

**Space-Time Tomographic Reconstruction
of Deforming Objects**

Dissertation by

Guangming Zang

In Partial Fulfillment of the Requirements

For the Degree of

Doctor of Philosophy

King Abdullah University of Science and Technology

Thuwal, Kingdom of Saudi Arabia

January, 2020

EXAMINATION COMMITTEE PAGE

The dissertation of Guangming Zang is approved by the examination committee

Committee Chairperson: Prof. Wolfgang Heidrich

Committee Co-Chair: Prof. Peter Wonka

Committee Member: Prof. Gilles Lubineau

Committee Member: Prof. Francesco De Carlo

©January, 2020

Guangming Zang

All Rights Reserved

ABSTRACT

Space-Time Tomographic Reconstruction of Deforming Objects

Guangming Zang

X-ray computed tomography (CT) is a popular imaging technique used for reconstructing volumetric properties for a large range of objects. Compared to traditional optical means, CT is a valuable tool for analyzing objects with interesting internal structure or complex geometries that are not accessible with. In this thesis, a variety of applications in computer vision and graphics of inverse problems using tomographic imaging modalities will be presented:

The first application focuses on the CT reconstruction with a specific emphasis on recovering thin 1D and 2D manifolds embedded in 3D volumes. To reconstruct such structures at resolutions below the Nyquist limit of the CT image sensor, we devise a new 3D structure tensor prior, which can be incorporated as a regularizer into more traditional proximal optimization methods for CT reconstruction. The second application is about space-time tomography: Through a combination of a new CT image acquisition strategy, a space-time tomographic image formation model, and an alternating, multi-scale solver, we achieve a general approach that can be used to analyze a wide range of dynamic phenomena. Based on the second application, the third one is aiming to improve the tomographic reconstruction of time-varying geometries undergoing faster, non-periodic deformations, by a warp-and-project strategy. Finally, with a physically plausible divergence-free prior for motion estimation, as well as a novel view synthesis technique, we present applications to dynamic fluid imaging (e.g., 4D soot imaging of a combustion process, a mixing fluid process, a fuel injection process, and view synthesis for visible light tomography), which further demonstrates the flexibility of our optimization frameworks.

ACKNOWLEDGEMENTS

I thank my PhD advisor, Prof. Dr. Wolfgang Heidrich, for his valuable guidance, scholarly inputs and consistent encouragement I received throughout the research work. This dissertation would definitely not have been possible without his continued patience, and endless support.

My additional gratitude goes to my co-advisor, Prof. Dr. Peter Wonka, for his valuable guidance and patient academic support that helped me realize my life long dream of completing a PhD degree as expected.

I am very grateful to my PhD committee member, Prof. Dr. Gilles Lubineau, for his consistent support for CT and Micro-CT facilities provided to carry out the research work, and sharing his expertise in mechanical engineering.

Furthermore, I would like to thank Prof. Dr. Francesco De Carlo deeply for joining my PhD committee. His very thoughtful comments and feedback, such as the suggestion to adapt the algorithms shown in this dissertation and apply on synchrotron facilities, are really encouraging.

I would also like to thank Dr. Ramzi Idoughi and Dr. Mohamed Aly, for many fruitful discussions on tomography and optimization. It is so enjoyable to work and collaborate with them. Dr. Ramzi Idoughi involved in all the projects presented in this dissertation, and I am really lucky to work on these interesting projects together with him during my PhD studies.

Thanks to Ran Tao for his help with CT and Micro-CT scanning. Thanks to Anthony Bennett, Jianguo Du, and Yuansi Tian for helping with the experiments set up in the dynamic fluid project.

TABLE OF CONTENTS

Examination Committee Page	2
Copyright	3
Abstract	4
Acknowledgements	5
List of Figures	9
List of Tables	11
1 Introduction	12
2 Background and Related Work	19
2.1 Proximal Algorithms	19
2.2 Dynamic Reconstruction	21
2.2.1 Dynamic 3D Surface Reconstruction	21
2.2.2 Dynamic Tomographic Reconstruction.	22
2.3 Computed Tomography in Graphics	24
2.4 Deformation and Motion Capture	25
3 Sparse View and Super-Resolution Tomography	28
3.1 Introduction	28
3.2 Structure Tensor Prior (STP)	29
3.2.1 Structure Tensor	29
3.2.2 Definition	30
3.2.3 Proximal Operator for STP	33
3.3 SART for the Data Term	36
3.3.1 Data Term Solver	36
3.3.2 SART	37
3.3.3 Definition	38
3.3.4 Algorithm	39

