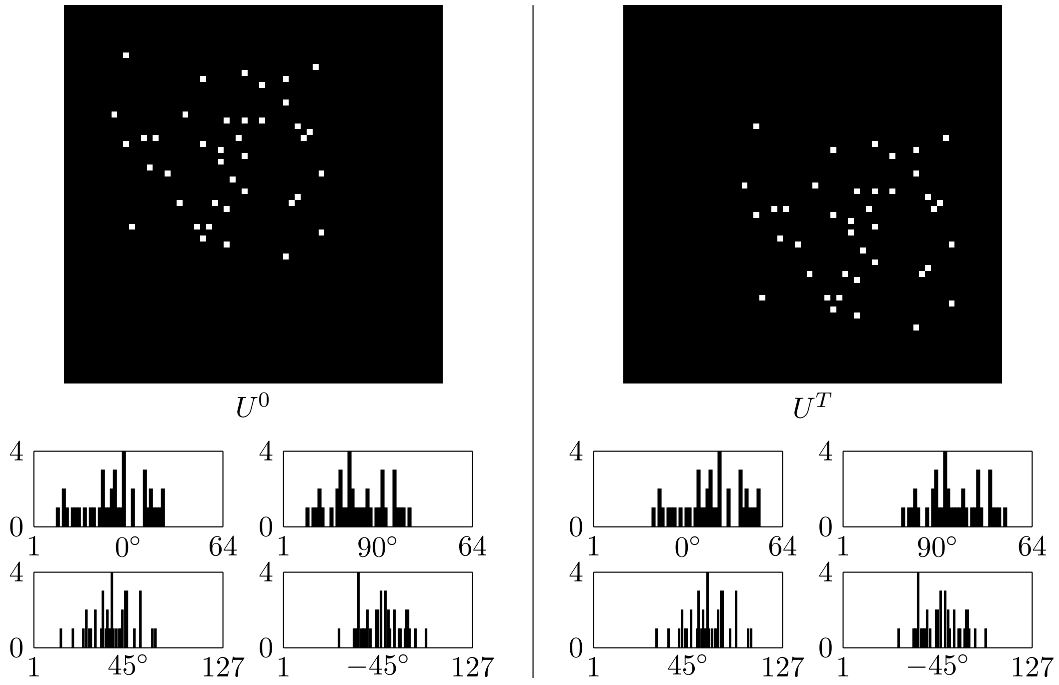


Figure 3.1 - The image vectors U^0 and U^T rearranged as 64×64 images used as the ground truth for the first and last point in time. 40 particles are randomly distributed inside a certain region and shifted towards the lower right corner. 4 projections of each serve as input for algorithms designed for solving the joint reconstruction and motion problem.

in figure 3.3. Only the velocity vectors at pixels having a gray value of or above 0.005 on a scale from 0 (black) to 1 (white) are plotted in order to preserve clarity.

3.6.1 Weak Coupling Experiment

The first experiment is the execution of the weak coupling algorithm 3.3 for the joint problem. For both the outer weak coupling iteration and the inner Chambolle-Pock algorithm 3.2, the norm distance $\left\|\mu^{k+1} - \mu^k\right\|_2$ with respect to the corresponding iteration is employed as stopping criterion. As soon as this distance falls below a certain predefined threshold, the iterative refinement is stopped.

The first time the inner Chambolle-Pock algorithm is executed, $\psi^0 = \mathbb{1}$ is chosen as starting point. In later executions ψ^0 reuses the last value of ψ computed in the inner Chambolle-Pock iteration of the previous outer Weak Coupling step, meaning it is not reset to ψ^0. The entries of the starting point μ^0 corresponding to the start and end images are chosen to be $u^0_{(1,\cdot,\ldots,\cdot)} = U^0$ and $u^0_{(T,\cdot,\ldots,\cdot)} = U^T$, i.e. equal to the optimal image reconstructions. The remaining entries are set to 0. At first, all parameters ρ, γ, ζ, η including those of the inner Chambolle-Pock algorithm θ, τ and σ are set to 1.

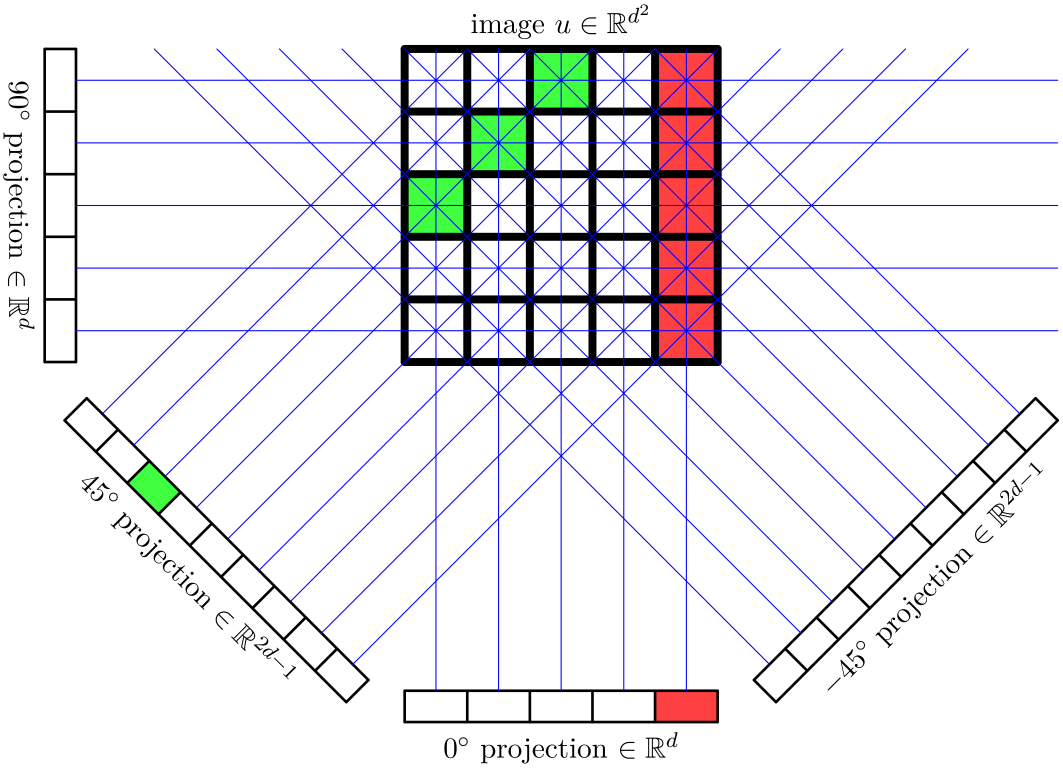


image $u \in \mathbb{R}^{d^2}$
90° projection $\in \mathbb{R}^d$
45° projection $\in \mathbb{R}^{2d-1}$
0° projection $\in \mathbb{R}^d$
−45° projection $\in \mathbb{R}^{2d-1}$

Figure 3.2 - Setup used for the simulated tomographic sensor $A \in \{0,1\}^{d^2 \times (6d-2)}$ with 4 projections along different angles utilized for experiments and illustrated for $d = 5$. The projections are concatenated forming the observation vector $b \in \mathbb{R}^{6d-2}$. $A_{i,j} = 1$ if the j-th ray is incident to the i-th pixel, otherwise $A_{i,j} = 0$. This results in observation b_j being the sum of pixels incident to the corresponding ray so that $Au = b$; illustrated for two rays with the regarding observation and pixels colored in green and red, respectively.

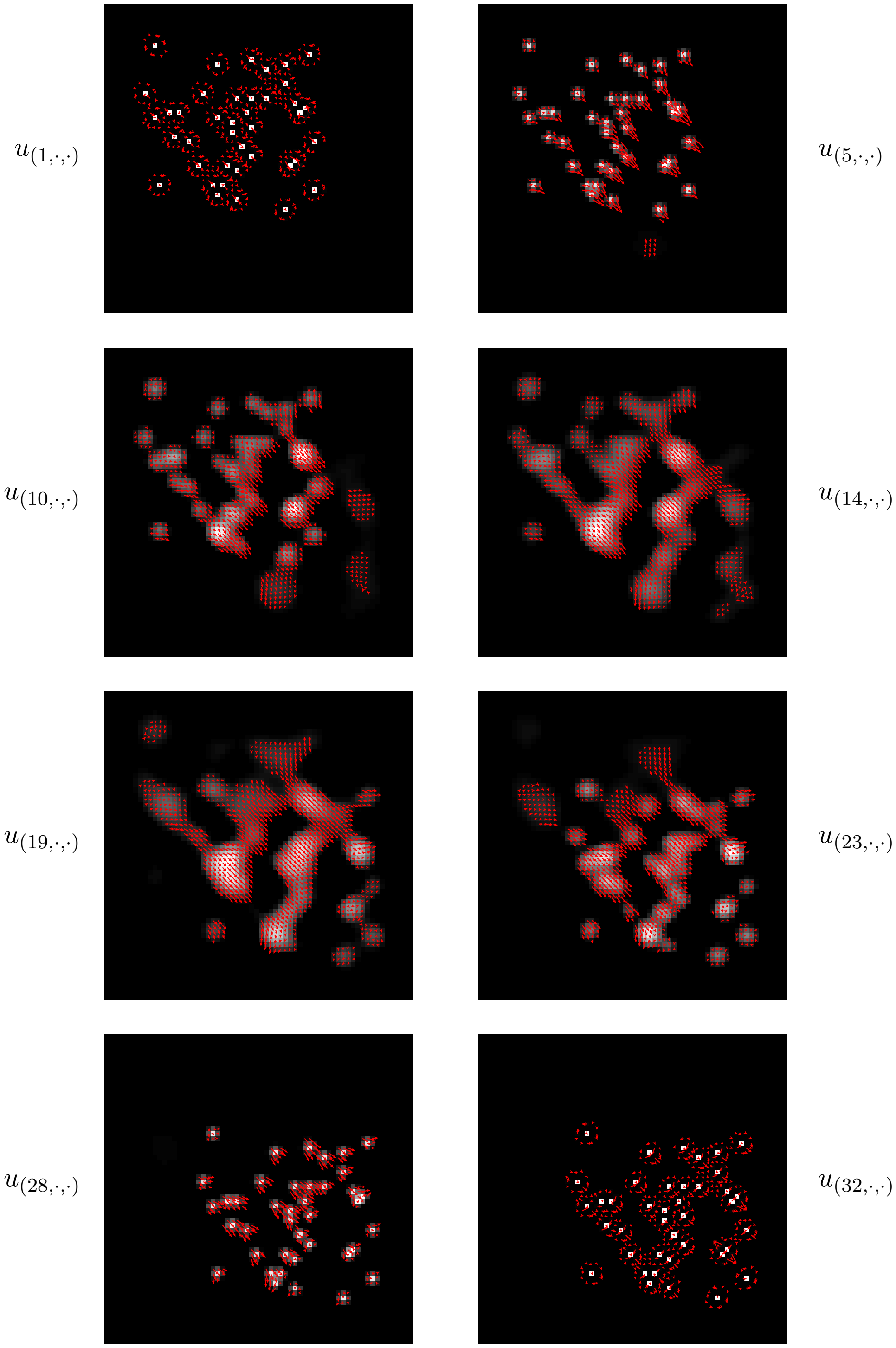


Figure 3.3 **-** Reference motion estimation result without reconstruction computed by using Benamou and Brenier's fluid mechanics framework (section 2.6.2). Two 64×64 images were used as input showing 40 particles shifted towards the lower right corner (figure 3.1). The time was discretized in 32 steps, but besides the first and the last frame only a few more in between are shown. The red arrows illustrate the estimated motion.

Running the weak coupling algorithm 3.3 with the previously described settings does not yield any result since it does not terminate. This is not due to the inner Chambolle-Pock iteration which always stops successfully. The overall algorithm does not stop even though a large number of parameter combinations ρ, γ, ζ and η are considered where it is ensured that τ and σ do not violate the constraint $\tau\sigma\|\Phi\|^2 < 1$ required for proving convergence of the Chambolle-Pock algorithm as described in section 2.4.3.3. The operator norm is estimated as $\|\Phi\| \approx 2.15$ such that $\tau = \sigma = \frac{1}{\|\Phi\|} - \epsilon$ for a small $\epsilon > 0$ is used.

Further simplifications such as stopping after a fixed number of iterations or providing a good starting point including prior knowledge in the form of correctly recovered time boundary images are not successful, either. For this reason no result is shown here. Weak convergence and the loose coupling of the variables μ and $\bar{u}$ do not seem to be sufficient for leading to a converging algorithm. In this regard, the following experiment clearly yields better results.

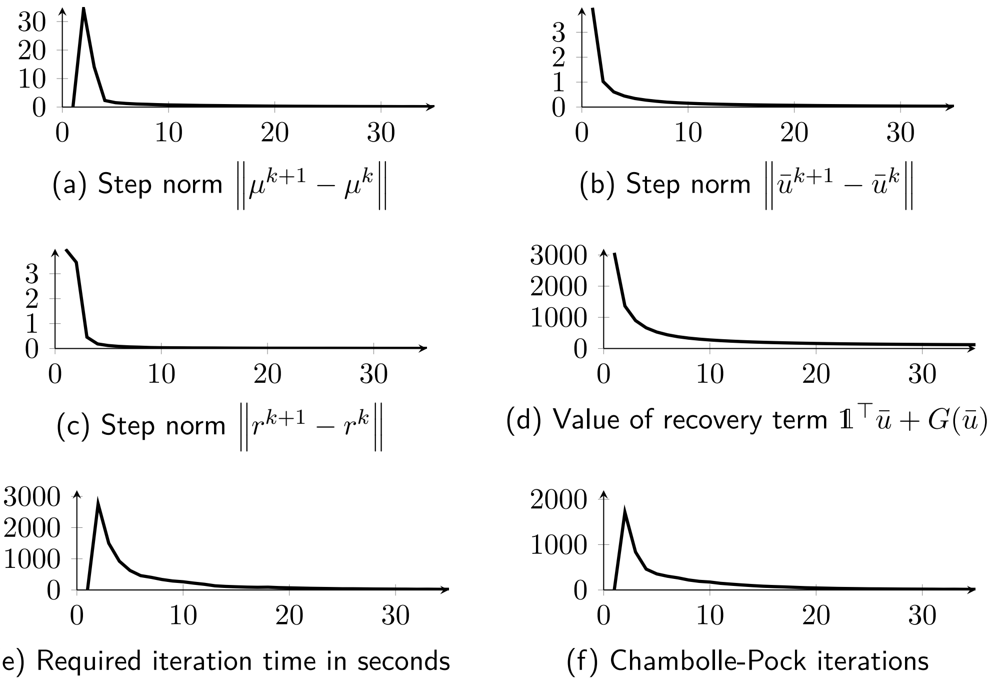
3.6.2 Scaled ADMM Experiment

The subject of the second experiment is the scaled ADMM algorithm 3.4 designed to solve the joint reconstruction and motion estimation problem. Again, the norm distance $\left\|\mu^{k+1} - \mu^k\right\|_2$ is used as stopping criterion for the inner and outer loop. The initial values μ^0, $\bar{u}^0$, r^0 and ψ^0 are chosen to be all 0. Similar to the weak coupling experiment, ψ^0 is not reset for the next execution of the Chambolle-Pock step and parameters of this step are chosen the same in order to satisfy the constraint $\tau\sigma\|\Phi\|^2 < 1$ and $\theta = 1$. Choosing high values for ρ, i.e. a strong coupling of the variables μ and $\bar{u}$ showed to be beneficial for convergence. On the other hand, large ρ also reduce the quality of the recovered images. Hence, $\gamma = \rho = 1000$ seems to provide a good compromise enabling an alternating minimization.

When using the previously described settings, the algorithm converged after 312 iterations and took approximately 4 hours[1]. Still, the result clearly differs from the reference solution. The original time boundary images can just be made out in outlines, the frames in between are heavily flickering and the estimated motion appears random. This flickering is already known from the computation of the reference flow via algorithm 2.7 where it is observed when looking at intermediate results before convergence.

An analysis showing the progression of all three step norms $\left\|\mu^{k+1} - \mu^k\right\|$, $\left\|\bar{u}^{k+1} - \bar{u}^k\right\|$ and $\left\|r^{k+1} - r^k\right\|$ together with the required number of Chambolle-Pock iterations, the value of $\mathbb{1}^\top \bar{u} + G(\bar{u})$ and the required time in each iteration is plotted in figure 3.4. The step norms show that the algorithm clearly converges with regards to the considered criteria. For example, the value of $\mathbb{1}^\top \bar{u} + G(\bar{u})$ seems to converge to 80 which is the correct value, since the time boundary images both have 40 nonzero pixels.

[1] On a PC with an Intel Core i5-2410M together with 8GB memory



(a) Step norm $\left\|\mu^{k+1} - \mu^k\right\|$

(b) Step norm $\left\|\bar{u}^{k+1} - \bar{u}^k\right\|$

(c) Step norm $\left\|r^{k+1} - r^k\right\|$

(d) Value of recovery term $\mathbb{1}^\top \bar{u} + G(\bar{u})$

(e) Required iteration time in seconds

(f) Chambolle-Pock iterations

Figure 3.4 - Progression of selected values occurring in the iterations of the scaled ADMM algorithm for the reconstruction and motion joint problem. Only the first 35 out of 312 iterations are shown on the x-axis. The algorithm is clearly converging.

However, the algorithm stops at $\mathbb{1}^\top \bar{u} + G(\bar{u}) \approx 93.7$ because μ and consequently $\bar{u}$ changes by less than 10^{-4}. For the same reason the number of Chambolle-Pock iterations decreases to 1 and the iteration time to 2 seconds in the end. For the purpose of excluding too early termination of the algorithm, the stopping criterion is removed, leading to the same flickering results even after 130000 iterations and more than 3 days of computation, approximately.

In order to speed up convergence, a modification is applied to the second scaled ADMM step. Instead of biasing the optimization towards the correct time boundary images through $\left\|\bar{A}\bar{u} - \bar{b}\right\|_2$, the aforementioned speed up is enforced by replacing it with $\delta_{\{\bar{A}\bar{u}=\bar{b}\}}(\bar{u})$. This amounts to the second scaled ADMM step changing to

$$\bar{u}^{k+1} = \underset{\bar{u}\in\mathbb{R}^{2n}}{\arg\min} \left\{ \left\|M\mu^{k+1} - \bar{u} + r^k\right\|_2^2 \,\middle|\, \bar{A}\bar{u} = \bar{b},\ \bar{u} \geq 0 \right\} . \tag{3.6.1}$$

After carrying out the experiment with this modification, the algorithm produces the best scaled ADMM result in just 5 iterations shown in figure 3.5. The recovery of the time boundary images is perfect. Obviously, an incomplete reconstruction of those keeps the algorithm running. However, the intermediate frames still flicker and the motion estimation does not improve. Reducing ρ which increases the priority of the motion estimation in the first iterative step does not change anything at it.

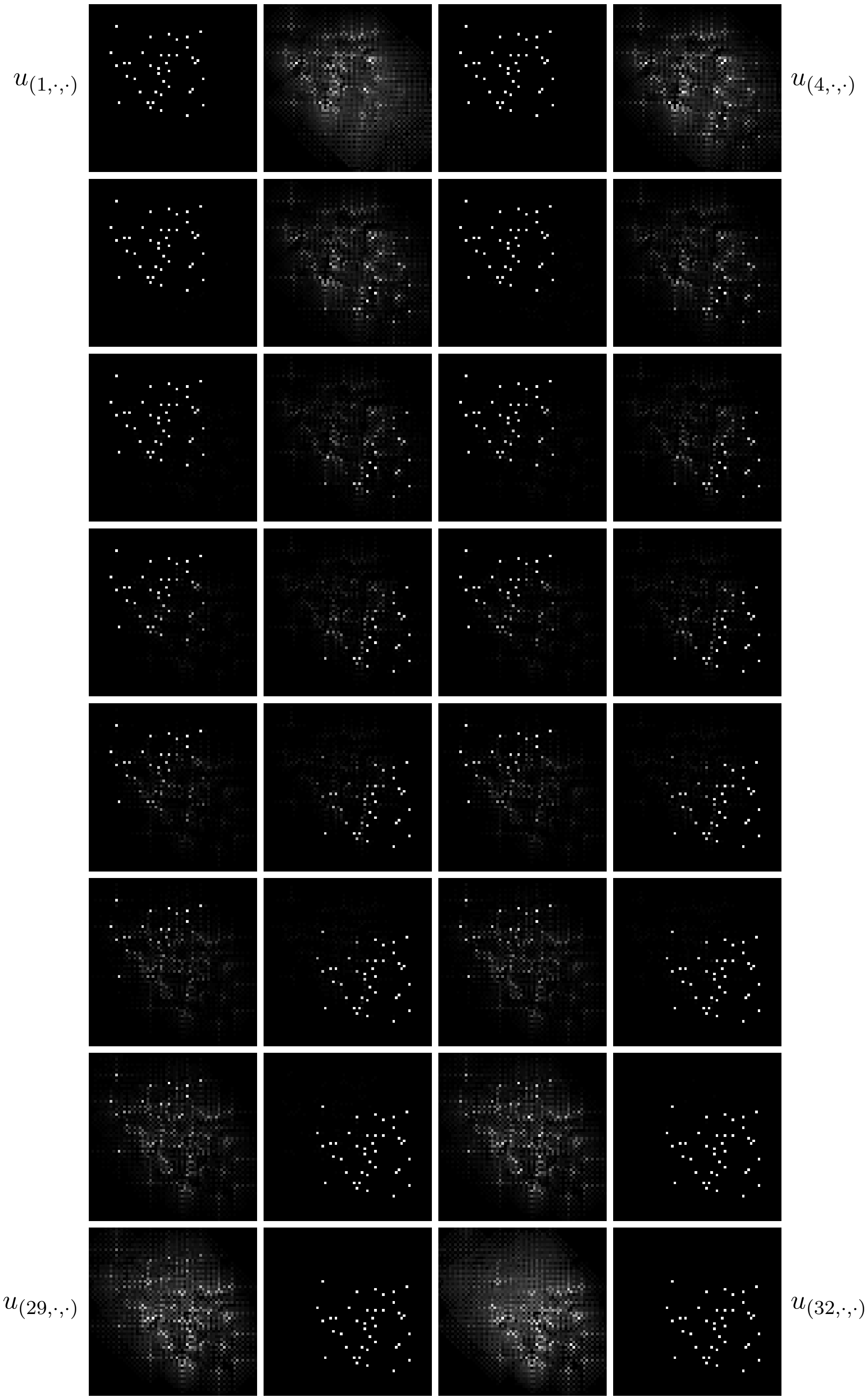


Figure 3.5 **-** Time resolved pixel value output of the scaled ADMM algorithm 3.4 with modification (3.6.1) for the joint problem. The underlying motion is a constant shift of all particles towards the lower right corner. The result flickers in time conveying the impression of $u_{(i,\cdot,\cdot)}$ and $u_{(i+2,\cdot,\cdot)}$ being subsequent. The estimated motion is not shown.

Summing up the observations, it is not sure whether the scaled ADMM algorithm for the solution of the reconstruction and motion joint problem converges. On the one hand it is known that iterating ADMM until a high accuracy is achieved can be very slow as mentioned in section 2.4.3.1, which may be the case here. On the other hand there is the possibility that discontinuities in the method implemented for managing the minimization over the nonlinear set $\bar{K}$ required in the inner Chambolle-Pock algorithm prevent accurate results. It would require further investigation, relaxation, parameter tuning and several weeks of computation to issue a statement in this regard which is refrained from in favor of the more promising approach detailed in the next section.

3.6.3 PPXA Experiments

The basis for the third approach to solve the reconstruction and motion estimation joint problem is the Parallel Proximal Algorithm 3.5. The iteration is stopped when $\left\|\mu^{k+1} - \mu^k\right\|_2$ falls below a certain threshold. As initial values $y_1^0 = y_2^0 = y_3^0 = y_4^0 = 0$ and parameters $\epsilon = \frac{1}{2}$, $\gamma = 1$ and $\omega_1 = \omega_2 = \omega_3 = \omega_4 = \frac{1}{4}$ are set. The functions F_2 and F_3 are strictly imposing their regarding constraint on μ as described in (3.5.6).

After 1000 steps of PPXA iterations, the algorithm yields the result shown in figure 3.6. Similarities of the recovered time boundary images are recognizable and the estimated motion tends to go in the correct direction. Further iterations does not seem to improve the results significantly. The analysis in figure 3.7 shows shrinking step norms supporting the impression of convergence. In contrast to that at least $F_1(\mu)$ (figure 3.7b) does not converge and $D\mu$ in $F_2(\mu)$ (figure 3.7d) is not fulfilled. Function F_1 heavily oscillates and F_2 even recedes from the optimal value 0. This seems contradictory to declining step norms, but it might be due to PPXA having the feature “that some error [...] is tolerated in the computation of the [...] proximity operator” [CP08].

Thanks to the modest discretization of the nabla operator (3.5.8) in F_2 one PPXA iteration takes approximately 1 second. More sophisticated finite differences leads to an increase of iteration time to several minutes and more, and thus investigating this option is omitted.

Results can be influenced by changing the weights ω_1, ω_2, ω_3 and ω_4 assigning different priorities to the corresponding target function terms. Thus, choosing ω_3 multiple times greater than the remaining parameters, optionally together with ω_4, results in an improved reconstruction of the time boundary images after 1000 iterations whereas the motion appears random. Vice versa, for higher ω_2, a more clear motion is recognizable which is, however, based on wrong time boundary image reconstructions. Varying ω_1 has no visible effect on the results. It turns out that distributing the weights equally is a good compromise yielding a result which is closest to the reference solution.

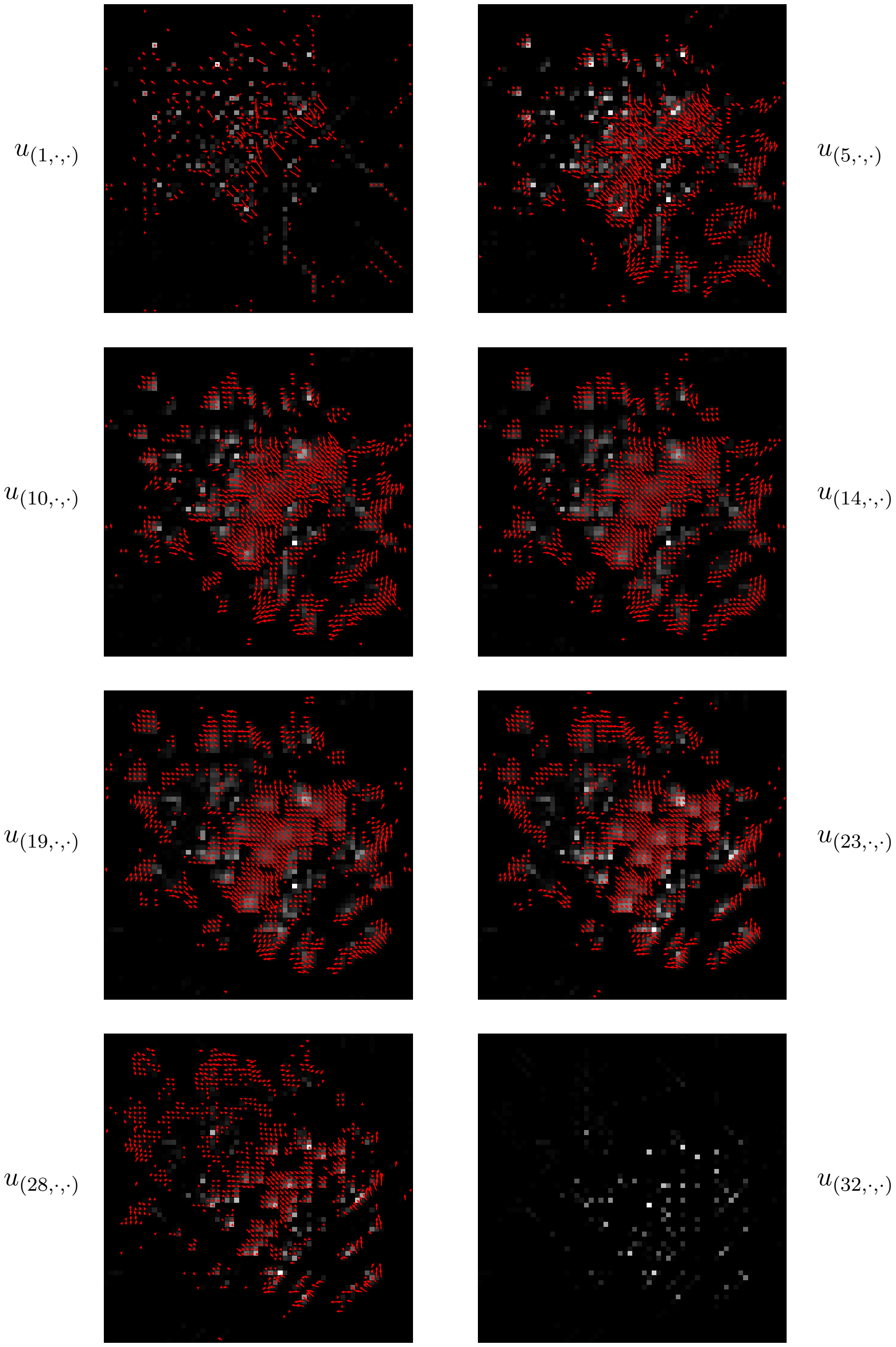
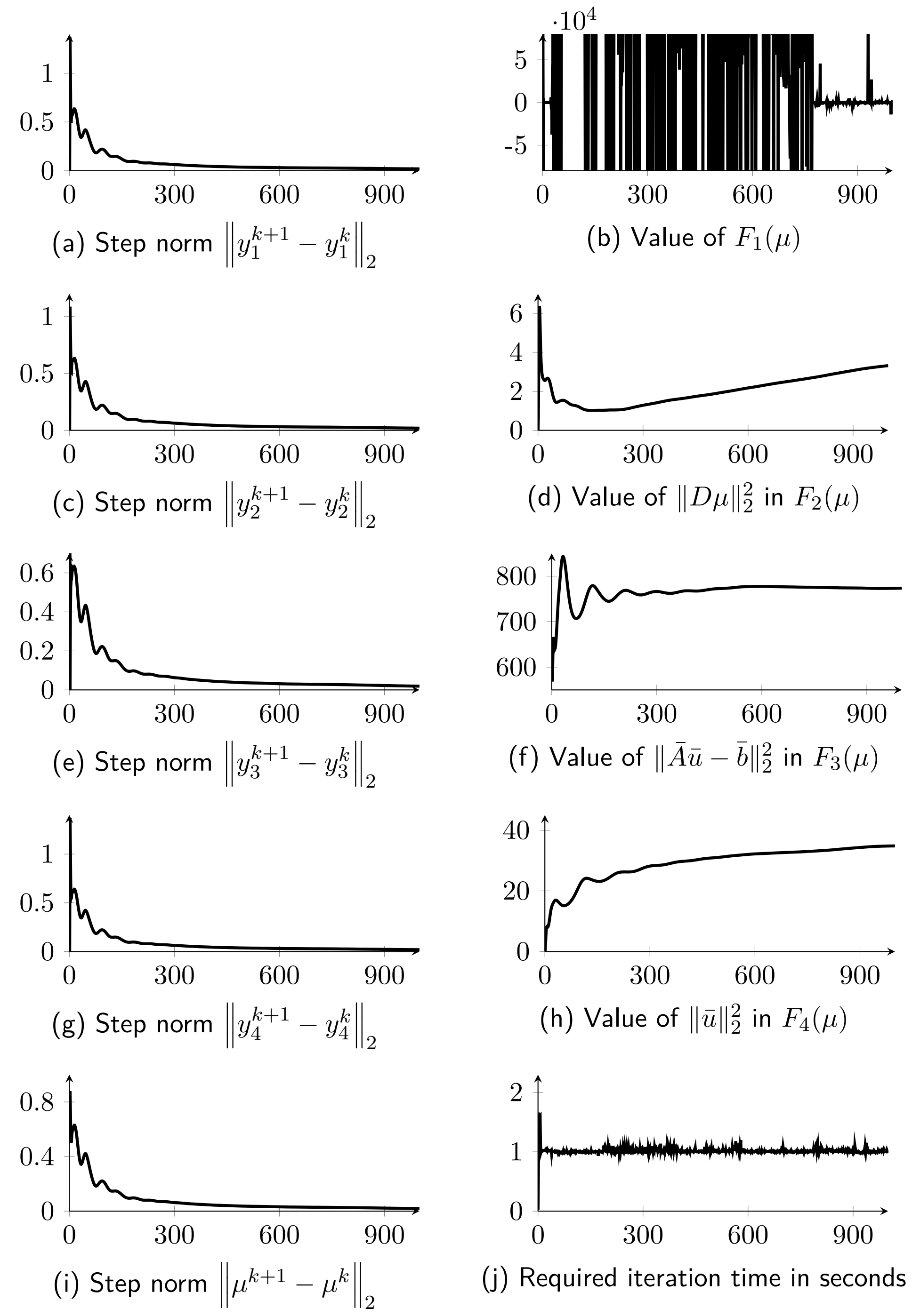


Figure 3.6 - Result of PPXA applied to the joint problem (algrithm 3.5) after 1000 iterations. Between the time boundary images $u_{(1,\cdot,\cdot)}$ and $u_{(32,\cdot,\cdot)}$ only a few selected intermediate frames are shown. For clarity reasons, the motion is only illustrated for pixel values having a gray value greater or equal to 0.005 as red arrows. A trend towards the reference result (figure 3.3) is recognizable.



(a) Step norm $\left\| y_1^{k+1} - y_1^k \right\|_2$

(b) Value of $F_1(\mu)$

(c) Step norm $\left\| y_2^{k+1} - y_2^k \right\|_2$

(d) Value of $\|D\mu\|_2^2$ in $F_2(\mu)$

(e) Step norm $\left\| y_3^{k+1} - y_3^k \right\|_2$

(f) Value of $\|\bar{A}\bar{u} - \bar{b}\|_2^2$ in $F_3(\mu)$

(g) Step norm $\left\| y_4^{k+1} - y_4^k \right\|_2$

(h) Value of $\|\bar{u}\|_2^2$ in $F_4(\mu)$

(i) Step norm $\left\| \mu^{k+1} - \mu^k \right\|_2$

(j) Required iteration time in seconds

Figure 3.7 - Evolution of selected values occurring in the iterations of the PPXA algorithm for the reconstruction and motion joint problem after 1000 iterations. Despite of declining step norms, the algorithm does not seem to converge as indicated by the plots (b) and (d).

Experimenting with different combinations of the other parameters ϵ and γ and the relaxed versions of F_2 and F_3 according to (3.5.7) does not improve results. Neither, more iterations help in this regard since reconstruction and motion become more and more dissimilar to the reference as it can be observed at $\|D\mu\|_2^2$, for example. Further tuning of parameters and weights did not lead to essential improvements compared to the shown result with its decreasing step norms, oscillating function F_1 and increasing norm $\|D\mu\|_2^2$ starting at a certain iteration.

3.7 Conclusion from Joint Optimization of Reconstruction and Motion

The primary purpose of joining the reconstruction and motion joint problem is to take advantage of possible synergy effects. Those are only observable if the joint approaches are at least capable of reproducing the reference result computed by a separate image recovery followed by a motion estimation, afterwards. All three investigated methods, weak coupling (section 3.3), scaled ADMM (section 3.4) and PPXA (section 3.5) fail in this regard.

Whereas the weak convergence of the weak coupling algorithm seems to prevent any kind of convergence, the scaled ADMM result suffers from flickering at subsequent frames which remains even after more than 100000 iterations. ADMM being known for its very slow convergence to high accuracy [BPC+10] is one possible explanation. However, the reconstruction of the time boundary images is perfect. Since both methods utilize the Chambolle-Pock algorithm as an inner method for managing the first alternating minimization step, the computational effort and, consequently, the running time of one iteration is comparably slow. None of the two approaches yields a result similar to the reference, especially considering the motion.

On the other hand a true joint optimization can be observed at the PPXA iterations. By adjusting parameters appropriately, priorities can be shifted between reconstruction and motion terms, and an improvement in the according variables can be observed. However, in comparison to the reference solution only the reconstruction quality can benefit significantly. With higher priority on the motion, results have wrong flow directions based on unfinished reconstructions. Assigning equal weights seems to be a good trade-off in order to obtain a better recovery and motion estimation performance. Nevertheless, results similar to the reference solution cannot be achieved since the similarity starts to decline from a certain number of iterations onwards for all considered parameter combinations. Again, a reason for that is the weak convergence of PPXA which is insufficient for high accuracy.

Another explanation for all approaches not resulting in the reference solution are inappropriate physical constraints imposed to the motion. The reference solution already shows the pixel intensities or mass, respectively, diverge from a nonzero pixel to the surrounding reconcentrating at the end. At frames in between the time boundary images a diffuse mass cloud can be observed rather than separate particles which are the requirement for image recovery. This seems to render the methods used for the reconstruction and the motion estimation incompatible to each other. Ideas how this problem could be handled, but which are not further investigated here, are described in section 3.7.1.

Summing up, in combination with methods used, the straight forward additive joining of objects functions from mass recovery and motion estimation does not lead to satisfactory results which allow to demonstrate synergy effects.

3.7.1 Incompressibility Constraints as Generalization

A way to suppress mixing of pixel intensities or mass, respectively, in the intermediate frames between the time boundary images of the motion part may be additional physical constraints. One option is to impose an incompressibility constraint such as the vanishing divergence of the flow velocity

$$\nabla \cdot v = 0 \tag{3.7.1}$$

is known to be. Benamou and Brenier themself suggest in their paper [BB00] to use the Euler Equations for an ideal incompressible fluid. These equations are (3.7.1) and additionally

$$\partial_t v + v \cdot \nabla v = -\nabla p \tag{3.7.2}$$

where $p(t, x)$ is the pressure field. However, they do not follow up on that path. Instead, together with Guittet [BBG04] they essentially multiply density and velocity variables so that several instances of the same variables exists which they call “phases”. They try to achieve incompressibility by coupling the different phases to a mutual constraint.