3.4	Implementation Details	41
3.5	Experiments	42
3.5.1	Sparse-View Reconstruction	42
3.5.2	Super Resolution Experiments	45
3.5.3	Segmentation Results	54
3.6	Conclusions	54
4	Space-Time Tomography	56
4.1	Introduction	56
4.2	Low Discrepancy View Sampling	56
4.3	Image Formation Model	60
4.4	Solver	62
4.4.1	Deformation Field Estimation	64
4.4.2	Volumetric Reconstruction	67
4.5	Experiments	70
4.5.1	Quantitative Evaluation	71
4.5.2	Qualitative Evaluation	76
4.6	Conclusion	88
5	Warp-and-Project Tomography	90
5.1	Introduction	90
5.2	Warp-and-Project Tomography	92
5.2.1	Image Formation Model	92
5.2.2	Full Optimization Problem	95
5.3	Experiments	104
5.4	Conclusion	116
6	Applications	120
6.1	Pipeline overview	122
6.2	4D Soot Imaging of a Combustion Process	125
6.3	4D Imaging of a Mixing Fluid Process	127
6.4	4D Imaging of a Fuel Injection Process	127
6.5	View Synthesis for Visible Light Tomography	129
6.6	Tomographic Reconstruction for Nanoscale Chip Data	132
7	Conclusion	134

LIST OF FIGURES

1.1	Super-resolution results	13
1.2	X-ray tomographic reconstruction of a wilting rose.	15
1.3	Warp-and-Project tomography.	17
3.1	A slice sample.	44
3.2	Comparison of iterative solvers.	46
3.3	Slice results for 3D zone plate.	47
3.4	RMSE results.	47
3.5	Representative slice visualization and comparison.	49
3.6	Sample slice.	49
3.7	The real flower and toothbrush reconstruction.	50
3.8	Comparison for reconstructed results.	51
3.9	Slice visualization for edge detection.	51
3.10	Segmentation results from different methods.	53
4.1	Acquisition strategy comparison.	58
4.2	Synthetic compressed deformation for a copper foam.	71
4.3	Synthetic rotated deformation for a copper foam.	72
4.4	Comparison of rose slices obtained after SART reconstruction.	76
4.5	Different types of objects and deformations used in the experiments.	77
4.6	Reconstruction obtained for the wilting rose dataset.	79
4.7	Comparison for rose dataset between SART-ROF and our method.	79
4.8	Comparison of three reconstruction approaches for different datasets.	80
4.9	Reconstruction obtained for the lentils/lupin seeds dataset.	81
4.10	Reconstruction obtained for the crystal sugar dataset.	81
4.11	The visualization of the sugar dataset.	82
4.12	Reconstruction obtained for the high viscosity fluid dataset.	83
4.13	Reconstruction obtained for the high viscosity fluid dataset.	84
4.14	Reconstruction obtained for the dried black mushrooms dataset.	85
4.15	Slice visualization and comparison for mushrooms data.	86
4.16	Reconstruction obtained for the rising dough dataset.	87

4.17 The visualization results of the dough dataset.	87
5.1 Diagram of our method.	94
5.2 Several frames from the fastest fluid animation.	105
5.3 The translation of the copper foam.	107
5.4 Absolute error for SART-ROF, ST-Tomography and ours.	108
5.5 Compression process of the copper foam.	109
5.6 Algorithm comparison for the compression of copper foam.	110
5.7 PSNR values of the volume reconstructions for each key frame.	111
5.8 Rising dough reconstructed by our method.	113
5.9 Comparison between SART-ROF, ST-tomography and Ours.	113
5.10 3D reconstructions of the dough dataset.	114
5.11 Reconstruction results for the porous rock dataset.	115
5.12 Rendering results for the rock from our method.	115
5.13 Re-hydration of dried tremella fuciformis.	116
5.14 Metal objects being dropped into a high viscosity liquid.	117
5.15 Pills dissolving in water.	118
5.16 Slice rendering of water absorption by Hydro-gel balls.	119
5.17 3D rendering of water absorption by Hydro-gel balls.	119
6.1 The captured projections for the soot dataset (low density).	121
6.2 The captured projections for the soot dataset (high density).	121
6.3 Reconstruction results for the low density soot.	126
6.4 Reconstruction results for the high density soot.	126
6.5 The set up for the high speed fluid imaging.	127
6.6 Captured spay data with different conditions	128
6.7 The set up for the injection experiments.	128
6.8 The visualization results.	128
6.9 The pipeline	130
6.10 Numerical results for different reconstruction methods.	130
6.11 The comparison for different methods.	131
6.12 Volume visualization for chip data	132
6.13 Slice visualization for chip data	133