A different common approach to enforce incompressibility is to view the Monge-Kantorovich problem 2.6.2 in terms of geodesics i.e. absolute continuous curves in a space of probability measures. This space can be extended to e.g. the set of doubly stochastic probability measures in a space representing the set of all Borel measures with a certain property which, then, is the configuration space of incompressible fluids. Brenier [Bre99, Bre03] and others as well, like Ambrosio and Figalli [AF07, AF08] take this path whereas Loeper [Loe06] approximates the Euler equations (3.7.2) with the help of “semi-geostrophic equations”.

4 Compressed Motion Sensing

The topic of this section is the development of an extension to the classical compressed sensing framework (section 2.5) for the recovery of time-varying signals. In particular, the temporal variation of indirectly measured intensity distributions is considered that can be interpreted as motion. This is the motivation for calling the approach *compressed motion sensing* or in short *CMS*.

Assume a signal undergoes a change over time having states $x \in \mathbb{R}^n$ and $y \in \mathbb{R}^n$ at two subsequent points in time which can be described by a transformation $T : \mathbb{R}^n \to \mathbb{R}^n$ such that $T(x) = y$. Observations $b_x \in \mathbb{R}^m$ and $b_y \in \mathbb{R}^m$ with $m < n$ are available for x and y, respectively, both acquired by the same sensor $A \in \mathbb{R}^{m \times n}$. The two separate recovery problems $Ax = b_x$ and $Ay = AT(x) = b_y$ then lead to the joint CMS problem

$$\begin{bmatrix} A & 0 \\ 0 & A \end{bmatrix} \begin{bmatrix} x \\ T(x) \end{bmatrix} = \begin{bmatrix} b_x \\ b_y \end{bmatrix} .$$

In the following the analysis is restricted to linear CMS, meaning $y = Tx$ originates from x through a linear transformation realized by a matrix $T \in \mathbb{R}^{n \times n}$ so that

$$\begin{bmatrix} A \\ AT \end{bmatrix} x = \begin{bmatrix} b_x \\ b_y \end{bmatrix} . \tag{4.0.1}$$

It is worth to mention that in the case of constant change this formulation can easily be extended to more than two points in time by just appending AT^2, AT^3, etc. to the sensor and the corresponding observations to the right hand side. Independent from the number of time points, only the first signal status x is being recovered and the subsequent once can easily be calculated as $T^k x$ where k is the discrete time step. If the transformation T is varying with each time step which is assumed here, (4.0.1) should be solved pairwise for consecutive frames.

The main difference to classical compressed sensing models is the presence of the transformation T which is unknown in general. One can say CMS is compressed sensing with additional observations b_y and a second but unknown sensor AT. This perspective motivates to investigate the joint estimation of the recovered signal x and transformation T and to explore possible synergy effects.

A typical application scenario is the reconstruction of mass distributions and their movement, represented as nonnegative values on a regular grid. The following investigations are based on binary $D = 2$ and $D = 3$-dimensional images $x \in \{0, 1\}^n$ showing $s \in \mathbb{N}$ particles and having a pixel or voxel grid of size $d \in \mathbb{N}$ in each dimension, such that $n = d^D$. Some synthetic scenarios with known ground truth are created by placing the particles in the middle of the grid cells unless stated otherwise, i.e. 0 and 1 indicates the absence and presence of a particle in a cell, respectively.

For the first considerations the investigation is restricted to a very simple scenario: Each particle occupies exactly one cell and general linear transformations are assumed to be permutations, i.e. $T = P \in \mathcal{P}_n$ such that

$$Bx = b \qquad \text{with} \qquad B := \begin{bmatrix} A \\ AP \end{bmatrix} \in \mathbb{R}^{2m \times n} \quad \text{and} \quad b := \begin{bmatrix} b_x \\ b_y \end{bmatrix} \in \mathbb{R}^{2m} \ . \tag{4.0.2}$$

B is called the *CMS sensor* and $Bx = b$ the *CMS system*.

Typical questions regarding sparsity dependent recoverability of the signal assuming the permutation P is known are addressed in section 4.1. Hereby, the CMS system reduces to an ordinary underdetermined linear equation which is investigated from the viewpoint of compressed sensing. Thereafter, section 4.2 deals with the additional estimation of the motion in form of matrix P including extensions to more general signals.

4.1 Signal Recovery

Throughout this section it is assumed that the permutation matrix P of the CMS system (4.0.2) is known unless stated otherwise.

Obviously, $Ax = b_x$ is the part of the CMS system which is commonly used in order to recover the signal x. The lower half consists of a second sensor, i.e. the permuted sensor AP, and additional m observations b_y available for determining x. Hence, recovery quality as predicted by CS must be at least as good as considering the first equation system only, see section 2.5. However, doubling of information is no guarantee for a recovery quality gain. If there is no motion, meaning the permutation is the identity, $P = I_n$, then the two systems $Ax = b_x$ and $APx = Ax = b_y$ coincide in cases of no measurement errors. In other words the maximum rank of the CMS Sensor $B \in \mathbb{R}^{2m \times n}$ is merely m depending on the rank of A, since $\operatorname{rank}(A) = \operatorname{rank}(AP)$. Hence, there cannot be any improvement and no gain from using CMS if $P = I_n$.

It is not as clear whether the upper bound for $\operatorname{rank}(B)$ is $2m$ or less. This is the subject of section 4.1.1 followed by sparsity boundary estimates and recovery guarantees in section 4.1.2 and experiments in section 4.1.3.

4.1.1 Sensor Rank

Naturally, the higher the rank of the CMS sensor B is, the better is the recovery of the signal x. Clearly, the rank is dependent on both the sensor A and the permutation

Table 4.1 - Cycle structure of permutations leading to invertible (*good permutations*) and singular (*bad permutations*) CMS sensors $B = \begin{bmatrix} A \\ AP \end{bmatrix} \in \mathbb{R}^{2m \times 2m}$ represented by the matrix $P \in \mathcal{P}_{2m}$. The sensor $A \in \mathbb{R}^{m \times 2m}$ is chosen to be a Gaussian random matrix. The different permutation structures are ordered by the number of fixed points for each m. Each table cell shows the cycle structure of the remaining elements which are no fixed points in the form l_1-$\ldots$-l_c with $l_j \in \{2, \ldots, 2m\}$ for all $j \in [c]$ meaning a permutation with c cycles having lengths $l_1, \ldots, l_c$. The shortcuts "all" and "id" mean all cycle combinations regarding the corresponding number of fixed points, whereas "no" or "-" stands for no permutation and no permutation exists with the corresponding number of fixed points, respectively. Empty columns with no permutations are not shown. The ratio of good permutations with respect to all possible permutations is 50%, 70.8333%, 85.9722%, 94.2684% and 97.9707% for $m = 1$, 2, 3, 4 and 5.

	Cycle structure of good permutation					Cycle structure of bad permutation							
	fixed points					fixed points							
m	0	1	2	3	4	2	3	4	5	6	7	8	10
1	all	-	no	-	-	id	-	-	-	-	-	-	-
2	all	all	no	-	no	all	-	id	-	-	-	-	-
3	all	all	4	no	no	2-2	all	all	-	id	-	-	-
4	all	all	6 4-2 3-3	5	no	2-2-2	3-2	all	all	all	-	id	-
5	all	all	8 6-2 5-3 4-4 4-2-2 3-3-2	7 5-2 4-3	6	2-2-2-2	3-2-2	4-2 3-3 2-2-2	all	all	all	all	id

matrix P. The basic question addressed next is how the permutation P influences rank(B). Therefore, the investigation starts with the case $n = 2m$ leading to a square CMS sensor $B \in \mathbb{R}^{2m \times 2m}$.

In order to get a first idea an initial experiment is performed where the sensor $A \in \mathbb{R}^{m \times 2m}$ is chosen as a Gaussian random matrix with normalized columns having maximum rank. These matrices are known to be the best performing sensors regarding recovery quality as described in section 2.5. Afterwards, rank(B) is calculated for every possible permutation matrix P where the investigation is restricted to the cases $m = 1, \ldots, 5$. In this context, permutations leading to an invertible CMS sensor are called *good* whereas the *bad* ones yield a singular CMS sensor. The result is summarized in table 4.1.

Adopting the notation from section 2.3, the CMS sensor B is always singular if $f \geq m$ where f is the number of fixed points of the permutation. Invertibility is clearly given in cases of a permutation with $f \leq 1$ fixed points whose number converges from above to

$$\lim_{m\to\infty} \frac{!(2m) + 2m\,(!(2m-1))}{(2m)!} = \frac{2}{e} \approx 73.58\%$$

due to lemma 2.11, so that the majority of permutations is good for $m > 2$. Interestingly, in the range of $f \in \{2, \dots, m-1\}$, the specific permutation cycle structure determines whether the CMS sensor is invertible. It seems like fewer but longer cycles are beneficial for invertibility. The ratio of the overall number of good permutations with respect to all possible permutation matrices of the same size grows with increasing m: 50%, 70.8333%, 85.9722%, 94.2684% and 97.9707%. Running the experiment using different Gaussian random matrices yields the very same result each time. Likewise, analyzing few randomly picked sensors of larger dimensions confirms the impression of exclusive dependency on the permutations cycle structure.

Singularity occurring in the cases when $f \geq m$ can be shown easily.

Lemma 4.1 *(singularity of the CMS sensor for $f \geq m$)*:
Let $A \in \mathbb{R}^{m\times 2m}$ be an arbitrary matrix. Then, the CMS sensor $B \in \mathbb{R}^{2m\times 2m}$ is singular if the permutation induced by $P \in \mathcal{P}_{2m}$ has $f \geq m$ fixed points.

Proof. Assume $A = \begin{bmatrix} L & R \end{bmatrix}$ where $L, R \in \mathbb{R}^{m\times m}$, and assume the permutation induced by P has $f \geq m$ fixed points, so that w.l.o.g.

$$B = \begin{bmatrix} A \\ AP \end{bmatrix} = \begin{bmatrix} L & R \\ L & \pi(R) \end{bmatrix}$$

for a permutation π acting on the columns of R. Using the Schur complement, B is singular if and only if $\pi(R) - LL^{-1}R = \pi(R) - R$ is singular. That is the case, since $(\pi(R) - R)\mathbb{1} = 0$ meaning $\mathbb{1} \in \mathcal{N}(\pi(R) - R)$. □

For showing the invertibility of the CMS sensor in general, further constraints to the sensor matrix A are necessary, since zero columns, for example, would heavily influence rank(A) and, consequently, rank(B). The randomness of A seems to play a crucial rule in this regard, since it is easy to derive examples leading to a singular CMS sensor for every single permutation as the following examples illustrate.

Example 4.2 *(permutation with cycles of length 2 leading to a singular CMS sensor)*:
For $A = \begin{bmatrix} 1 & 0 & 1 & -1 \\ 0 & 1 & 1 & 3 \end{bmatrix}$ the permutation $p = \{(2\,1)\,(4\,3)\}$ yields a singular CMS sensor with rank(B) = 3 which is

$$B = \begin{bmatrix} 1 & 0 & 1 & -1 \\ 0 & 1 & 1 & 3 \\ 0 & 1 & -1 & 1 \\ 1 & 0 & 3 & 1 \end{bmatrix}.$$

Permutation p is considered good based on evaluation with a number of randomly chosen matrices $A \in \mathbb{R}^{2\times 4}$.

Example 4.3 *(permutations consisting of a single cycle leading to singular CMS sensors)*:
For $A = \begin{bmatrix} 2 & 0 & 1 & 4 \\ 0 & -2 & 1 & 5 \end{bmatrix}$ the permutations $p = \{(4\,1\,2\,3)\}$ and $p' = \{(2\,3\,4\,1)\}$ both consist of one single cycle and result in singular CMS sensors

$$B = \begin{bmatrix} 2 & 0 & 1 & 4 \\ 0 & -2 & 1 & 5 \\ 4 & 2 & 0 & 1 \\ 5 & 0 & -2 & 1 \end{bmatrix} \qquad \text{and} \qquad B' = \begin{bmatrix} 2 & 0 & 1 & 4 \\ 0 & -2 & 1 & 5 \\ 0 & 1 & 4 & 2 \\ -2 & 1 & 5 & 0 \end{bmatrix} ,$$

respectively, where both have $\operatorname{rank}(B) = \operatorname{rank}(B') = 3$. These are two permutations considered good after evaluation with randomly chosen matrices $A \in \mathbb{R}^{2\times 4}$.

Example 4.2 and 4.3 show that even if $\operatorname{spark}(A) > m$, there is no guarantee for the invertibility of the CMS sensor. However, it appears to be rather unlikely that for a randomly picked Gaussian matrix A, a basically good permutation turns out to be a bad one for B. In this regard, the set of matrices behaving different than a Gaussian random matrix $A \in \mathbb{R}^{m\times 2m}$ seems to be small compared to the set of all real $m \times 2m$ matrices. This leads to conjecture 4.4 which covers all good permutations with the determined cycle structure of the initial experiment (table 4.1).

Conjecture 4.4 *(invertibility of the CMS sensor)*:
Let $A \in \mathbb{R}^{m\times 2m}$ be a Gaussian random matrix. Let the permutation represented by matrix $P \in \mathcal{P}_{2m}$ have $k \in \mathbb{N}$ cycles, where each fixed point counts as one cycle. If

$$k \leq m \;, \tag{4.1.1}$$

then the CMS sensor $B = \begin{bmatrix} A \\ AP \end{bmatrix} \in \mathbb{R}^{2m\times 2m}$ is almost surely invertible.

The following corollary generalizes this result to arbitrary dimensions.

Corollary 4.5 *(extension of invertibility condition to nonsquare CMS sensors)*:
Let conjecture 4.4 be true and $A \in \mathbb{R}^{m\times n}$ be a Gaussian random matrix. Let the permutation represented by matrix $P \in \mathcal{P}_{2m}$ have $k \in \mathbb{N}$ cycles, where each fixed point counts as one cycle. If

$$k \leq n - m \;, \tag{4.1.2}$$

then the CMS sensor $B = \begin{bmatrix} A \\ AP \end{bmatrix} \in \mathbb{R}^{2m\times n}$ has almost surely $\operatorname{rank}(B) = 2m$.

Proof. W.l.o.g. let the first $2m$ columns of P induce m or less cycles. Applying conjecture 4.4 on the first $2m$ columns of B reveals that they are almost surely linearly independent. Hence B has maximum rank. □

In summary, maximum recovery gain can be expect from using the CMS sensor if the permutation has sufficiently few cycles given conjecture 4.4 is true. This impression is further consolidated in sections 4.1.1.1 and 4.1.1.2 which offer a different viewpoint on how the CMS sensor rank properties derive, including the counterpart to conjecture 4.4 being proven. In the following the inequalities (4.1.1) or (4.1.2), respectively, are referred to as *cycle criterion*. Section 4.1.1.3 gives an probabilistic estimation on when it can be expected to be fulfilled.

4.1.1.1 Permutation Eigenspace

Deeper insights into the properties of the CMS sensor can be obtained from investigating its image. Corollary 2.14 implies that a permutation matrix $P \in \mathcal{P}_n$ can only have two different real eigenvalues, i.e. 1 and -1, and real eigenvectors depend on their algebraic multiplicity. Eigenvalue $\varphi_{l_j}^{l_j} = 1$ for $j \in [k]$ has algebraic multiplicity k whereas all other eigenvalues $\varphi_{l_j}^{i}$ with $i < l_j$ have a smaller multiplicity as long as there is a cycle length which is not a multiple of the shortest cycle length $\min_{j \in [k]} l_j$. An exception to this is the case when all cycles have the shortest cycle length 2 as common divisor. Then, both eigenvalues, 1 and -1, have the same multiplicity and, thus, there exists an equal number of real eigenvectors. In general all eigenvectors form an orthogonal basis and have a clear structure.

Lemma 4.6 *(eigenvector basis obtained from a permutation)*:
The eigenvectors of a permutation matrix $P \in \mathcal{P}_n$ with cycle structure (2.3.1) form an orthogonal basis U of $\mathbb{C}^n$ which is

$$U = \bigcup_{j \in [k]} U_j \quad \text{with} \quad U_j = \left\{ u \in \mathbb{C}^n \;\middle|\; u_{[n] \setminus C_j} = 0,\; u_{C_j} = \begin{bmatrix} \varphi^1 \\ \vdots \\ \varphi^{l_j} \end{bmatrix},\; \varphi \in \Phi_j \right\} \text{,} \tag{4.1.3}$$

where $\Phi_j = \left\{ \varphi_{l_j}^{i} \;\middle|\; i \in [l_j] \right\}$ is the set of different powers of an arbitrary l_j-th primitive root of unity $\varphi_{l_j} \in \mathbb{C}$.

Proof. Since P is normal, its eigenvectors form an orthogonal basis U of $\mathbb{C}^n$. For each $j \in [k]$, lemma 2.14 implies that the eigenvalues contained in the set Φ_j originate from cycle C_j of length l_j. Let $u \in \mathbb{C}^n$ with $u_{C_j} = \left[\varphi^1, \ldots, \varphi^{l_j}\right]^\top$ for an arbitrary $\varphi \in \Phi_j$ and all other entries $u_{[n] \setminus C_j} = 0$. It holds

$$(Pu)_{C_j} = \left[\varphi^{l_j}, \varphi^1, \ldots, \varphi^{l_j - 1}\right]^\top = \left(\varphi^{-1} u\right)_{C_j} .$$

Thus, u is an eigenvector to the eigenvalue $\varphi^{-1} = \varphi^{l_j - 1}$ and, consequently, U_j contains all eigenvectors corresponding to the eigenvalues in Φ_j. □

In particular, the previous proof shows that the eigenvector corresponding to the eigenvalue $\varphi_{l_j}^{l_j} = 1$ is $u \in \mathbb{R}^n$ with $u_{[n] \setminus C_j} = 0$ and $u_{C_j} = 1$ for each $j \in [k]$, which play a special role in a stricter version of lemma 4.1 covering all permutations considered as bad in the initial experiment (table 4.1).

Theorem 4.7 *(singularity of the CMS sensor)*:
Let $A \in \mathbb{R}^{m \times 2m}$ and let the permutation represented by matrix $P \in \mathcal{P}_{2m}$ have $k \in \mathbb{N}$ cycles, where each fixed point counts as one cycle. If

$$k > m \ ,$$

then the CMS matrix $B = \begin{bmatrix} A \\ AP \end{bmatrix} \in \mathbb{R}^{2m \times 2m}$ is singular.

Proof. Consider the mapping of matrix B from the eigenspace of P induced by eigenvalue 1 to the corresponding image, i.e. $B : E_1 \to H_1$ where

$$E_1 := \left\{ x \in \mathbb{R}^{2m} \,\middle|\, Px = x \right\}$$

and

$$H_1 := \left\{ \begin{bmatrix} y \\ y \end{bmatrix} \,\middle|\, y \in \mathbb{R}^m \right\} = \{ x \in \mathbb{R}^{2m} \mid x_i = x_{i+m} \text{ for } i \in [m] \} \ . \tag{4.1.4}$$

Let k be the multiplicity of the eigenvalue 1 of P, then $\dim(E_1) = k$ with regard to (2.3.4) and the corresponding eigenvectors $\left\{ u \in \mathbb{R}^n \,\middle|\, u_{[n] \setminus C_j} = 0, \ u_{C_j} = 1, \ j \in [k] \right\}$. Furthermore, $\dim(H_1) < m$. For any arbitrary square matrix B it holds, that $\operatorname{rank}(B) = 2m$ if and only if $\dim\{Bx \mid x \in S\} = \dim(S)$ for every linear subspace $S \subseteq \mathbb{R}^{2m}$. Since E_1 is a linear subspace of $\mathbb{R}^{2m}$, it follows that, if $k = \dim(E_1) > m > \dim(H_1)$, then $\operatorname{rank}(B) < 2m$. Hence, the CMS matrix B is singular, if the number of cycles k exceeds m. □

In other words the basis vectors in U according to (4.1.3) corresponding to eigenvalue 1 are all mapped to H_1 which has a dimension not exceeding m. Consequently, if there are $m + 1$ or more vectors in U corresponding to eigenvalue 1, the image of the remaining $m - 1$ or less basis vectors under B cannot cover the remaining space of $\mathbb{R}^{2m}$ since they are too few in order to span the complement $\mathbb{R}^{2m} \setminus H_1$ having dimension m, as well. In the case of eigenvalue 1 with multiplicity m or less, the remaining eigenvectors in U have to span $\mathbb{R}^{2m} \setminus H_1$ for an invertible CMS sensor. Those other eigenvectors are mapped to complex spaces

$$\begin{aligned} H_\psi :=& \left\{ \begin{bmatrix} y \\ \psi y \end{bmatrix} \,\middle|\, y \in \mathbb{R}^m, \ \psi \in \mathbb{C}, \ \|\psi\|_2 = 1 \right\} \\ =& \{ x \in \mathbb{C}^{2m} \mid \psi \in \mathbb{C}, \ \|\psi\|_2 = 1, \ x_i \in \mathbb{R}^m, \ x_i = \psi x_{i+m} \text{ for } i \in [m] \} \end{aligned} \tag{4.1.5}$$

with $\psi \in \mathbb{C}$. Every H_ψ for a fixed ψ is a half space of $\mathbb{C}^{2m}$, since $\dim(H_\psi) = m$ and $H_{\psi_1} \cap H_{\psi_2} = \{0\}$ for different $\psi_1, \psi_2 \in \mathbb{C}$, obviously. Thus, any set of $2m$ vectors with no more than m elements from a set of the form (4.1.5) with the same ψ span the entire space $\mathbb{C}^{2m}$, but only H_1 and H_{-1} form an m-dimensional half space of R^{2m}. This is the reason why all permutations consisting of m cycles of length 2 yield an invertible CMS sensor B in the initial experiment (table 4.1). Example 4.2 shows once more that this is not the case for every arbitrary matrix A. Theorem 4.7 can be naturally extended to nonsquare CMS sensors.

Corollary 4.8 *(extension of singularity condition to nonsquare CMS sensors)*:
Let $A \in \mathbb{R}^{m\times n}$ and let the permutation represented by matrix $P \in \mathcal{P}_n$ have $k \in \mathbb{N}$ cycles, where each fixed point counts as one cycle. If

$$k > n - m \ ,$$

then the CMS sensor $B = \begin{bmatrix} A \\ AP \end{bmatrix} \in \mathbb{R}^{2m\times n}$ has $\operatorname{rank}(B) < 2m$.

Proof. W.l.o.g. let the first $2m$ columns of P induce $m+1$ or more cycles. Applying theorem 4.7 on the first $2m$ columns of B reveals that they do not have maximum rank $2m$. The remaining $n-2m$ column vectors do not add to $\operatorname{rank}(B)$ since their indexes are fixed points under the permutation induced by P such that their image is in the half space H_1 defined by (4.1.4) which is already entirely spanned by the first $2m$ columns of B. Hence $\operatorname{rank}(B) < 2m$. □

In summary this makes the condition of the permutation $P \in \mathcal{P}_n$ having

$$k \le n - m \tag{4.1.6}$$

disjoint cycles a necessary condition for an invertible CMS sensor $B = \begin{bmatrix} A \\ AP \end{bmatrix} \in \mathbb{R}^{2m\times n}$ with $A \in \mathbb{R}^{m\times n}$. In the following it is called the *maximum rank condition*.

4.1.1.2 Nullspace Angles

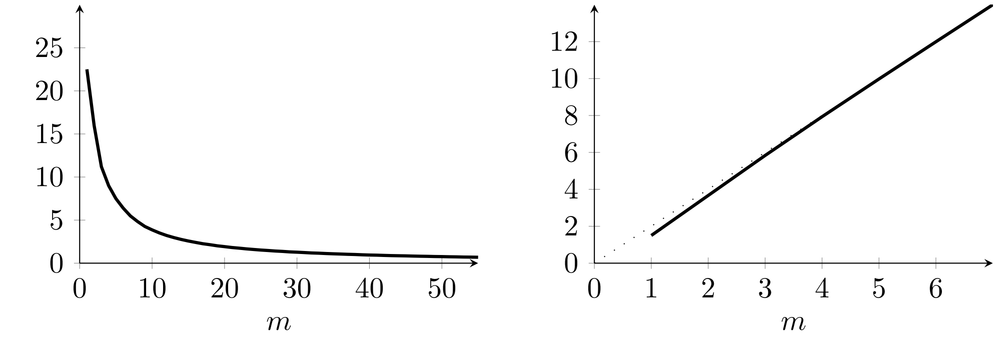
The invertibility of a square CMS sensor is strongly related to the properties of the two nullspaces $\mathcal{N}(A)$ and $\mathcal{N}(AP)$. An invertible CMS sensor $B \in \mathbb{R}^{2m\times 2m}$ means a vanishing nullspace

$$\operatorname{rank}(B) = 2m \qquad \Leftrightarrow \qquad \mathcal{N}(B) = \{0\} \qquad \Leftrightarrow \qquad \mathcal{N}(A) \cap \mathcal{N}(AP) = \{0\} \ .$$

Due to lemma 2.4 this is the case if the smallest principal angle $\theta(\mathcal{N}(A), \mathcal{N}(AP))$ is positive as described in section 2.1. It can easily be derived with the help of definition 2.7 introducing the polar decomposition without computing the nullspaces first.

Lemma 4.9 *(smallest principal angle between $\mathcal{N}(A)$ and $\mathcal{N}(AP)$)*:
Let $A \in \mathbb{R}^{m\times n}$ and $P \in \mathcal{P}_n$. Moreover, let L be the matrix whose normalized columns form an orthogonal basis of the image of $A^\top$, denoted by $\mathcal{R}(A^\top)$. The smallest principal angle between the nullspaces $\mathcal{N}(A)$ and $\mathcal{N}(AP)$ is

$$\theta(\mathcal{N}(A), \mathcal{N}(AP)) = \arccos\left(\sigma_{\max}\left(L^\top P^\top L\right)\right) \ .$$



(a) Mean smallest principal angle $\theta(\mathcal{N}(A), \mathcal{N}(AP))$ in degree.

(b) Mean $\operatorname{rank}(B)$. For comparison the maximum possible rank $2m$ is plotted as dotted line.

Figure 4.1 - Mean smallest principal angle between the nullspaces $\mathcal{N}(A)$ and $\mathcal{N}(AP)$ and the mean rank of the corresponding CMS sensor $B = \begin{bmatrix} A \\ AP \end{bmatrix} \in \mathbb{R}^{2m \times 2m}$ over 50000 different Gaussian random matrices $A \in \mathbb{R}^{m \times 2m}$ and permutation matrices $P \in \mathcal{P}_{2m}$ for each m. $\theta(\mathcal{N}(A), \mathcal{N}(AP))$ is converging to zero whereas $\operatorname{rank}(B)$ asymptotically approaches the maximum possible rank for increasing m rather fast.

Proof. Since $\dim(A^\top) = \dim((AP)^\top) = \dim(A^\top P^\top)$, it holds

$$P^\top A^\top = L_{AP} R_{AP}^{\frac{1}{2}} = P^\top A^\top (APP^\top A^\top)^{-\frac{1}{2}} (APP^\top A^\top)^{\frac{1}{2}} = P^\top L_A R_A^{\frac{1}{2}}$$

where L_X and R_X are the matrices with $X^\top = L_X R_X^{\frac{1}{2}}$ obtained from the polar decomposition (2.2.1) for $X = A$ and $X = AP$, respectively. Thus, the columns of $P^\top L$ with $L = L_A$ form an orthogonal basis for $\mathcal{R}(P^\top A^\top) = \mathcal{R}((AP)^\top)$. Due to $\mathcal{N}(X)^\perp = \mathcal{R}(X^\top)$ it follows

$$\begin{aligned} \theta(\mathcal{N}(A), \mathcal{N}(AP)) =& \theta\left(\mathcal{N}(A)^\perp, \mathcal{N}(AP))^\perp\right) \\ =& \theta\left(\mathcal{R}(A^\top), \mathcal{R}((AP)^\top)\right) = \arccos\left(\sigma_{\max}\left(L^\top P^\top L\right)\right) \ . \end{aligned}$$

□

The behavior of $\theta(\mathcal{N}(A), \mathcal{N}(AP))$ is evaluated in an experiment with 50000 different Gaussian random matrices A and permutations P for each $m \in [55]$. In addition, the rank of the corresponding CMS sensor is computed. The results shown in figure 4.1 indicate a declining smallest principal angle converging to zero for increasing m and a CMS sensor rank which is the maximum possible for sufficiently large m. For $m \geq 12$, all 50000 different CMS sensors have maximum rank $2m$. Once more, this solidifies the impression of conjecture 4.4 being correct.

4.1.1.3 Maximum Rank Condition Probability

A permutation $p \in \mathcal{S}_{2m}$ is considered *good* if for its number of cycles holds that $k \leq m$ (maximum rank condition (4.1.6)). This sections answers the question about the probability of a random permutation being good. A normalized random variable in relation to the number of cycles k can be defined by

$$K_n := \frac{k - \log(n)}{\sqrt{\log(n)}} \ .$$

According to [SL66] it is asymptotically normally distributed, that is

$$\lim_{n\to\infty} \mathbb{P}(K_n \leq \kappa) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\kappa} e^{-\frac{z^2}{2}} \,\mathrm{d}z \quad \text{for} \quad \kappa \in \mathbb{R}, \tag{4.1.7}$$

whereas by Chebyshev's inequality [HMRAR13], it can be obtained for any n

$$\mathbb{P}(K_n \geq \kappa) \leq \frac{1}{\kappa^2} \ .$$

The following lemma is estimating the probability for a random perturbation to satisfy the maximum rank condition.

Lemma 4.10 *(maximum rank condition probability)*:
Let $p \in \mathcal{S}_n$ be a random permutation drawn uniformly from $\mathcal{S}_n$. Then, for any $m \geq 3$, the maximum rank condition (4.1.6) for $n = 2m$ is fulfilled with probability at least

$$\mathbb{P}(k \leq m) \geq 1 - \frac{\sqrt{2\log(n)}}{\sqrt{\pi}(n - 2\log(n))} \exp\left(-\frac{(n-2\log(n))^2}{8\log(n)}\right) \tag{4.1.8}$$

whereas

$$\mathbb{P}(k \leq m) = 1$$

holds asymptotically for $n \to \infty$.

Proof. Applying (4.1.7) and the tail bound [FR13, Lemma C.7] yields

$$\lim_{n\to\infty} \mathbb{P}(K_n \leq \kappa) = 1 - \frac{1}{\sqrt{2\pi}} \int_{\kappa}^{\infty} e^{-\frac{z^2}{2}} \,\mathrm{d}z \geq 1 - \frac{1}{\sqrt{2\pi}} \min\left\{\sqrt{\tfrac{\pi}{2}}, \tfrac{1}{\kappa}\right\} \exp\left(-\frac{\kappa^2}{2}\right) \ .$$

for $\kappa > 0$. By substitution of

$$m = \frac{n}{2} := \kappa\sqrt{\log(n)} + \log(n) \quad \Leftrightarrow \quad \kappa = \frac{n - 2\log(n)}{2\sqrt{\log(n)}} \ ,$$

it can be obtain $0 < \frac{1}{\kappa} < \sqrt{\frac{\pi}{2}}$ for $n \geq 6$ and consequently

$$\mathbb{P}(c + f \leq m) \geq \mathbb{P}(K_n \leq \kappa) \geq 1 - \frac{\sqrt{2\log(n)}}{\sqrt{\pi}(n - 2\log(n))} \exp\left(-\frac{(n-2\log(n))^2}{8\log(n)}\right) \ .$$

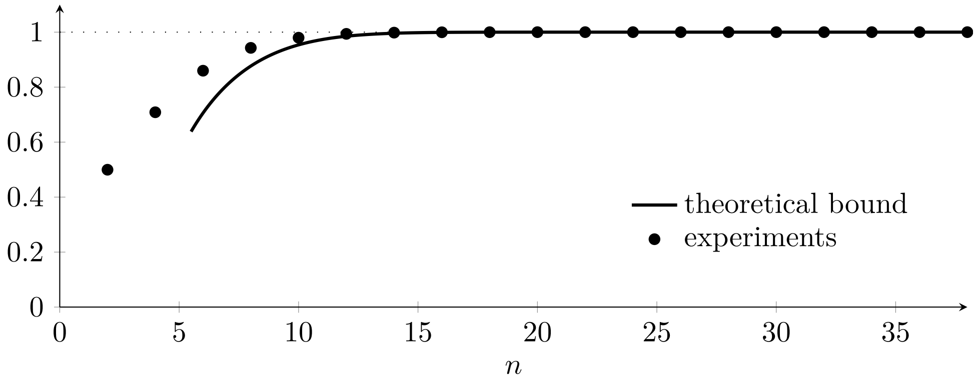


Figure 4.2 - Lower probability bound (4.1.8) of a random permutation $p \in \mathcal{P}_n$ drawn uniformly from $\mathcal{S}_n$ having $\frac{n}{2}$ or less cycles k, $\mathbb{P}\left(k \leq \frac{n}{2}\right)$, together with the experimental validation. This is the minimum probability for a fulfilled maximum rank condition (4.1.6) of a CMS sensor with $n = 2m$. Experimentally, many uniformly drawn random permutation matrices of different sizes n are generated, and the relative numbers of permutations with $k \leq \frac{n}{2}$ confirm the theoretical result. Since the bound quickly approaches 1 with increasing n, the maximum rank condition holds with very high probability for interesting problem sizes n.

Regarding the asymptotic case it holds

$$\kappa = \frac{n - 2\log(n)}{2\sqrt{\log(n)}} \geq -\frac{1}{10^4}n^2 + \frac{9}{40}n - \frac{1}{2} \quad \text{for} \quad n \geq 2 \ ,$$

and hence for increasing n

$$\begin{aligned}\mathbb{P}\left(c + f \leq m\right) &\geq \mathbb{P}(K_n \leq \kappa) \\ &= 1 - \frac{1}{\sqrt{2\pi}(-\frac{1}{10^4}n^2 + \frac{9}{40}n - \frac{1}{2})} \exp\left(-\frac{1}{2}\Big(\frac{1}{10^4}n^2 - \frac{9}{40}n + \frac{1}{2}\Big)^2\right) \\ &\xrightarrow{n\to+\infty} 1 \ .\end{aligned}$$

□

Figure 4.2 illustrates the behavior of the lower bound derived in lemma 4.10, graphically, together with an experimental validation. In this experiment 10^6 random permutation matrices are generated drawn uniformly from $\mathcal{S}_n$ with n satisfying $n = 2m$ for the considered range $n \in [40]$. The relative number of permutations satisfying the maximum rank condition (4.1.6) gives an estimate of the desired probability. The result confirms the derived bound very clearly. Thus, it is save to say that the maximum rank condition holds with very high probability for interesting problem sizes n.

4.1.2 Sparsity and Recovery Performance

The previous sections only investigate the invertibility of the CMS sensor (4.0.2) that guarantees exact recovery of *any* vector x disregarding its sparsity. In the following the sparsity of x is assumed, in addition.

Motivated by application-oriented scenarios, the investigation is limited to tomographic recovery problems. In the following, the results from section 2.5.4 are extended to the CMS sensor. The randomness assumption of sensor A is dropped and the deterministic sensor A_d^D with $D = 2$ or $D = 3$ from (2.5.26) is considered instead. As already described, this corresponds to a tomographic projection matrix from D orthogonal projections. This kind of sensors does not have maximum rank m being discussed in section 2.5.4.4. Furthermore, a binary signal $x \in \{0, 1\}^n$ is assumed here resulting in observations $b \in \mathbb{N}_0^{2m}$ when acquired by the CMS sensor

$$B_d^D = \begin{bmatrix} A_d^D \\ A_d^D P \end{bmatrix} \in \{0, 1\}^{2m \times n} \tag{4.1.9}$$

with permutation matrix $P \in \mathcal{P}_n$.

Recovery guarantees will be derived based on the tools from section 2.5.4.3. In particular, the reduced system (section 2.5.4.2) corresponding to the CMS system induced by an s-sparse signal is considered. In order to guarantee that reduced systems behave on average like the adjacency matrix of a well connected expander graph their dimensions will be constrained. To this end, the expected number of nonzero observation is derived first, based on the expected nonzero observation of $Ax = b_x$. In particular, it is shown that the later are equal to the expected nonzero observation of $APx = b_y$ and that b_y is also generated by random uniformly distributed s-sparse signal entries. The following lemma shows that an unknown permutation matrix P ensures the independence of y from x with

$$x, y \in \mathcal{X}_s^n := \{z \in \{0, 1\}^n \mid \|z\|_0 = s\}$$

where $\|z\|_0$ is the sparsity of a vector z defined by (2.5.1), that is the size of the support $\operatorname{supp}(z)$.

Lemma 4.11 *(uniform distribution of the signal recorded at a second time step)*: Let $X \in \mathcal{X}_s^n$ and $\Pi \in \mathcal{P}_n$ be independent and uniformly distributed random variables of s-sparse vectors and permutation matrices $x \in \mathcal{X}_s^n$ and $P \in \mathcal{P}_n$. Then $Y = \Pi X \in \mathcal{X}_s^n$ is uniformly distributed, too.