LIST OF TABLES

3.1	Parameters for the X-ray CT data.	42
3.2	Parameters used with each method	43
3.3	PSNR and SSIM results from different reconstruction methods. . . .	52
3.4	Segmentations Results for different methods.	53
4.1	Calculated PSNR, and SSIM for different rotation velocities.	72
4.2	Interaction of the sampling strategy with our reconstruction method.	73
4.3	Calculated PSNR, and SSIM for different reconstruction methods. . .	75
4.4	Calculated PSNR, and SSIM for different compression velocities. . . .	75
4.5	Parameters used in the acquisition and for the optimization.	76
5.1	Numerical comparisons with ground truth data for different algorithms.	106
5.2	Parameters used in the acquisition and reconstruction.	112
5.3	Calculated PSNR, and SSIM for different reconstruction methods. . .	112

Chapter 1

Introduction

X-ray tomography is a popular imaging technique used for reconstructing volumetric properties for a large range of objects [1]. For example, it is used for industrial inspection, luggage inspection, research and development in mechanical engineering and material sciences, biomedical diagnosis and treatment, and it serves as an input to many computer vision algorithms, including methods for automatic segmentation, detection, and recognition.

This thesis presents four separate but closely related applications of inverse problems in tomography: the first is 3D super-resolution and sparse view tomography (Chapter 3), the second is space time reconstruction for dynamic phenomena (Chapter 4), the third one is the problem of reconstructing rapidly deforming objects with a warp and project strategy (Chapter 5), the fourth is the capture of physically plausible dynamic fluids (Chapter 6). The first three applications are based on X-ray CT scanning, while the fourth one utilizes traditional RGB cameras. All these four applications share a general numerical optimization framework (as shown in Equation 1.1) that allows the novel regularizers to be incorporated into the framework flexibly, depending on the different applications.

One important goal of CT scanning is to get the most out of the available equipment and to reconstruct as many details as possible. This calls for super-resolution reconstruction methods that can reconstruct volumes with higher resolution than the input scans, as well as methods that can reconstruct high resolution volumes using fewer projections (which is also called sparse-view reconstruction). While this is not

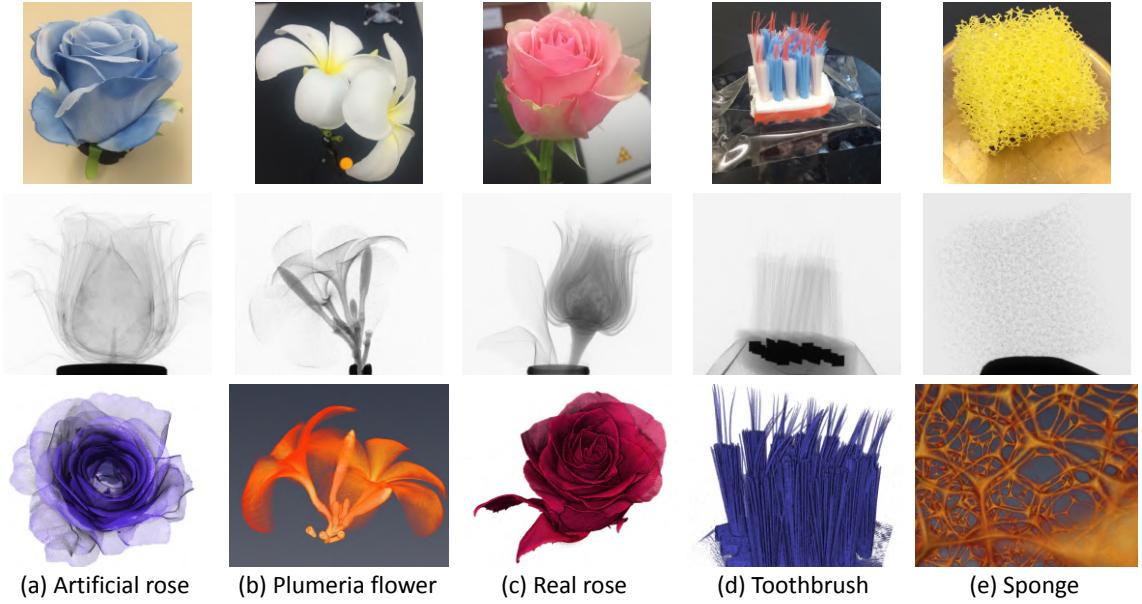


Figure 1.1: Five datasets with thin 2D (a-c) and 1D (d-e) structures embedded in 3D volumes are presented in Chapter 3. Top row: scanned objects. Middle row: representative projection images. Bottom row: rendering results of volumes reconstructed by our method.

possible for general objects, it can be achieved by making reasonable assumptions about the structure of the input volume. In Chapter 3, we propose a novel optimization framework that addresses super-resolution and sparse view reconstruction especially suited for volumes with thin sheets and tube-like structures. Specifically, we demonstrate that our framework is capable of reconstructing volumetric features smaller than the Nyquist limit in each individual projection, as well as better performance in the sparse-view acquisition setting.