Proof. Let $x, y \in \mathcal{X}_s^n$ be any realizations. Then there are $s!(n-s)!$ different permutations mapping the support $\{i \mid x_i = 1\}$ to $\{i \mid y_i = 1\}$ and consequently $\{i \mid x_i = 0\}$ to $\{i \mid y_i = 0\}$ so that $y = Px$ for corresponding permutation matrices $P \in \mathcal{P}_n$. Let $\mathcal{P}_n(x, y)$ denote this set of permutation matrices. For any other realization $x \neq \tilde{x} \in \mathcal{X}_s^n$, the corresponding set $\mathcal{P}_n(\tilde{x}, y)$ exists and has the same cardinality. Furthermore, from $y = Px = \tilde{P}\tilde{x}$ it follows $P \neq \tilde{P}$ for $\tilde{P} \in \mathcal{P}_n(\tilde{x}, y)$ because otherwise $0 = P(x - \tilde{x})$ which contradicts $x \neq \tilde{x}$, i.e. x and $\tilde{x}$ have different supports. Hence, taking the independency assumption of X and Π into account, it holds

$$\begin{aligned}\mathbb{P}(y) &= \sum_{x\in\mathcal{X}_s^n}\ \sum_{P\in\mathcal{P}(x,y)} \mathbb{P}(x)\mathbb{P}(P) = \sum_{x\in\mathcal{X}_s^n}\ \sum_{P\in\mathcal{P}(x,y)} \frac{1}{\binom{n}{s}}\frac{1}{n!} \\ &= \binom{n}{s} s!(n-s)! \frac{1}{\binom{n}{s}}\frac{1}{n!} = \frac{1}{\binom{n}{s}} = \mathbb{P}(x)\ .\end{aligned}$$

□

This enables the derivation of the expected dimension of the reduced CMS system with

$$2m_{\text{red}} := |I^c| = |\{i \mid b_i > 0\}|$$

rows and

$$n_{\text{red}} := |J^c| = |\{j \mid B_{i,j} = 0,\ i \in I\}|$$

columns depending on sparsity s of $x \in \mathcal{X}_s^n$ where the sets I and J include the redundant row and column indexes similar to (2.5.28) for the CMS sensor B_d^D. Let M_{red} and N_{red} denote the random variables for m_{red} and n_{red}, respectively.

Lemma 4.12 *(expected number of rows in the reduced CMS system)*:
Let the s-sparse signal $x \in \mathcal{X}_s^n$ and permutation matrix $P \in \mathcal{P}_n$ be drawn uniformly random from $\mathcal{X}_s^n$ and $\mathcal{P}_n$, respectively. Then the expected number of rows after reducing the CMS system with sensor $B_d^D \in \{0,1\}^{2m\times n}$ is

$$\mathbb{E}(2M_{\text{red}}) = 2Dd^{D-1}\left(1 - \left(1 - \frac{1}{d^{D-1}}\right)^s\right)\ . \tag{4.1.11}$$

Proof. Lemma 4.11 shows that the distributions of x and $y = Px$ are the same, so both time specific observations $b_x \in \mathbb{N}_0^m$ and $b_y \in \mathbb{N}_0^m$ have expected number of nonzero entries $Dd^{D-1}\left(1 - \left(1 - \frac{1}{d^{D-1}}\right)^s\right)$ due to (2.5.29). Consequently, $b = \begin{bmatrix} b_x \\ b_y \end{bmatrix} \in \mathbb{N}_0^{2m}$ has twice as much. □

Figure 4.3a on page 95 shows an example of (4.1.11) for $D = 3$ which, naturally, converges to 1.

The expected value of a priori nonzero signal entries requires some more effort. The sensor matrix A_d^D has d nonzero entries in each row and D in each column, which reflects the fact that each projecting ray is incident to d cells of the grid and vice versa every cell is incident to D rays. Since sensor B_d^D is generated by concatenation, the number of nonzero entries in each column will be doubled. Thus, B_d^D can be interpreted as a notional sensor measuring every cell with $2D$ rays onto its $2D$ projections. Even though the geometric meaning gets lost, expressions like *ray* and *projection* in connection with the CMS sensor are still used below.

Lemma 4.13 *(expected number of columns in the reduced CMS system)*:
Let the s-sparse signal $x \in \mathcal{X}_s^n$ and permutation matrix $P \in \mathcal{P}_n$ be drawn uniformly

random from $\mathcal{X}_s^n$ and $\mathcal{P}_n$, respectively. Then the expected number of columns after reducing the CMS system with sensor $B_d^D \in \{0,1\}^{2m \times n}$ is

$$\mathbb{E}(N_{\text{red}}) = d^D \left(1 + \sum_{k=1}^{2D} (-1)^k \binom{2D}{k} \left(1 - \frac{k(d-1)+1}{d^D}\right)^s\right) . \tag{4.1.12}$$

Proof. The probability for a single grid cell to be incident to a specific ray is $\frac{d}{n} = \frac{1}{d^{D-1}}$. In general, the probability that a grid cell is incident to at least one out of $k \in [2D]$ different rays $r_1, \dots, r_k$ intersecting at one single cell (in the case of $k > 1$) is

$$q_{d,k} := \frac{k(d-1)+1}{d^D} .$$

Hence, the probability that $l = 0, \dots, s$ out of the s nonzero entries of x are incident to at least one of those rays is

$$\mathbb{P}\left(\sum_{i \in \bigcup_{r \in \{r_1,\dots,r_k\}} C_r} x_i = l\right) = \binom{s}{l} q_{d,k}^l p_{d,k}^{s-l} \qquad \text{with} \qquad p_{d,k} := 1 - q_{d,k} \tag{4.1.13}$$

where C_r is the set of cell indexes not being incident to the ray corresponding to the r-th sample. Let the random variable R_r for $r \in [2Dd^{D-1}]$ denote the event when the r-th sample is zero, i.e. $R_r = 1$ if $b_r = 0$ and $R_r = 0$ if $b_r > 0$, respectively. For $l = 0$, (4.1.13) gives the expected value

$$\mathbb{E}\left(\prod_{i=1}^k R_{r_i}\right) = \mathbb{P}\left(\prod_{i=1}^k R_{r_i} = 1\right) = p_{d,k}^s = \left(1 - \frac{k(d-1)+1}{d^D}\right)^s , \tag{4.1.14}$$

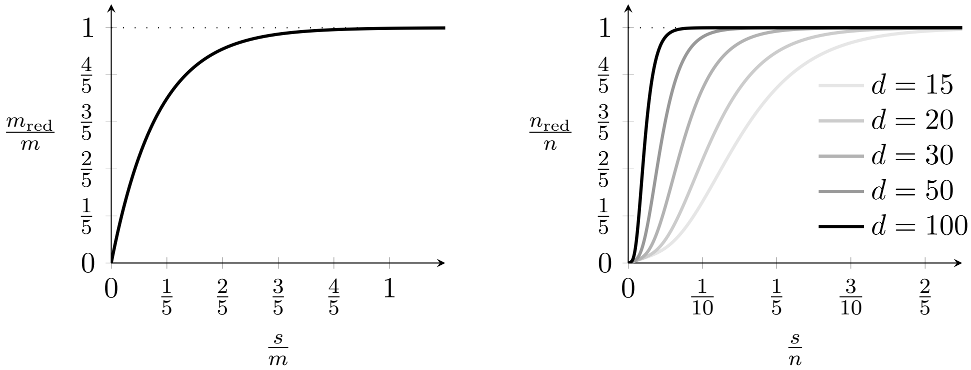
for the event when all samples corresponding to the rays $r_1, \dots, r_k$ are zero, i.e. $b_{r_1} = \dots = b_{r_k} = 0$. A nonzero entry in a cell incident to the rays $r_1, \dots, r_{2D}$ means that the regarding column is supposed to be removed from the system if $b_{r_1} = \dots = b_{r_{2D}} = 0$ and $R_{r_1} = \dots = R_{r_{2D}} = 1$, respectively. Thus, the event $\prod_{i=1}^{2D}(1 - R_{r_i}) = 1$ corresponds to the case when the regarding column remains in the system. The probability for this to happen is the same for all d^D cells and the corresponding rays being incident to them. Thanks to the linearity of the expectation, the result is

$$\mathbb{E}(N_{\text{red}}) = d^D \mathbb{E}\left(\prod_{i=1}^{2D}(1 - R_{r_i})\right) = d^D \left(1 + \sum_{k=1}^{2D} (-1)^k \binom{2D}{k} p_{d,k}^s\right)$$

where the last equation is obtained through multiplying out the product and applying (4.1.14) to the resulting terms. □

An illustration for $D = 3$ is shown in figure 4.3b. Note that (4.1.14) for $k = 1$ validates lemma 4.1.11 once more, since the important events are $1 - R_{r_i} = 0$.

Based on the knowledge of the expected dimensions of the reduced system, recovery



(a) Expected number of rows $\mathbb{E}(2M_{\text{red}})$ for $d = 20$. For other values of d the plot looks similar.

(b) Expected number of columns $\mathbb{E}(N_{\text{red}})$ for different values of d.

Figure 4.3 - Expected size of the reduced CMS system with sensor B_d^3 depending on sparsity s. All curves above have been experimentally validated by generating 1000 instances of uniform s-sparse signals in $\mathcal{X}_s^n$ and averaging the reduced systems dimensions.

guarantees via CMS can be derived depending on the critical sparsity parameter s that generates such reduced systems. If $\frac{2m_{\text{red}}}{n_{\text{red}}} \geq 1$, then the reduced system is square or overdetermined and perfect recovery becomes possible, provided reduced systems have full rank. Also it leads to the critical sparsity

$$s_{\text{crit}} := \max\left\{ s \in [n] \;\middle|\; \frac{2m_{\text{red}}}{n_{\text{red}}} \geq 1 \right\} \tag{4.1.15}$$

reflecting the case of a square system matrix. For fixed d, a standard root finding algorithm can find the corresponding solution for the sparsity s by solving

$$2\mathbb{E}(M_{\text{red}}) - \mathbb{E}(N_{\text{red}}) = 0 \ , \tag{4.1.16}$$

where $\mathbb{E}(2M_{\text{red}})$ and $\mathbb{E}(N_{\text{red}})$ are given in (4.1.11) and (4.1.12). This was done in figure 4.4 showing s_{crit} for several different dimensions d. For comparison the corresponding graphs of the original sensor (2.5.26) with $D = 3$ and of a Gaussian random sensor $\in \mathbb{R}^{m\times n}$ derived from (2.5.19) as described in section 2.5.3 are plotted, too. The curve for the critical sparsity s_{crit}^A that induces square reduced systems with respect to the original sensor A_d^3 can also be done using the expected number of nonzero rows (2.5.29) and the expected number of nonredundant cells (2.5.30). The fraction $\frac{s_{\text{crit}}}{s_{\text{crit}}^A}$ is plotted in figure 4.5 and shows an increasing function for $d \geq 10$ having a minimum of 3.4, approximately, and that becomes larger than 17 for $d \geq 10^4$. That means the sparsity s for signals acquired by the CMS sensor is allowed to be at least 3.4 times greater compared to the one of the original sensor. The more the discretization grid size d grows, the better the recovery performance of the CMS sensor is in this scenario.

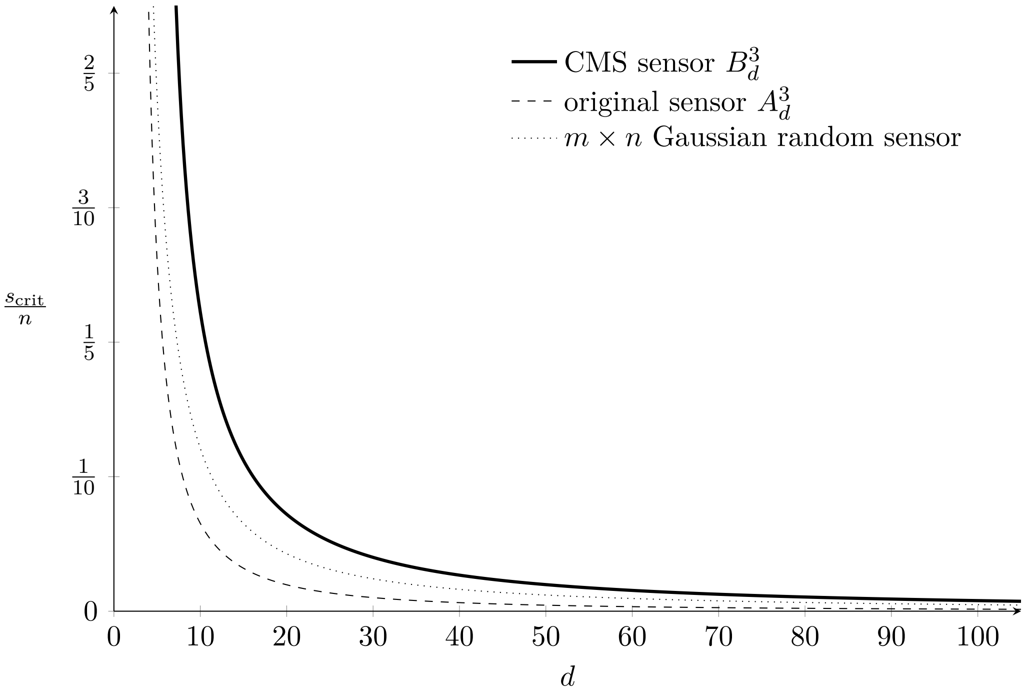


Figure 4.4 - Critical sparsity s_{crit} that induces square reduced systems in the case of the CMS sensor B_d^3 (4.1.9, continuous curve), in the case of the original sensor A_d^3 (2.5.26, dashed curve) and in the case of a Gaussian random sensor of the same size as A_d^3 (dotted curve). For B_d^3, s_{crit} is obtained by solving $\mathbb{E}(2M_{\text{red}}) = \mathbb{E}(N_{\text{red}})$ for s, where $\mathbb{E}(2M_{\text{red}})$ and $\mathbb{E}(N_{\text{red}})$ are given in (4.1.11) and (4.1.12). A higher s_{crit} value implies recovery of denser vectors (particle distributions) and shows that CMS performs best.

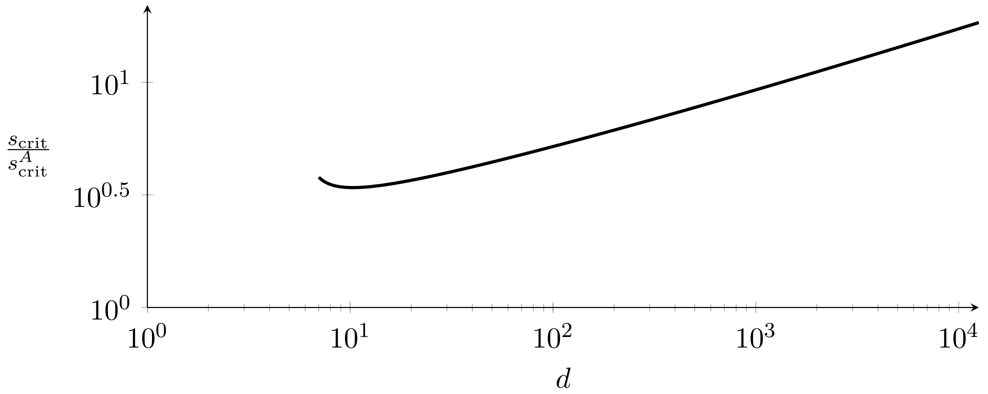


Figure 4.5 - Fraction $\frac{s_{\text{crit}}}{s_{\text{crit}}^A}$ between the critical sparsities of sensor (2.5.26) and the corresponding CMS sensor (4.1.9), that is the theoretical performance gain from using CMS regarding the sparsity of the signal. The graph shows a minimum at $d = 10$ of approximately 3.4 and the superiority grows further with increasing d.

The CMS sensor clearly shows the best performance compared to the other two, but this is not surprising as it has twice as much measurements available due to the known permutation matrix P describing the transformation between the considered points in time. Estimating P at the same time is more relevant for the intended application and the topic of section 4.2.

As already mentioned, CMS sensors of the investigated kind do not have maximum rank. This issue is addressed in the next section.

4.1.2.1 Perturbation of the CMS System

Sensors of the form (2.5.26) do not have maximum rank $2m$. Rather than deriving the critical sparsity for overdetermined reduced CMS systems, a slight perturbation of the nonzero sensor entries will be considered in order to obtain recovery guarantees for denser vectors. Therefor, a *perturbed* sensor is used as given in definition 2.50. As described in section 2.5.4.4, the perturbation of a sensor does not change its sparsity structure, so that the previous investigation of the size of the reduced CMS system is still valid for a perturbed sensor $\tilde{A}_d^D$. The corresponding perturbed CMS sensor then is

$$\tilde{B}_d^D := \begin{bmatrix} \tilde{A}_d^D \\ \tilde{A}_d^D P \end{bmatrix} \in \mathbb{R}^{2m \times n} \; . \tag{4.1.17}$$

The subsequent proposition is a sufficient condition that guarantees uniqueness of a nonnegative and sparse enough vector sampled by a CMS sensor of the form (4.1.17) with high probability.

Proposition 4.14 *(unique recovery using perturbed CMS sensor)*:
There exists a perturbed matrix $\tilde{A}_d^D$ that has the same sparsity structure as A_d^D from (2.5.26) such that the perturbed system $\tilde{B}_d^D x = \tilde{B}_d^D x^*$, with $\tilde{B}_d^D$ defined as in (4.1.17), admits unique recovery of an s-sparse nonnegative signal $x^* \in \mathcal{X}_s^n$ with high probability, i.e. the set

$$\left\{ x \in \mathbb{R}^n \;\middle|\; \tilde{B}_d^D x = \tilde{B}_d^D x^*, x \geq 0 \right\} \tag{4.1.18}$$

is almost surely a singleton, if s satisfies condition $s \leq s_{\text{crit}}$, where s_{crit} solves (4.1.16).

This proposition is the analogy to theorem 2.54 for the CMS system and the proof is analogous, too. All elements x in (4.1.18) have equal ℓ_1-norm since

$$\|x\|_1 = \mathbb{1}_n^\top x = \frac{1}{2D} \mathbb{1}_m^\top \tilde{B} x = \frac{1}{2D} \mathbb{1}_m \tilde{B} x^*$$

holds. Thus, enforcing sparsity by ℓ_1-regularization as in (2.5.27) would be redundant. For sparse recovery, it is enough to take nonnegativity into account by solving (2.5.31).

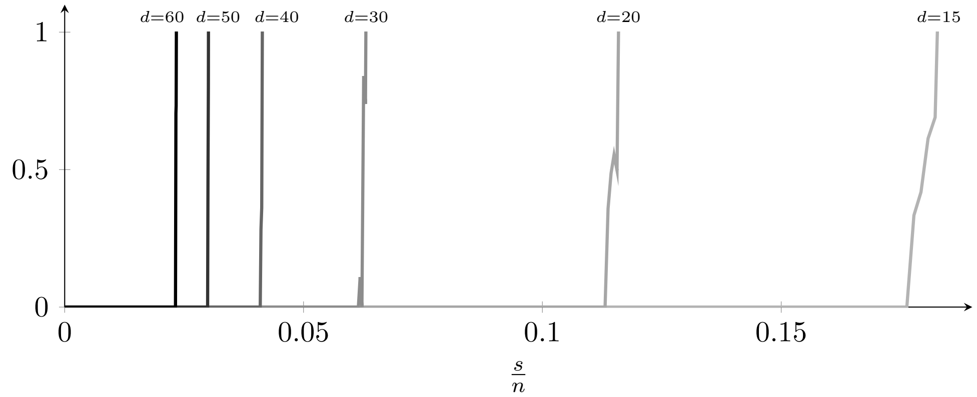


Figure 4.6 **-** Averaged error norm $\|\hat{x} - x^*\|_2$ between the recovered and optimal signal out of 200 recovery runs of $Bx = b$ with random b for different grid sizes d depending on the relative sparsity $\frac{s}{n}$. For every d there is a bound up to which perfect recovery is possible. Beyond this bound the error rises very quickly as more particles are added to the image.

4.1.3 Recovery Experiments with Known Permutation

In the previous section theoretical recovery guarantees of an s-sparse binary signal $x \in \mathcal{X}_s^n$ sampled by a CMS sensor with support (4.1.9) are derived. In the analysis, the critical boundary s_{crit} on the sparsity computed from the expected size of the reduced system plays an essential role. The validation of these results on the basis of the CMS sensor B_d^3 is the subject of this section.

Several experiments are run in which random signals discretized on a $d \times d \times d$ grid are recovered. One experiment consists of a given grid dimension d, sparsity s, uniformly random permutation matrix $P \in \mathcal{P}_n$ as part of the CMS sensor and a uniformly random binary signal $x^* \in \mathcal{X}_s^n$. Next, observations are generated by $b = B_d^3 x^*$ and the size of the corresponding reduced system is computed (see section 2.5.4.2). A solution

$$\hat{x} = \underset{x \geq 0}{\arg\min} \|B_d^3 x - b\|_1$$

is obtained by using the reduced system and in order to certify recovery the recovery error $\|\hat{x} - x^*\|_2$ is computed. For every combination of several grid dimensions d and sparsities s this is done 200 times. The averaged recovery error depending on the relative sparsity $\frac{s}{n}$ is shown in figure 4.6 for a few grid dimensions d. The maximum relative sparsity yielding a mean error norm of numerically zero is the boundary up to which perfect recovery can be empirically guaranteed. Figure 4.7 collects the estimated boundary of every experiment and compares it to the theoretical result in form of s_{crit} derived in (4.1.15). Overall, the experimental results are in accordance with the theory. The estimated sparsity boundary is significantly greater than the theoretical bound s_{crit}.

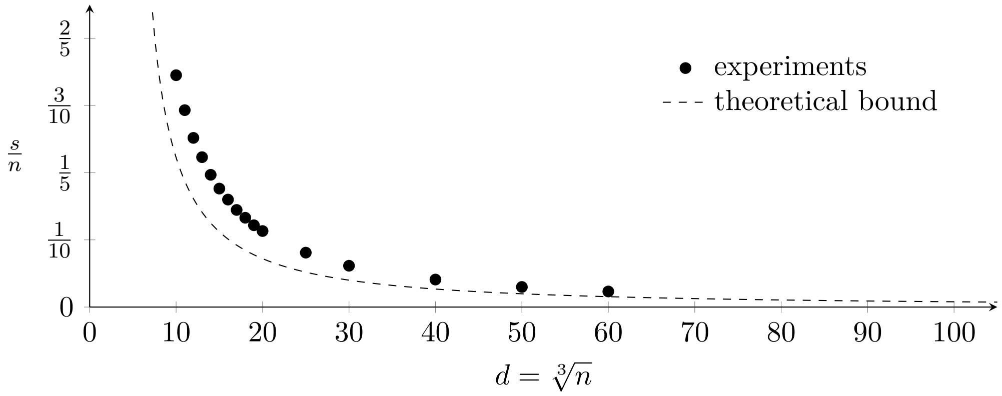


Figure 4.7 **-** Experimental validation of the derived recovery guarantee in form of the relative sparsity boundary $\frac{s}{n}$ for $D = 3$ together with the theoretical bound derived with the help of s_{crit} according to (4.1.15) for comparison. The experiments are in agreement with the theory and show a significantly better recovery of denser images than predicted.

4.2 Simultaneous CMS Reconstruction and Motion Estimation

The centerpiece of compressed motion sensing and the subject of the present section is the joint computation of a signal from observations and its change over time. Two-dimensional or three-dimensional particle images and their corresponding motion is the considered case of application.

Similar to previous sections the signal is assumed to be an s-sparse and n-dimensional binary vector resulting from the discretization of a D-hypercube-shaped domain $\Omega \subset \mathbb{R}^D$ into a grid graph with n cells having an arbitrarily fixed order. The signal has the states $x \in \mathcal{X}_s^n$ at a point in time and $y \in \mathcal{X}_s^n$ at a subsequent one. An 1 entry indicates the presence of a particle at the grid cell associated with the vector index whereas 0 means no particle there. The change happening in between the states x and y can be described by a permutation matrix $P \in \mathcal{P}_n$ so that $y = Px$ imposing a 1-to-1 correspondence between all n grid cells. This is more than necessary since only the motion of the s particles is of current interest.

Section 4.2.1 details the representation of particle motions between two frames by permutation matrices. Based on this, the joint recovery and motion estimation problem is elaborated in section 4.2.2.

4.2.1 Linear Particle Assignment

Section 2.6.1 introduces an optimal transport method leading to a linear program (2.6.5) suitable for the estimation of 1-to-1 assignment problems. However, an application to the current situation is not easily possible since this method assumes a present particle at every location meaning $x = y = \mathbb{1}$. In order to adapt it, the assignment estimation has to be restricted to the support of the signal states x and y or, respectively, to the s particles.

Considering full $x \in \mathcal{X}_s^n$ and $y \in \mathcal{X}_s^n$, there are lots of permutations fulfilling the transformation $x = Py$ with $P \in \mathcal{P}_n$. All those permutation matrices mapping a known $x \in \mathcal{X}_s^n$ to a known $y \in \mathcal{X}_s^n$ are collected in the set $\Pi_n(x, y)$. Corresponding to s moving particles, an element $P \in \Pi_n(x, y)$ maps the support

$$S_x := \operatorname{supp}(x) \qquad \text{to} \qquad S_y := \operatorname{supp}(y)$$

and, consequently, the $n - s$ empty cells $S_x^c := [n] \setminus S_x$ to $S_y^c := [n] \setminus S_y$. Due to this partition of P, the number of permutations fulfilling $x = Py$ is $|\Pi_n(x, y)| = s!\,(s-n)!$. The corresponding permutation restricted to the s cells with present particles can be represented by $P_{S_y,S_x} \in \mathcal{P}_s$ in the view of

$$y_{S_y} = P_{S_y,S_x} x_{S_x} \qquad \text{whereas} \qquad y_{S_y^c} = 0 \ .$$

Associating the assignment of $j \in S_x$ to $i \in S_y$ with the cost given by element $C_{i,j}$ of matrix $C \in \mathbb{R}_+^{s \times s}$ enables the application of the initially mentioned optimal transport method. This leads to the *linear assignment* problem

$$\min_{P \in \mathcal{B}_s} \operatorname{tr}\left(C^\top P\right) \tag{4.2.1}$$

where $\mathcal{B}_s = \left\{P \in \mathbb{R}_+^{n \times n} \colon P\mathbb{1} = \mathbb{1}, P^\top \mathbb{1} = \mathbb{1}\right\}$ is the assignment polytope 2.6.1 which is a relaxation to the feasible set of permutation matrices $\mathcal{P}_s$, see section 2.6.1, and $P = P_{S_y,S_x}$. The entries of the cost matrix C_s are related to the *energy* required to move the particles in x to y. A natural choice is the Euclidean distance between every two grid vertexes i and j, i.e. $C_{i,j} = \|v_i - v_j\|_2$, where $v_i \in \Omega$, $i \in [n]$ denotes a vertex location. Of course, other distance functions are possible and might be even more suitable if prior knowledge on the motion is available.

Now, as the estimation of P_{S_y,S_x} is determined, (4.2.1) must be seen in the context of the entire grid. Therefore, arbitrary assignments of s particles on n grid cells are considered next, where the permutation matrix $P \in \mathcal{P}_s$ is enlarged accordingly, leading to a larger displacement matrix $T \in \{0, 1\}^{n \times n}$.

Proposition 4.15 *(displacement matrix with permutation entries)*:
Let $x, y \in \mathcal{X}_s^n$ and $C \in \mathbb{R}_+^{n \times n}$ be given. Assume that $\tilde{P} \in \Pi_n(x, y)$ is a solution of

$$\min_{P \in \mathcal{P}_n} \operatorname{tr}\left(C^\top P\right) \qquad \text{subject to} \qquad Px = y, P^\top x = y, P \geq 0. \tag{4.2.2}$$

Then the assignment matrix $T \in \{0, 1\}^{n \times n}$ with $T_{S_y,S_x} := \tilde{P}_{S_y,S_x} \in \mathcal{P}_s$ and $T_{S_y^c,S_x^c} = 0_{n-s}$ is a solution of (4.2.2), too.

Proof. The constraints are met since $Tx = \tilde{P}x = y$, $T^\top y = \tilde{P}^\top y = x$ and $T \geq 0$. Furthermore, $\operatorname{tr}\left(C_{S_x,S_y}^\top T_{S_y,S_x}\right) = \operatorname{tr}\left(C_{S_x,S_y}^\top \tilde{P}_{S_y,S_x}\right)$ is minimal due to $T_{S_y,S_x} \in \mathcal{P}_s$

and $\operatorname{tr}\left(C^{\top}_{S^c_x,S^c_y} T_{S^c_y,S^c_x}\right) = 0$ does not change the target value. □

Hence, the optimal assignment T between $x \in \mathcal{X}^n_s$ and $y \in \mathcal{X}^n_s$ is a sparse matrix with s nonzero entries that equals a permutation matrix when restricted to the support of x and y. In this regard, the remaining $n-s$ assignments $P_{S^c_x,S^c_y}$ are unimportant regarding valid correspondences between nonzero entries of x and y. This means the underlying permutation can be assumed to have a cycle containing those $n-s$ elements which maximizes the chance of the maximum rank condition (4.1.6) to be fulfilled. Moreover, proposition 4.15 allows to formulate a linear joint optimization problem where the signals are unknown, as described in the next section.

4.2.2 Joint Reconstruction and Displacement Estimation

The problem of jointly estimating $x, y \in \mathcal{X}^s_n$ and $P \in \Pi_n(x,y)$ is based on merging the CMS system (4.0.2) with the linear assignment problem in (4.2.2) into a single optimization problem. For some given transportation costs $C \in \mathbb{R}^{n\times n}_{+}$ and observations $b_x, b_y \in \mathbb{R}^m$ acquired by a known ordinary sensor $A \in \mathbb{R}^{m\times n}$ this gives

$$\begin{aligned} \underset{\substack{x,y\in\mathbb{R}^n \\ P\in\mathbb{R}^{n\times n}}}{\text{minimize}} \quad & \operatorname{tr}\left(C^{\top}P\right) \\ \text{subject to} \quad & Ax = b_x,\ Ay = b_y,\ x, y \geq 0, \\ & Px = y,\ P^{\top}y = x,\ P \geq 0\ . \end{aligned} \tag{4.2.3}$$

This is a block biconvex problem, that is (4.2.3) is convex with respect to P for every arbitrary fixed x and y on the one hand, and on the other hand it is convex with respect to x and y for every fixed P. Those two blocks of variables (x, y) and P could be minimized alternatingly as introduced in section 2.4.3 or by a block coordinate descent approach that sequentially updates the two blocks via proximal minimization, see e.g. [XY13]. Instead, the non-convex constraints involving variables of both blocks, $Px = y$ and $P^{\top}y = x$, are replaced so that the linear program

$$\begin{aligned} \underset{\substack{x,y\in\mathbb{R}^n \\ P\in\mathbb{R}^{n\times n}}}{\text{minimize}} \quad & \operatorname{tr}\left(C^{\top}P\right) \\ \text{subject to} \quad & Ax = b_x,\ Ay = b_y,\ x, y \geq 0, \\ & P\mathbb{1} = y,\ P^{\top}\mathbb{1} = x,\ P \geq 0 \end{aligned} \tag{4.2.4}$$

is solved. (4.2.4) is referenced to as *CMS program*. By utilizing the displacement matrix T again, the following proposition shows that the important assignments, those between x_{S_x} and y_{S_y}, coincide for solutions of (4.2.3) and (4.2.4).

Proposition 4.16 *(coinciding solutions for particle displacements)*:
Let $x, y \in \mathcal{X}^n_s$, $P \in \Pi_n(x,y)$ and $T \in \{0,1\}^{n\times n}$ with $T_{S_y,S_x} := P_{S_y,S_x} \in \mathcal{P}_s$ and $T_{S^c_y,S^c_x} := 0_{n-s}$. If the tupel (x, y, T) is a solution of (4.2.3), then (x, y, T) is a solution of (4.2.4). Likewise, a solution (x, y, T) to (4.2.4) is a solution to (4.2.3), too.

Proof. It holds that $Tx = T\mathbb{1} = y$ and $T^\top y = T^\top \mathbb{1} = x$ and hence (x, y, T) is feasible for both (4.2.3) and (4.2.4) and optimal at the same time since the target functions are equal. □

For a given cost matrix C no recovery guarantee can be given. For example, a constant matrix imposing equal costs to all possible assignments is highly insufficient for estimating the correct one. However, if the provided cost matrix is appropriate, unique recovery can be expected, again by using a perturbed sensors as introduced in section 2.5.4.4.

Corollary 4.17 *(unique recovery using perturbed sensor)*:
Assume $x \in \mathcal{X}_s^n$ is mapped to $y = Px$ via $P \in \Pi_n(x, y)$. Then there exists a perturbation $\tilde{A}$ of A_d^D from (2.5.26) and a cost matrix $C \in \mathbb{R}_+^{n\times n}$ such that x, y and the assignment matrix $T \in \{0,1\}^{n\times n}$ with $T_{S_y,S_x} := P_{S_y,S_x} \in \mathcal{P}_s$, $T_{S_y^c,S_x^c} := 0_{n-s}$ and $y = Tx$ can be recovered perfectly with high probability by solving problem (4.2.4), specialized to

$$\begin{aligned} \underset{\substack{u,v\in\mathbb{R}^n \\ T\in\mathbb{R}^{n\times n}}}{\text{minimize}} \quad & \operatorname{tr}\left(C^\top T\right) \\ \text{subject to} \quad & \tilde{A}u = \tilde{A}x,\ \tilde{A}v = \tilde{A}y,\ u, v \geq 0, \\ & T\mathbb{1} = y,\ T^\top \mathbb{1} = x,\ T \geq 0 \end{aligned} \tag{4.2.5}$$

provided that $s \leq s_{\text{crit}}$, with s_{crit} defined by (4.1.15) .

Proof. By proposition 4.14 there exists $\tilde{A}$ such that $x \in \mathcal{X}_s^n$ is the unique nonnegative solution of

$$\tilde{B}u = \begin{bmatrix} \tilde{A} \\ \tilde{A}P \end{bmatrix} u = \begin{bmatrix} \tilde{A} \\ \tilde{A}P \end{bmatrix} x = \begin{bmatrix} \tilde{A}x \\ \tilde{A}y \end{bmatrix}.$$

Furthermore, the tupel (x, y, P) is a solution to (4.2.3) by assumption so that (x, y, T) with $T_{S_y,S_x} = P_{S_y,S_x} \in \mathcal{P}_s$ and $T_{S_y^c,S_x^c} = 0_{n-s}$ is a (vertex) solution to (4.2.5) due to proposition 4.16 for an appropriate $C \in \mathbb{R}_+^{n\times n}$. □

The CMS program (4.2.4) is the main contribution to simultaneous computation of signal recovery and its change or, respectively, to joint image reconstruction and motion estimation. It is experimentally examined in the next section.

4.2.3 CMS Experiments

If the transformation in form of the permutation matrix P of the CMS system (4.0.2) is known, it reduces to an ordinary linear equation system which is underdetermined in most applications. Corresponding experiments showing the potential recovery capabilities of CMS are carried out in section 4.1.3. Here, besides the signal states x and y, the permutation P is assumed to be unknown and supposed to be recovered. Therefore, the experiments are formulated as a CMS program (4.2.4) and solved by a standard linear minimization solver.

In order to save memory and computation time it is very reasonable to reduce the two equation systems $Ax = b_x$ and $Ay = b_y$ as part of the CMS program first, according to the procedure described in section 2.5.4.2. Using the notation from there correspondingly for x and y, this amounts in the actual solving of the *reduced CMS program*

$$\begin{aligned} \underset{\substack{x_{\text{red}}\in\mathbb{R}^{|J_x^c|}\ y_{\text{red}}\in\mathbb{R}^{|J_y^c|} \\ P_{\text{red}}\in\mathbb{R}^{|J_y^c|\times|J_x^c|}}}{\text{minimize}} \quad & \operatorname{tr}\left(C_{J_y^c,J_x^c}^{\top} P_{\text{red}}\right) \\ \text{subject to} \quad & A_{I_x^c,J_x^c} x_{\text{red}} = (b_x)_{I_x^c}\,,\ A_{I_y^c,J_y^c} y_{\text{red}} = (b_y)_{I_y^c}\,,\ x_{\text{red}}, y_{\text{red}} \geq 0, \\ & P_{\text{red}} \mathbb{1} = y_{\text{red}},\ P_{\text{red}}^{\top} \mathbb{1} = x_{\text{red}},\ P_{\text{red}} \geq 0\ , \end{aligned} \tag{4.2.6}$$

where the original solutions are obtained from setting

$$x_{J_x^c} = x_{\text{red}}, \quad x_{J_x} = 0 \qquad \text{and} \qquad y_{J_y^c} = y_{\text{red}}, \quad y_{J_y} = 0$$

afterwards. The utilization of the reduced CMS program is implicitly assumed, even though it is not mentioned in the following.

All experiments in this section consists of the reconstruction of binary images $x \in \mathcal{X}_s^n$ and $y \in \mathcal{X}_s^n$ of dimension $D = 2$ or $D = 3$ from a few $(D-1)$-dimensional projections at two points in time and the simultaneous estimation of the particle motion in between. The images are square and cubic-shaped having size d in each dimension so that their number of pixels and voxels, respectively, is equal to $n = d^D$. Each image shows exactly s particles.