State of the art robust CT reconstruction usually employs iterative methods [1, 2] and poses the problem as an optimization problem of the form

$$\min_{\mathbf{f}} \underbrace{f(\mathbf{f})}_{\text{data fidelity}} + \underbrace{g(\mathbf{Kf})}_{\text{regularizer}}, \quad (1.1)$$

where $\mathbf{f} \in \mathbb{R}^n$ is the unknown discrete 3D reconstruction volume or a single frame in a 4D space-time volume sequence, $f(\cdot)$ is the data fidelity term that measures

how well the volume fits the measured input projections, and is usually of the form $f(\mathbf{f}) = \|\mathbf{Af} - \mathbf{p}\|_2^2$, where \mathbf{A} describes the projection geometry, and \mathbf{p} represents the observed projection images. $g(\mathbf{Kf})$ is the regularizer consisting of a loss function $g(\cdot)$ and a linear operator \mathbf{K} that transforms the volume \mathbf{f} into a sparse domain (e.g. for Total Variation, $g(\cdot) = \|\cdot\|_1$ and \mathbf{K} is the volume gradient operator). Problem (1.1) is a general model, and can incorporate many noise models, e.g. Poisson [3] or Gaussian noise [4]; and regularizers, e.g. ℓ_1 or Total Variation (TV). Such optimization problems are commonly solved with proximal algorithms [5], which allow the decomposition of (1.1) into independent proximal operators, one for the linear least squares data term, and one for the non-linear regularizer.

The regularizer can be used to enforce specific prior information about the reconstructed volume. Our major contribution in Chapter 3 is to show that enforcing sparsity on the eigenvalues of the 3D structure tensor allows for super-resolved reconstruction of thin structures such as thin sheets or tubes, see for example Fig. 1.1. The intuition is that the 3D structure tensor should have two zero (or extremely small) eigenvalues on a 2D manifold embedded into a 3D volume, since the volume will only vary along the normal direction. Likewise, for curves embedded in 3D, one of the eigenvalues is expected to be zero (or extremely small).

The linear least squares problem in the data term requires a matrix-free solver in order to control memory consumption, and Conjugate Gradients is frequently used for this purpose [6]. In this thesis we show that using the Simultaneous Algebraic Reconstruction Technique (SART [7]) for this problem yields better results, especially in reconstructions from a sparse numbers of projections. While SART has historically played an important role in solving the unregularized CT problem, we demonstrate how to use it for solving the data term proximal operator, which to our knowledge has not been done before.

Besides 3D (i.e. spatial) tomography, which focuses on the geometry recovery,

capture of 3D and dynamic phenomena has long been a topic of interest in computer graphics. While much of the work in this area has been based on traditional computer vision methods with either passive or active illumination [8, 9], tomographic methods have also played an important role in computer graphics research [10, 11, 12, 13, 14]. In engineering and medicine, x-ray computed tomography has become a valuable tool

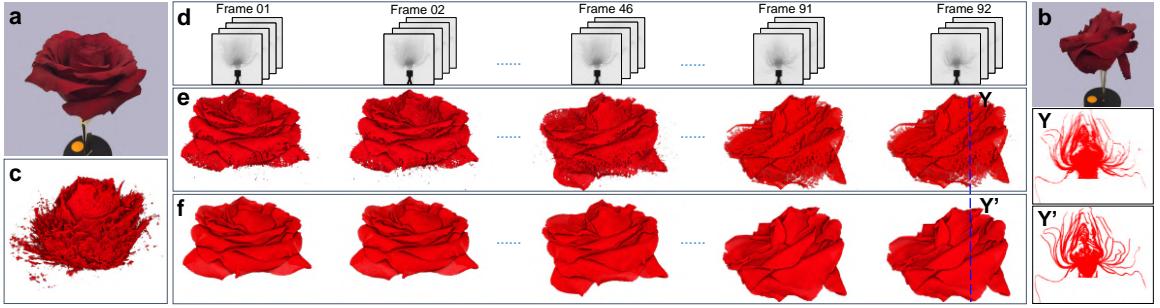


Figure 1.2: X-ray tomographic reconstruction of a rose undergoing significant wilting during the scan due to loss of water. Images (a) and (b) show photographs of the rose directly before and directly after the scanning process. Traditional CT reconstruction (c) from all 5520 projections in the scan sequence show significant distortions due to misalignment of features. When grouping the projections into 92 frames of 60 projections each (d), the deformation over each frame becomes negligible, but now the number of projections per frame is insufficient for high-quality reconstruction of the corresponding volumes (e, \mathbf{Y}). By comparison, our full space-time reconstruction algorithm yields a time sequence of highly detailed volumes for different time steps (f, \mathbf{Y}'). More details can be found in Chapter 4.