4.2.3.1 Evidence of Synergy

The first experiment deals with small 6×6 images showing 2 particles, only. The matrix A_6^2 from (2.5.26) is used as a sensor generating the observations $b_x, b_y \in \mathbb{R}^{12}$. Those are used as input together with a cost matrix $C \in \mathbb{R}^{36\times36}$ filled with Euclidean distances

$$C_{i,j} = \|v_i - v_j\|_2 \tag{4.2.7}$$

between pixel locations $v_i, v_j \in \mathbb{R}^2$ for $i, j \in [6]$ in this case. Figure 4.8 illustrates the output of the CMS program. In addition, the figure shows the usual ℓ_1-minimization recovery for positive signals (2.5.27) applied separately to both linear equation systems $Ax = b_x$ and $Ay = by$ for obtaining the subsequent images. The ℓ_1-recoveries fail whereas CMS yields perfect recovery of the two images and the motion in between. Particularly remarkable is the fact that the sensor A_6^2 is rather poor since it cannot recover a signal with sparsity $s > 1$ in a standard CS scenario with high

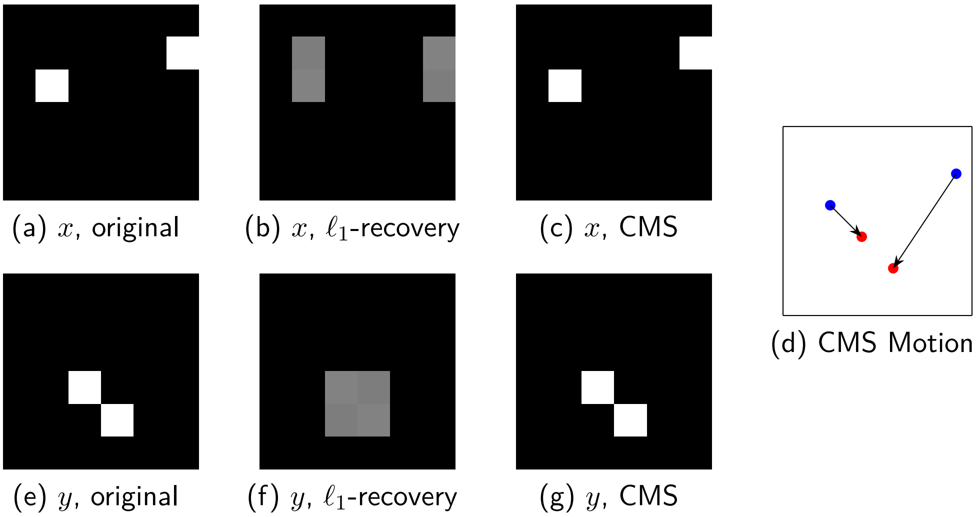


Figure 4.8 **-** Recovery of 2-sparse subsequent 6×6 images x (a) and y (e) by separate standard ℓ_1-recovery and by CMS, respectively, based on a sensor matrix A_6^2 (2.5.26). Standard recoveries (b) and (f) by solving (2.5.27) fail due to poor sensor properties of A_6^2, despite sparsity. Using the same number of measurements the corresponding CMS sensor (4.1.9) leads to unique recovery (c) and (g) and correspondence (motion) information (d) by solving (4.2.4).

probability [PS14]. Nevertheless, CMS is capable of doing so and seems to turn this poor sensor into one worth considering in an application. This is a clear evidence of synergy enabled by the joint computation of recovery and motion. The subsequent CMS results are presented in a similar fashion as in figure 4.8d.

4.2.3.2 Realistic Scenarios

In order to test CMS in more application-oriented scenarios for $D = 2$ and $D = 3$, several similar experiments with different grid sizes d and number of particles s are carried out. The underlying movements are chosen to be realistic turbulent random flows discretized on the grid so that y is s-sparse when computing it with the help of the flow from x. On this basis, observations $b_x, b_y \in \mathbb{R}^m$ are sampled by using different sensor types and used as input for the CMS program (4.2.4). Again, Euclidean distances (4.2.7) are the transport costs between grid cells.

Figures 4.9 shows a 2D and 4.10 a 3D example of the experimental results. In the 2D experiment, $s = 80$ particles are displaced and the images are sampled by a tomographic sensor with 4 projections as shown in figure 3.2. The 3D experiment uses sensor A_{256}^3 from (2.5.26) with 3 orthogonal projections and 500 particles. In both cases CMS recovers the correct images within 38 (2D) and 15 (3D) seconds, approximately, on an usual desktop PC[1]. This seems counter-intuitive but is due to the two systems $Ax = b_x$ and $Ay = b_y$ of the CMS program (4.2.4) being overde-

[1] Intel Core i5-2410M together with 8GB memory

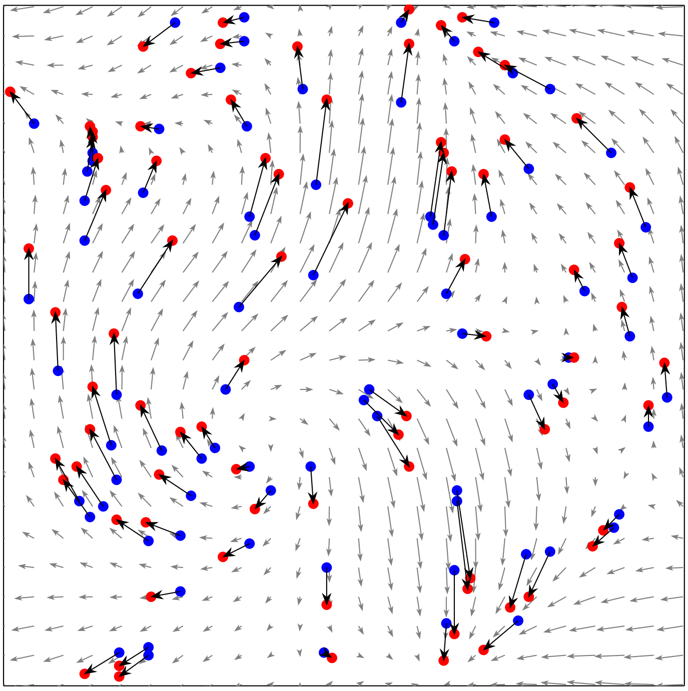


Figure 4.9 **-** Output of the CMS program (4.2.4) recovering 2D images on a 256×256 grid using an original sensor illustrated in figure 3.2. Particles of the first and the second image are marked in blue and red, respectively, and the detected correspondences in between are drawn as black arrows. The underlying motion used for generating this example is shown as gray arrows.

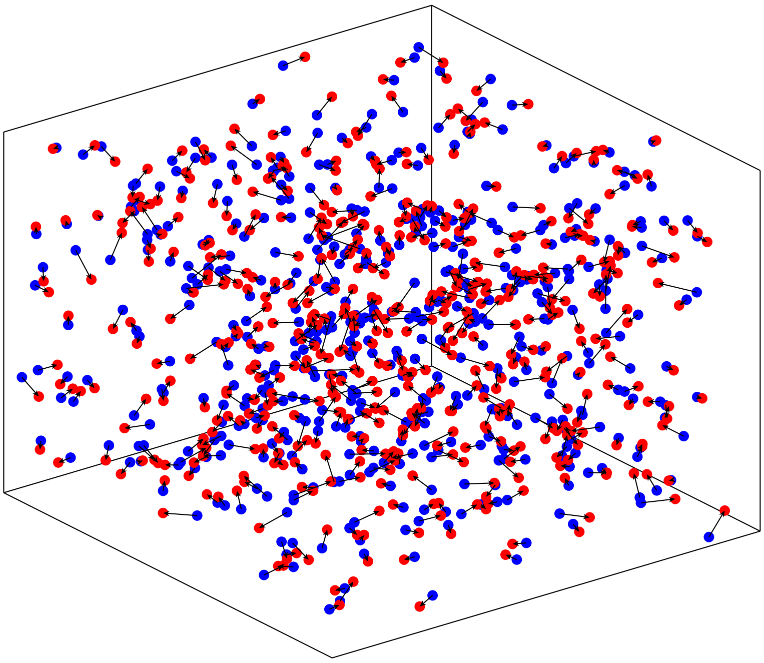


Figure 4.10 **-** Output of the CMS program (4.2.4) applied to 3D images with 500 particles on a $256 \times 256 \times 256$ grid using CMS sensor A^3_{256} (2.5.26). Particles of the first and the second image are marked in blue and red, respectively, and the detected correspondences in between are drawn as black arrows.

termined after reduction in the 3D example, which is, in turn, a consequence of a relative sparsity $\frac{s}{n} \approx 2.98 \cdot 10^{-5}$ for 3D compared to approximately $1.22 \cdot 10^{-3}$ in 2D. In general, the greater the relative sparsity, the longer the computation takes since the size of the reduced system naturally grows.

Besides CMS recovery, separate ℓ_1-reconstructions of the two images are computed in all experiments, as well. In many cases separate recovery fails whereas, in contrast, the CMS approach yields perfect reconstructions as already seen in the minimal example in figure 4.8.

When looking carefully at computed solution and possible corresponding permutations, respectively, it can be noticed that underlying realistic motions lead to comparably long cycles. An example is the direction field shown in figure 4.9 in form of gray arrows. Realistic motions always look similar to this, that is, following the direction of an arrow, the subsequent one rarely points back in opposite direction. If this was the case a particle at either position would move back and forth repeatedly which is unrealistic and corresponds to a permutation cycle of length 2. But, particles rather tend to follow a longer path which means in turn longer, and most often much longer permutation cycles. Thus, for typical realistic particle motions, the maximum rank condition 4.1.6 seems fulfilled with overwhelming probability.

4.2.3.3 Incorporated Prior Knowledge

CMS offers the possibility to incorporate prior knowledge by defining the cost matrix C accordingly. In order to illustrate the heavy influence of this choice on the output of the CMS program (4.2.4), the following experiment is carried out.

The setting is a 3D volume with a $257 \times 257 \times 257$ grid discretization filled with $s = 200$ particles. Again, A^3_{257} from (2.5.26) is used as sensor. The motion used to generate $y \in \mathcal{X}^3_{200}$ from $x \in \mathcal{X}^3_{200}$ can be described as

$$y = \left\lfloor \begin{bmatrix} \cos(\alpha) & \sin(\alpha) & 0 \\ -\sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 1 \end{bmatrix} x + \begin{bmatrix} 0 \\ 0 \\ v_z \end{bmatrix} \right\rceil$$

with $\alpha = \operatorname{rad}(5v_z)$ where $\lfloor \cdot \rceil$ denotes the nearest integer function necessary for y being a binary image. This represents a rotation around and a constant shift along the z-axis where $v_z \in \mathbb{N}$ is the vertical velocity on a voxel basis illustrated in figure 4.11a. In addition to an Euclidean cost matrix (4.2.7), one allowing particles to move along their orbit around the z-axis without penalty is used, i.e.

$$C(u, v) = \min_{z \in \mathbb{R}^3} \left\{ \|z - v\|_2^2 \;\middle|\; \left\| \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} \right\| = \left\| \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \right\| \right\} \tag{4.2.8}$$

for two grid locations $u, v \in \mathbb{R}^3$. (4.2.8) is referenced as *orbit costs* here. This corresponds to an application-oriented situation where an rotation of particles is expected but the deviation from this is unknown. In this case the deviation is the constant shift v_z. The result of this vortex scenario for different values of velocity v_z and either cost matrices can be found in figure 4.11.

Using Euclidean costs, the CMS solution is correct for approximately a vertical velocity $v_z = 2$ accompanied by a rotation of 10°. Larger values lead to wrong assignments between particles especially starting to occur in the outer region of the vertex where displacements are larger than closer to the center. In connection with orbit costs, perfect motion recovery remains possible even for large displacements with a velocity $v_z = 18$ and a rotation of 90°.

This experiment still works with slightly larger velocities $v_z > 18$ and corresponding rotation angles. Similarly, the sparsity of $s = 200$ particles is not the limit by far, but is used in order to produce recognizable results. The quality of the motion estimation is still satisfying for $s = 500$ and above, but at approximately $s = 350$ misassignments start to occur.

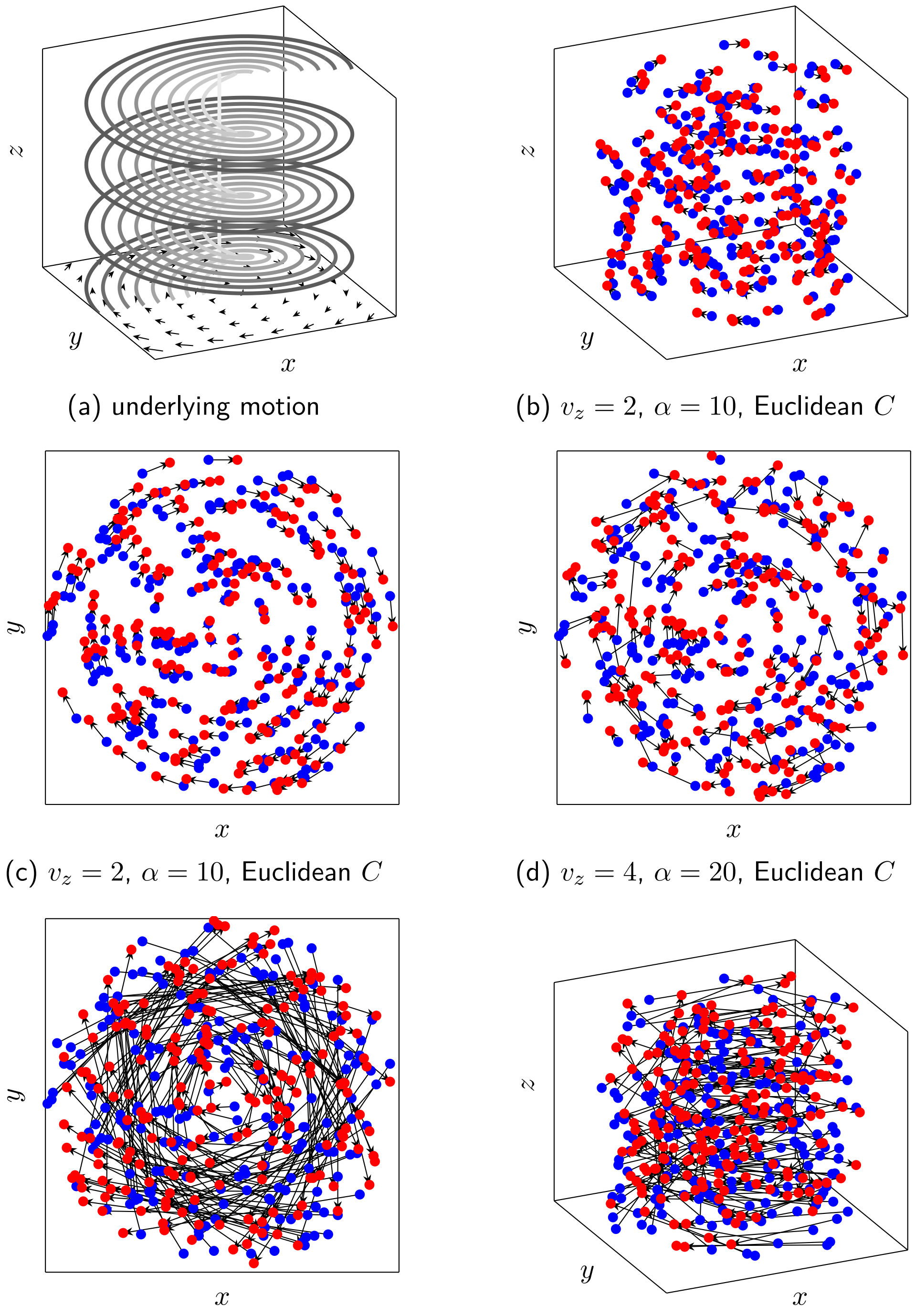
4.2.4 Relaxation

The compressed motion sensing approach presented above is very fast and reliable if the underlying images allow a 1-to-1 correspondence. This is not the case if particles enter or leave the image domain. Furthermore, it might occur that in practice particles are closely located and occupy the same discretization cell. In this case, the image cannot be represented by a binary signal in $\mathcal{X}_s^n$ any more. Thus, an extension to more general signals is reasonable.

Assuming one binary image $x \in \mathcal{X}_s^n$ shows more particles than $y \in \mathcal{X}_s^n$, i.e. $s_x := \|x\|_0 > \|y\|_0 =: s_y$. Then, the constraints of the CMS program (4.2.4) $P\mathbb{1} = y$ and $P^\top \mathbb{1} = x$ impose s_x nonzero rows and s_y nonzero columns of P summing to 1 whereas the remaining rows and columns are 0. Since $s_x \neq s_y$, such a matrix P does not exist because the sum of all elements is independent from the summation order. Hence, the CMS program is infeasible and has no solution in this case, suggesting to relax the aforementioned constraints. The *relaxed CMS program* which does exactly that is

$$\begin{aligned} \underset{\substack{x,y\in\mathbb{R}^n\\ P\in\mathbb{R}^{n\times n}}}{\text{minimize}} \quad & \operatorname{tr}\left(C^\top P\right) \\ \text{subject to} \quad & Ax = b_x,\ Ay = b_y,\ x, y \geq 0, \\ & l_y y \leq P\mathbb{1} \leq u_y y,\ l_x x \leq P^\top \mathbb{1} \leq u_x x,\ P \geq 0 \end{aligned} \tag{4.2.9}$$

where $l_x, l_y \in]0,1[$ and $u_x, u_y > 1$ are parameters determining a certain degree of freedom to P. The reduced CMS program (4.2.6) can be relaxed, correspondingly. As the following experiments show, this relaxation ensures the solubility of a CMS scenario using binary images with an unequal amount of particles or nonbinary nonegative images.



(a) underlying motion

(b) $v_z = 2$, $\alpha = 10$, Euclidean C

(c) $v_z = 2$, $\alpha = 10$, Euclidean C

(d) $v_z = 4$, $\alpha = 20$, Euclidean C

(e) $v_z = 18$, $\alpha = 90$, Orbit C

(f) $v_z = 18$, $\alpha = 90$, Orbit C

Figure 4.11 - Vortex scenario in a $257 \times 257 \times 257$ volume with additional vertical shift (a) and detected motions by CMS (b)-(f). Choosing a cost matrix C with Euclidean distances, the correct motion is recovered for a vertical velocity of $v_z = 2$ and a corresponding rotation angle $\alpha = 10$ (b),(c). After increasing both, recovery with an Euclidean C fails (d), but CMS is still capable of recovering the correct motion by using a different cost matrix (4.2.8) referred to as *orbit costs* here. Even for comparably large displacements the CMS solution is perfect (e),(f).