for its ability to reveal internal structures in objects that cannot be sensed by optical means. Tomographic reconstruction in graphics has been primarily focused on visible light tomography [10, 15, 16], although there are a few examples of using x-ray CTs as well [11, 12]. While visible light tomography is inherently limited to transparent materials, it does have the advantage of using regular cameras and light sources, which makes it feasible to construct multi-camera setups that can be used to capture dynamic phenomena such as fluids (e.g. [17, 18, 19, 20, 21, 14]). By comparison, x-ray CT is typically limited to a single source and sensor array or a very small number of such sources and arrays, and the required diversity of view points is created through mechanical motion of either the sample (typical of industrial CTs) or the source-

sensor assembly (as in medical CTs). Unfortunately, the need for mechanical motion has so far hindered the ability to use x-ray CTs for imaging dynamic phenomena or continuously changing geometries, except in some very specific setting such as objects that undergo periodic motion (e.g. beating hearts [22]).

In this thesis (Chapter 4), we aim to generalize tomographic reconstruction to the 4-dimensional case of space-time reconstruction of continuously deforming objects and phenomena. We target the general case of non-periodic motion, though we do assume relatively slow, smooth motion fields. Specifically, we assume that the deformation is negligible when considering a small number of projections (e.g 10-60) acquired in sequence, but that the motion is significant over the time it takes to acquire the hundreds or even thousands of projections needed to scan the whole object at a high spatial resolution.

Figure 1.2 shows an example of such a setting. The scanned object is a rose that undergoes significant deformation due to loss of water during the scanning process. This deformation makes it impossible to reconstruct a detailed volume using traditional 3D tomography methods: reconstructions from short sequences of projections are severely under-determined and thus lack detail, while reconstructions from large sequences of frames fail due to misalignment of the geometry. On the other hand, our 4D space-time tomographic reconstruction is able to reconstruct not just a single volume but a full-time sequence of highly detailed volumes. It also estimates the deformation fields between these frames which allows our method to be used for a full analysis of the wilting process.

Based on the space time tomography, in Chapter 5, the assumption that deformation is negligible when considering a small number of projections is then relaxed by leveraging a novel warp and project strategy. By applying this method, unprecedented reconstruction quality can be obtained for rapidly deforming object, as shown in Figure 1.3.

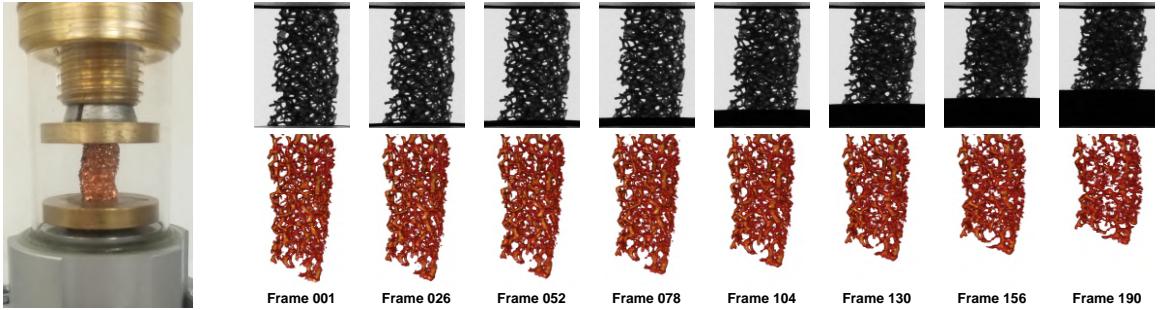


Figure 1.3: In Chapter 5, we introduce a novel CT reconstruction method for objects that undergo rapid deformation during the scan. Shown here is a copper foam crumpling under a compressive force during the scan. The whole complex animation is reconstructed using only 192 projection images that all correspond to different deformation states of the foam.

Beyond these points, the specific contributions made by this thesis are:

- A new CT image acquisition strategy based on low-discrepancy sampling, which improves the distribution of projection angles over time.
- A new image formation model for non-parametric 4D (space-time) tomographic reconstruction, together with the appropriate regularization strategies for tackling the ill-posed inverse problem.
- A matrix-free, multi-scale optimization framework for solving this inverse problem. With this flexible framework, different powerful priors for different tasks, can be incorporated easily.
- By applying the methods in this thesis on extensive real data from a range of different application scenarios, we demonstrate that tomography techniques are quite general and powerful tools in solving inverse problems.

In the remainder of this thesis, Chapter 2 gives a brief background for CT and related work. Next, Chapter 3 introduces sparse view and super-resolution tomography. The framework for space time tomography is then presented in Chapter 4. In Chapter 5, the warp and project tomography will be introduced. Chapter 6 discusses the applications of dynamic fluid imaging that based on the optimization frameworks described in Chapter 4 and Chapter 5. The thesis concludes with some closing remarks

in Chapter 7.

The content of Chapter 3, Chapter 4, and Chapter 5 of this thesis is mainly based upon three publications:

- [23] G. Zang, M. Aly, R. Idoughi, P. Wonka, and W. Heidrich. Super-Resolution and Sparse View CT Reconstruction. European Conference on Computer Vision (ECCV), 2018.
- [24] G. Zang, R. Idoughi, R. Tao, G. Lubineau, P. Wonka, and W. Heidrich. Space-time Tomography for Continuously Deforming Objects. ACM Trans. Graph.(Proc. SIGGRAPH), 37(4), 2018.
- [25] G. Zang, R. Idoughi, R. Tao, G. Lubineau, P. Wonka, and W. Heidrich. Warp-and-Project Tomography for Rapidly Deforming Objects. ACM Trans. Graph. (Proc. SIGGRAPH), 38(4), 2019.

Chapter 2

Background and Related Work

2.1 Proximal Algorithms

With the CT reconstruction problem expressed as an optimization problem (1.1), we turn to the question of finding appropriate solvers. Like several recent approaches, we rely on proximal algorithms [5], namely the first-order primal-dual algorithm proposed by Chambolle and Pock [26] (henceforth referred to as the CP algorithm). Proximal algorithms are able to solve complex optimization problems by splitting them into several smaller and easier sub-problems, that are solved independently, and then combined to find a solution to the original problem.

These simple sub-problems take the form of *proximal operators* [5]:

$$\mathbf{prox}_{\zeta h}(\mathbf{q}) \triangleq \operatorname{argmin}_{\mathbf{f}} h(\mathbf{f}) + \frac{1}{2\zeta} \|\mathbf{f} - \mathbf{q}\|_2^2, \quad (2.1)$$

where $\mathbf{q} \in \mathbb{R}^n$ is the input to the operator and $\zeta \in \mathbb{R}$ is a weighting parameter. For the CP algorithm to work, we need to determine and implement two proximal operators: The proximal operator for the data term: $\mathbf{prox}_{\tau f}(\mathbf{q})$, and the proximal operator for the convex conjugate [27] function $g^*(\cdot)$ of $g(\cdot)$ defined as: $\mathbf{prox}_{\mu g^*}(\mathbf{q})$. By using different regularization functions $g(\cdot)$ and matrices \mathbf{K} , we can plug in different priors based on different models of what the reconstructed volume should look like. Here We present the definition and the proximal operator of Anisotropic Total Variation (ATV), Isotropic Total Variation (ITV), and Sum of Absolute Differences (SAD), respectively:

- Anisotropic TV (ATV): This prior is defined as

$$h(\mathbf{f}) = \sum_{ijk} |\mathbf{f}_{i+1,j,k} - \mathbf{f}_{i,j,k}| + |\mathbf{f}_{i,j+1,k} - \mathbf{f}_{i,j,k}| + |\mathbf{f}_{i,j,k+1} - \mathbf{f}_{i,j,k}|, \quad (2.2)$$

where $\mathbf{f}_{i,j,k}$ is the voxel value at position (i, j, k) . This can be represented as $\tilde{g}(\mathbf{Kx})$, where $\tilde{g}(\cdot) = \|\cdot\|_1$ is the ℓ_1 norm and $\mathbf{K} = \mathbf{D} \in \mathbb{R}^{3n \times n}$ is the forward difference matrix. The proximal operator of $\tilde{g}^*(\cdot)$ can be shown to be [28]

$$\text{prox}_{\mu\tilde{g}^*}(\mathbf{q}) = \Pi_{B_\infty}(\mathbf{q}) = \begin{cases} 1 & u > 1 \\ q & |q| \leq 1 \\ -1 & u < -1 \end{cases}, \quad (2.3)$$

where the operations are component-wise, and is equivalent to the projection on the unit ball B_∞ of the ℓ_∞ norm. Note that we do not need to store the matrix D , and multiplication by D (computing the gradient) or by D^T (computing the divergence) can be efficiently computed on-the-fly [29]. Also note that $D^T D$ is the Laplace operator.