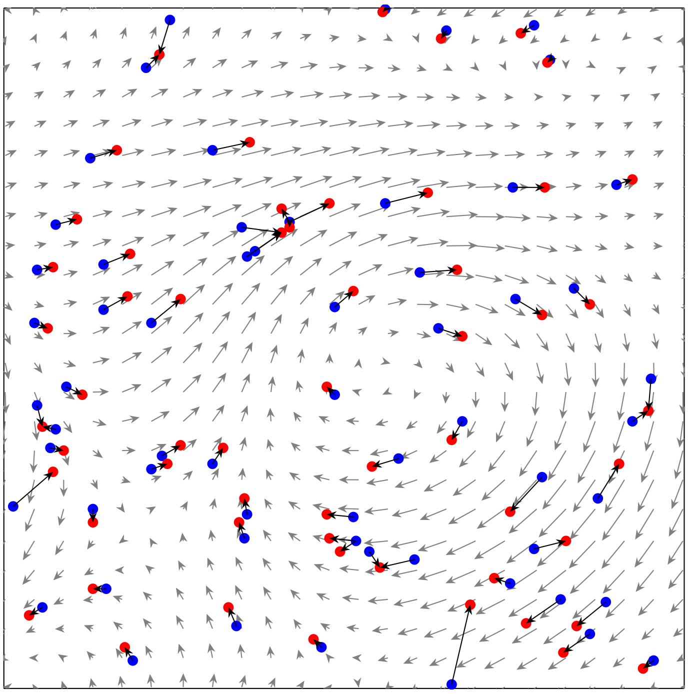


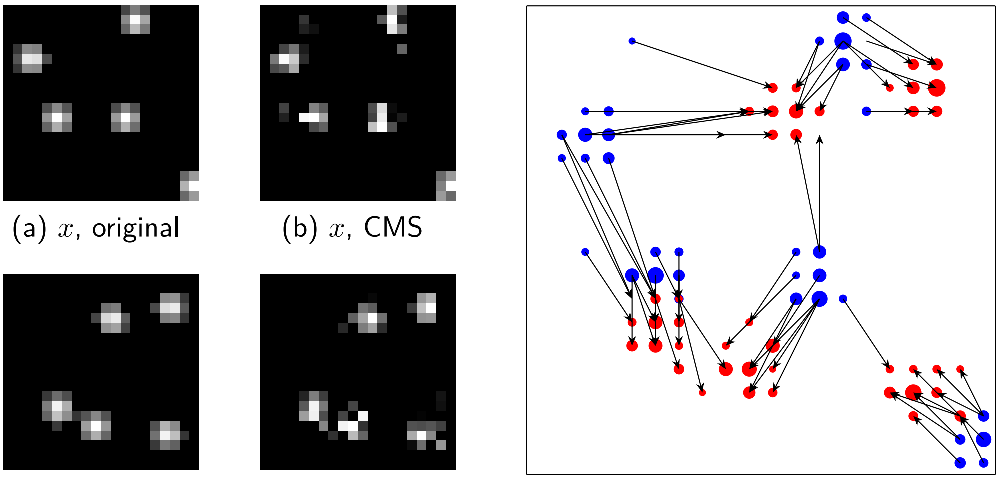
Figure 4.12 - Output of the relaxed CMS program (4.2.9) recovering 2D binary images on a 256×256 grid using the sensor illustrated in figure 3.2. Particles of the first and the second image are marked in blue and red, respectively, and the detected correspondences in between are drawn as black arrows. The underlying motion used for generating this example is shown as gray arrows. There is no 1-to-1 correspondence since the first image shows 60 particles whereas the second one shows only 57. Due to that some assignments are incorrect in certain regions.

4.2.4.1 Relaxed CMS Experiments

Two experiments are carried out in order to pin down the differences between the CMS program (4.2.4) and its relaxed companion (4.2.9). In both settings 2D square images with grid dimension d in each direction and, consequently, $n = d^2$ pixels are sampled by the sensor illustrated in figure 3.2 yielding observations $b_x, b_y \in \mathbb{R}^m_+$ as input for the relaxed CMS program. Moreover, both experiments use an Euclidean cost matrix (4.2.7) and parameters $l_x = l_y = 0.5$ and $u_x = u_y = 1.5$ determined arbitrarily.

In the first experiment s-sparse binary signals $x, y \in \mathcal{X}^n_s$ are recovered. Image x shows 60 particles randomly distributed over the area which are transported by a randomly generated turbulent flow discretized on the grid. Due to the movement, 3 particles are shifted out of the image domain so that the second image y shows 57 particles, only. Figure 4.12 visualizes the setup and recovery results.

The image reconstructions are both perfect, but regarding the motion, there cannot be a 1-to-1 correspondence, naturally, since both images contain a different number of particles. Especially, in regions where particles leave the image domain, local misassignments can be recognized such as at the upper, lower and left image border.



(a) x, original

(b) x, CMS

(d) y, original

(e) y, CMS

(c) CMS Motion

Figure 4.13 - Relaxed CMS recovery of 20×20 images x (a) and y (d) based on the sensor shown in figure 3.2. The recovered images (b) and (e) and the estimated motion information (c) are computed by solving the relaxed CMS program (4.2.9). In this form the pixels belonging to a single transported blob are not displaced as 1-to-1 correspondences

Those particles that are missing in the second image, can create incorrect correspondences in other regions as it happens in the upper left center of figure 4.12. However, the motion is satisfactory in this example, but may be unacceptable in others depending on the relative sparsity, the position of particles and the underlying motion.

Images originating from a real world experiment are rarely sparse binary, not even if the captured scene shows particles only. This is because particles can be present at any location and not just the positions corresponding to the discretization grid. Moreover, a camera cannot capture all particles in perfect focus and thus the particle appearance is smoothed out well approximated by a *Gaussian blob* located at its actual position. In the second experiment Gaussian blobs of the form

$$G_i(z) = \exp\left(-\frac{1}{2}\|z - \mu_i\|_2^2\right)$$

are used in order to represent the i-th particle at location $\mu_i \in \mathbb{R}^2_+$. The sum of all particle blobs $\sum_i G_i(z)$ then corresponds to the continuous image which is sampled on the discrete image grid. In this way, time boundary images $x, y \in \mathbb{R}^n_+$ of 20×20 pixels are generated showing 5 particle blobs placed uniformly random in the domain. Pixels with an intensity value below 0.2 are set to zero. Figure 4.13 illustrates the output of the reduced and relaxed CMS program together with the ground truth images.

The image reconstruction part of CMS is based on a sparse uniform distribution of the signal entries and needs to be generalized for more coherent structures like the blob images. However, the image reconstructions are acceptable since the blobs shown in the underlying original images can be recognized. The estimated motion shows similar misassignments as in the previous experiment and pronounces the weak point of the relaxed CMS formulation even more: Mass or signal intensity is not enforced to remain compact. There are no constraints causing the mass belonging to one blob to stick together as an unit and CMS tears it apart if it is advantageous for the target value.

5 Conclusion

In this work the problem of sparse signal recovery in dynamic sensing scenarios has been considered. In particular, the joint image reconstruction and motion estimation of a corresponding 3D particle distribution indirectly observed from linear measurements of a single static sensor at two different points in time is addressed by both continuous and discrete optimal transport.

The contributions of this thesis are:

- In section 2.5.3.1 the relation between the signal sparsity and the number of Gaussian measurements that guarantee uniqueness with high probability via ℓ_1-minimization with nonnegative or 0/1 box constraints is accurately described. The calculations build on recent compressed sensing theory [ALMT14] that upper bounds the statistical dimension of the descent cone of the structure enforcing regularization. These undersampling rates are used in chapter 4 and compared with the performance of the compressed motion sensor developed in this thesis.

- In chapter 3 the first approach for joint signal and motion recovery is presented. The problem is formulated as a continuous optimal transport between two indirectly observed densities with a physical constraint following the framework of Benamou-Brenier [BB00]. The contributed novelty is the extension of the Benamou-Brenier scheme to the projection constraint corresponding to the indirectly observed densities at two points in time. This still leads to a space-time convex variational problem and allows the integration of the projection constraints by different splitting techniques (weak coupling, scaled ADMM, parallel proximal algorithm). Unfortunately, it was not possible to substantiate by numerical experiments any synergy effect of joint reconstruction of signal and motion.

- In chapter 4 the compressed motion sensor is introduced. The recovery problem is modelled as observing a single signal using two different sensors, a real one and a virtual one induced by signal motion. First, the recovery properties of the resulting combined sensor are examined for the special case of 50% undersampling rate of the static sensor that is assumed to be in general position. Invertibility of the compressed motion sensor is shown under weak conditions on the number of cycles of the permutation matrix underlying motion. Next, it is showed along the lines of [PS14] that complementing the standard Tomo-PIV sensor with a motion sensor, based on wrapped projections due to known motion, significantly improves recovery performance, even beyond Gaussian sensing matrices. In particular, critical sparsities are derived that guarantee that the compressed motion sensor behaves on average like the adjacency matrix of a well connected expander graph. This allows to show further that not only can the signal be uniquely recovered with overwhelming probability by linear programming, but also the correspondence of signal values (signal motion) can be established between the two points in time. Moreover, numerical experiments confirm that via compressed motion sensing the performance of an undersampling static sensor is doubled or, equivalently, that the sufficient number of measurements of a static sensor can be halved. Finally, the more general case of reconstructing blob particles rather than point particles and their correspondence is considered along with a relaxed version of CMS and assessed by numerical experiments.

The work presented in this thesis admits several extensions:

- enhancing the Benamou-Brenier scheme with additional physical fluid flow constraints;
- extending the analysis to a continuous trajectory of time;
- designing constraints which prevent particles from merging or splitting as observed in the relaxed approach;
- relaxing the constraints to allow particles entering or leaving the domain.

Bibliography

[AABC15] I. Abraham, R. Abraham, M. Bergounioux, and G. Carlier. Tomographic Reconstruction from a Few Views: A Multi-Marginal Optimal Transport Approach. *Applied Mathematics and Optimization*, pages 1–19, 2015.

[ABRS08] H. Attouch, J. Bolte, P. Redont, and A. Soubeyran. Alternating Proximal Algorithms for Weakly Coupled Convex Minimization Problems. Applications to Dynamical Games and PDE's. *Journal of Convex Analysis*, 15(3):485–506, 2008.

[Adr05] R. J. Adrian. Twenty Years of Particle Image Velocimetry. *Experiments in fluids*, 39(2):159–169, 2005.

[AF07] L. Ambrosio and A. Figalli. Geodesics in the Space of Measure-Preserving Maps and Plans. *Archive for Rational Mechanics and Analysis*, 2007.

[AF08] L. Ambrosio and A. Figalli. On the Regularity of the Pressure Field of Brenier's Weak Solutions to Incompressible Euler Equations. *Calculus of Variations and Partial Differential Equations*, 31(4):497–509, 2008.

[ALMT14] D. Amelunxen, M. Lotz, M. McCoy, and J. Tropp. Living on the Edges: Phase Transition in Convex Programs with Random Data. *Information and Inference*, 3(3):224–294, 2014.

[BB00] J.-D. Benamou and Y. Brenier. A Computational Fluid Mechanics Solution to the Monge-Kantorovich Mass Transfer Problem. *Numerische Mathematik*, 84(3):375–393, 2000.

[BBG04] J.-D. Benamou, Y. Brenier, and K. Guittet. Numerical Analysis of a Multi-Phasic Mass Transport Problem. *Contemporary Mathematics*, 353:1–18, 2004.

[BPC$^+$10] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers. *Found. Trends Mach. Learning*, 3(1):1–122, 2010.

[Bre99] Y. Brenier. Minimal Geodesics on Groups of Volume-Preserving Maps and Generalized Solutions of the Euler Equations. *Communications on Pure and Applied Mathematics*, 52(4):411–452, 1999.

[Bre03] Y. Brenier. Extended Monge-Kantorovich Theory. In *Optimal Transportation and Applications*, pages 91–121. Springer, 2003.

[BV04] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[Chi12] Y. Chikuse. *Statistics on Special Manifolds*, volume 174 of *Lect. Not. Statistics*. Springer Science & Business Media, 2012.

[CP08] P. L. Combettes and J.-C. Pesquet. A Proximal Decomposition Method for Solving Convex Variational Inverse Problems. *Inverse Problems*, 24(6):065014, 2008.

[CP11a] A. Chambolle and T. Pock. A First-Order Primal-Dual Algorithm for Convex Problems with Applications to Imaging. *Journal of Mathematical Imaging and Vision*, 40(1):120–145, 2011.

[CP11b] P. L. Combettes and J.-C. Pesquet. Proximal splitting methods in signal processing. In *Fixed-Point Algorithms for Inverse Problems in Science and Engineering*, pages 185–212. Springer, 2011.

[CRT06] E. J. Candes, J. K. Romberg, and T. Tao. Stable Signal Recovery from Incomplete and Inaccurate Measurements. *Communications on Pure and Applied Mathematics*, 59(8):1207–1223, 2006.

[CT05] E. J. Candes and T. Tao. Decoding by Linear Programming. *IEEE Transactions on Information Theory*, 51(12):4203–4215, 2005.

[CWX10] T. T. Cai, L. Wang, and G. Xu. New Bounds for Restricted Isometry Constants. *IEEE Transactions on Information Theory*, 56(9):4388–4394, 2010.

[CZ97] Y. Censor and S. A. Zenios. *Parallel Optimization: Theory, Algorithms and Applications*. Oxford University Press on Demand, 1997.

[Deu95] F. Deutsch. The Angle Between Subspaces of a Hilbert Space. In *Approximation Theory, Wavelets and Applications*, pages 107–130. Springer, 1995.

[DH01] D. L. Donoho and X. Huo. Uncertainty Principles and Ideal Atomic Decomposition. *IEEE Transactions on Information Theory*, 47(7):2845–2862, 2001.

[Dix49] J. Dixmier. Étude sur les Variétés et les Opérateurs de Julia, avec quelques Applications. *Bulletin de la Société Mathématique de France*, 77:11–101, 1949.

[Don06] D. L. Donoho. Compressed Sensing. *IEEE Transactions on Information Theory*, 52(4):1289–1306, 2006.

[ESWvO06] G. E. Elsinga, F. Scarano, B. Wieneke, and B. W. van Oudheusden. Tomographic Particle Image Velocimetry. *Experiments in Fluids*, 41(6):933–947, 2006.

[FPPA14] S. Ferradans, N. Papadakis, G. Peyré, and J.-F. Aujol. Regularized Discrete Optimal Transport. *SIAM Journal on Imaging Sciences*, 7(3):1853–1882, 2014.

[FR13] S. Foucart and H. Rauhut. *A Mathematical Introduction to Compressive Sensing*. Springer, 2013.

[Gon42] V. Goncharov. Sur la Distribution des Cycles dans les Permutations. In *Doklady Akademii Nauk SSSR*, volume 35, pages 299–301, 1942.

[Has03] M. Hassani. Derangements and Applications. *Journal of Integer Sequences*, 6(2):3, 2003.

[Her09] G. T. Herman. *Fundamentals of Computerized Tomography: Image Reconstruction from Projections*. Springer Science & Business Media, 2009.

[HK99] G. T. Herman and A. Kuba. *Discrete Tomography: Foundations, Algorithms and Applications*, volume 61. Birkhäuser, 1999.

[HMRAR13] M. Habib, C. McDiarmid, J. Ramirez-Alfonsin, and B. Reed. *Probabilistic Methods for Algorithmic Discrete Mathematics*, volume 16. Springer Science & Business Media, 2013.

[JS03] F. Jarre and J. Stoer. *Optimierung*. Springer, 2003.

[Kan58] L. Kantorovitch. On the Translocation of Masses. *Management Science*, 5(1):1–4, 1958.

[KP18] J. Kuske and S. Petra. Performance Bounds for Co-/Sparse Box Constrained Signal Recovery. *An. St. Univ. Ovidius Constanta*, 26(1):22 pages, in print, 2018.

[KV08] B. Korte and J. Vygen. *Combinatorial Optimization*, volume 21 of *Algorithms and Combinatorics*. Springer, Berlin, 2008.

[LaV] LaVision GmbH. https://www.lavision.de/.

[LM79] P.-L. Lions and B. Mercier. Splitting Algorithms for the Sum of Two Nonlinear Operators. *SIAM Journal on Numerical Analysis*, 16(6):964–979, 1979.

[Loe06] G. Loeper. A Fully Nonlinear Version of the Incompressible Euler Equations: The Semigeostrophic System. *SIAM Journal on Mathematical Analysis*, 38(3):795–823, 2006.

[LS15] K. P. Lynch and F. Scarano. An Efficient and Accurate Approach to MTE-MART for Time-Resolved Tomographic PIV. *Experiments in Fluids*, 56(3):1–16, 2015.

[Nat95] B. K. Natarajan. Sparse Approximate Solutions to Linear Systems. *SIAM Journal on Computing*, 24(2):227–234, 1995.

[NBS10] M. Novara, K. J. Batenburg, and F. Scarano. Motion Tracking-Enhanced MART for Tomographic PIV. *Measurement Science and Technology*, 21(3):035401, 2010.

[Nyq28] H. Nyquist. Certain Topics in Telegraph Transmission Theory. 1928.

[PS09] S. Petra and C. Schnörr. TomoPIV meets Compressed Sensing. *Pure Mathematics and Applications*, 20(1-2):49 – 76, 2009.

[PS14] S. Petra and C. Schnörr. Average Case Recovery Analysis of Tomographic Compressive Sensing. *Linear Algebra and its Applications*, 441:168–198, 2014.

[PSS13] S. Petra, C. Schnörr, and A. Schröder. Critical Parameter Values and Reconstruction Properties of Discrete Tomography: Application to Experimental Fluid Dynamics. *Fundamenta Informaticae*, 125(3-4):285–312, 2013.

[Roc97] R. T. Rockafellar. *Convex Analysis*. Princeton University Press, 1997.

[RWWK13] M. Raffel, C. E. Willert, S. T. Wereley, and J. Kompenhans. *Particle Image Velocimetry: A Practical Guide*. Springer, 2013.

[Sca13] F. Scarano. Tomographic PIV: Principles and Practice. *Measurement Science and Technology*, 24(1):012001, 2013.

[SGS16] D. Schanz, S. Gesemann, and A. Schröder. Shake-The-Box: Lagrangian Particle Tracking at High Particle Image Densities. *Experiments in Fluids*, pages 57–70, 2016.

[Sha49] C. E. Shannon. Communication in the Presence of Noise. *Proceedings of the IRE*, 37(1):10–21, 1949.

[SKA15a] L. Saumier, B. Khouider, and M. Agueh. Optimal Transport for Particle Image Velocimetry. *Communications in Mathematical Sciences*, 13(1):269–296, 2015.

[SKA15b] L. Saumier, B. Khouider, and M. Agueh. Optimal Transport for Particle Image Velocimetry: Real Data and Postprocessing Algorithms. *SIAM Journal on Applied Mathematics*, 75(6):2495–2514, 2015.

[SL66] L. Shepp and S. Lloyd. Ordered Cycle Lengths in a Random Permutation. *Transactions of the American Mathematical Society*, 121(2):340–357, 1966.

[SW91] J. L. Stuart and J. R. Weaver. Matrices that Commute with a Permutation Matrix. *Linear Algebra and Its Applications*, 150:255–265, 1991.

[TBI97] L. N. Trefethen and D. Bau III. *Numerical Linear Algebra*, volume 50. SIAM, 1997.

[Tro15] J. A. Tropp. Convex Recovery of a Structured Signal from Independent Random Linear Measurements. In *Sampling Theory, a Renaissance*, pages 67–101. Springer, 2015.

[Vil03] C. Villani. *Topics in Optimal Transportation*. Number 58. American Mathematical Society, 2003.

[Vil08] C. Villani. *Optimal Transport: Old and New*, volume 338. Springer, 2008.

[Wie08] B. Wieneke. Volume Self-Calibration for 3D Particle Image Velocimetry. *Experiments in Fluids*, 45(4):549–556, 2008.

[Wie13] B. Wieneke. Iterative Reconstruction of Volumetric Particle Distribution. *Measurement Science and Technology*, 24(2):024008, 2013.

[WXT11] M. Wang, W. Xu, and A. Tang. A Unique "Nonnegative" Solution to an Underdetermined System: From Vectors to Matrices. *IEEE Transactions on Signal Processing*, 59(3):1007–1016, 2011.

[XY13] Y. Xu and W. Yin. A Block Coordinate Descent Method for Regularized Multiconvex Optimization with Applications to Nonnegative Tensor Factorization and Completion. *SIAM Jornal on Imaging Sciences*, 6(3):1758–1789, 2013.