- Isotropic TV (ITV): This prior is defined as

$$h(\mathbf{f}) = \sum_{ijk} \sqrt{|\mathbf{f}_{i+1,j,k} - \mathbf{f}_{i,j,k}|^2 + |\mathbf{f}_{i,j+1,k} - \mathbf{f}_{i,j,k}|^2 + |\mathbf{f}_{i,j,k+1} - \mathbf{f}_{i,j,k}|^2},$$

where it sums the magnitude of the gradient at each voxel. Using the forward matrix D above and defining a new matrix $\mathbf{E} \in \mathbb{R}^{3n \times n}$ that denotes the positions of the forward differences [29], we can define the function $h(\mathbf{f})$ as a norm $\|\mathbf{q}\|_E$ for $\mathbf{q} = \mathbf{Df} \in \mathbb{R}^{3n}$ defined as

$$\|w\|_{\mathbf{E}} = \|\sqrt{\mathbf{E}^T w^2}\|_1 = \sum_v \|w^v\|_2,$$

where the square root and square functions are component-wise, and w^v is the gradient at voxel $v = (i, j, k)$. Now we can express the ITV prior $h(\mathbf{f})$ in terms of the $\|\mathbf{q}\|_{\mathbf{E}}$ norm as

$$h(x) = \|\mathbf{D}\mathbf{f}\|_E = \tilde{g}(\mathbf{D}\mathbf{f}) \text{ where } \tilde{g}(\mathbf{q}) = \|\mathbf{q}\|_{\mathbf{E}}.$$

The proximal operator for $\tilde{g}^*(\cdot)$ can be shown to be [29]

$$\text{prox}_{\mu\tilde{g}^*}(\mathbf{q}) = \Pi_{B^*}(\mathbf{q}) = \frac{\mathbf{q}}{\mathbf{E} \max \left(\sqrt{\mathbf{E}^T \mathbf{q}^2}, \mathbf{1} \right)} \quad (2.4)$$

which is the projection on the unit ball B^* of the dual norm $\|\mathbf{q}\|_{\mathbf{E}^*}$, and where the division and max operations are performed component-wise.

- Sum of Absolute Differences (SAD): This prior is defined as

$$h(\mathbf{f}) = \sum_{ijk} \sum_{\mathbf{f}_n \in N(\mathbf{f}_{i,j,k})} |\mathbf{f}_n - \mathbf{f}_{i,j,k}|, \quad (2.5)$$

where $N(\mathbf{f}_{i,j,k})$ is the 3×3 neighborhood around voxel $\mathbf{f}_{i,j,k}$ (excluding voxel $\mathbf{f}_{i,j,k}$ itself). It can be seen as an extension to the ATV prior, just with a different matrix \mathbf{D} where more edges are considered for every voxel instead of just three. Hence its proximal operator is similar to Equation 2.3. It has been shown [21] to produce excellent results in stochastic tomography reconstruction.

2.2 Dynamic Reconstruction

2.2.1 Dynamic 3D Surface Reconstruction

Surface reconstruction from color and depth sensors has been an active research topic in computer graphics. The state-of-the-art techniques allow real-time 4D reconstruction of non-rigidly deforming scenes using one [30, 31] or more [9] or RGB-D cameras.

These approaches combine surface reconstruction over time with a parameterized nonrigid motion tracking in an optimization framework. Priors on motion are also integrated. In [32, 16] a 3D scanner is used to capture 4D point cloud data, which is used respectively to reconstruct the plant growth and the flower blooming process. Furthermore, some approaches were developed for the capture of time-varying fluid surfaces, where specific priors to the fluid field are usually incorporated [33]. However, these surface-based methods cannot reconstruct internal structures of objects or occluded geometry.

2.2.2 Dynamic Tomographic Reconstruction.

Numerical Methods for Tomographic Reconstruction. CT reconstruction methods can be grouped into two families: direct methods and iterative methods [1, 2]. Direct methods, like the filtered back-projection (FDK) algorithm [34], rely on the use of the Radon transform and its inverse. On the other hand, iterative methods propose to solve the discrete formulation of the reconstruction problem [2]. This approach involves very large matrices, which makes the memory consumption a key concern. The Simultaneous Algebraic Reconstruction Technique (SART), proposed by Andersen et al. [7], is one of the most efficient iterative reconstruction methods. It is based on the projection method of Kaczmarz, where the reconstructed volume is updated for each scan view. This approach has modest memory requirements, and yields better reconstruction results than the FDK algorithm when the number of used projections is limited [35]. Another advantage of iterative methods is their flexibility in incorporating prior information into the reconstruction process. For example, a total variation (TV) prior has been used for tomography reconstruction [36, 6].

Dynamic tomographic reconstruction. Dynamic tomographic reconstruction is a challenging task, which becomes unavoidable in situations where the scan target

deforms or degrades during the scanning process in a way that cannot be controlled or eliminated. This also opens the door to using of X-ray CT imaging for studies of dynamic phenomena, where the motion itself is the primary interest. Thus, several strategies have been proposed deal with dynamic objects.

In the medical field, the motion of the scanned organs is often periodic, like for the heart or the lungs. Several methods have been based on this observation to provide a dynamic reconstruction of the heart or the lungs [37, 38]. Such methods reconstruct each phase of the motion cycle independently, by using only projections belonging to the same phase. This requires a large number of projections to cover all the phases of the cycle, which results in a higher radiation dose for the patient. In addition, the obtained reconstruction presents clinically prohibitive artifacts [39]. To overcome these issues, 4D iterative reconstructions have been proposed [39, 22]. These methods employed either a spatial and temporal total variation regularization or an optical flow based registration between successive phases. However, the quality of the obtained reconstructions is still impacted by the low number of projections for each phase. Furthermore, a considerable effort has been made in this field to estimate the motion of the scanned object during the acquisition process [40, 41, 42]. Then the retrieved deformation field is employed to correct the reconstructed volumes. Often, this approach requires the knowledge of the initial state of the reconstructed volume, which will be used as a reference to estimate the motion.

On the hardware side, very fast CT of hundreds of projections per second is enabled by bright X-ray sources such as synchrotrons [43]. This paves the way for mechanical engineering and material science to obtain a better understanding of some dynamic processes. Indeed, the CT and micro-CT devices are commonly used in these fields to study dynamic experiments like the compression of a foam [44, 43], fatigue cracks [45, 46] or fluid flow in porous media [47, 48, 49]. However, even with this fast hardware, the acquisition time is still three orders of magnitude too slow for true

video rate volume reconstruction using traditional algorithms, and so better space-time reconstruction algorithms are still highly desirable.

As a result, many studies use a stop-motion style acquisition of static states representing a supposedly continuous deformation [46, 50, 45]. The motion field can then be found by digital volume correlation (DVC, essentially 3D optical flow). Some variants of the DVC algorithm have been proposed to deal with continuous deformations, that occur during the scans. One improvement is made by projecting the volumes on a set of defined basis functions (e.g. finite element basis) [51]. Then the DVC is applied only on these basis functions, in order to speed up the estimation of the motion and reduce the degrees of freedom. Another variant is to use a projection-based DVC (P-DVC) to estimate the deformation of a known reference shape (template) from only few projections [52, 53]. Finally, Jailin et al. [54] proposed a combination between a variant of the ART reconstruction and the basis functions P-DVC approach to reconstruct a deforming volume and retrieve its motion, in a multi-scale scheme. The main limitation of these approaches is their specificity to certain types of deforming objects, where it is easy to define the basis function and track the motion through them. For some phenomena such as rising dough, which undergoes significant topological change, these methods will fail to reconstruct the volumes.

2.3 Computed Tomography in Graphics

CT in vision and graphics has been successfully employed for 3D reconstruction and rendering. In order to obtain a volumetric rendering of trees, Reche-Martinez et al. [10] introduced an approach that combines a volumetric opacity estimation with a view-dependent texturing from different photographs. For fluid imaging, computed tomography allows the 3D reconstruction of flames [17, 19], the capture of non-stationary gas flows [20], the 3D imaging of turbulent fluids [21, 14], as well as the 3D velocity estimation inside a fluid [18]. After suspending a transparent object in-

side a fluid with the same refractive index, Trifonov et al. [15] realized a tomographic reconstruction of that object. Tomography was also introduced for other applications like 3D displays [55, 56], where multi-layered systems are employed. X-ray computed tomography has also been used for some applications in graphics, including the utilization of micro CT images to build volumetric appearance models for fabrics [11]. In addition, X-ray tomography has proven to be an excellent tool for flower modeling [13] or flower structure analysis [12].

The main interest of using X-ray CT in computer graphics is to capture the internal structures of opaque objects [11, 57], as well as to retrieve complicated surfaces with occlusions like flowers [12, 13].

2.4 Deformation and Motion Capture

Discrete Deformations. Refer to the situation where the target can be scanned before and after it undergoes a deformation, but where either no deformation is taking place during the scan, or the time scale of the deformation is very long compared to the scan time, such as in recent work by Kato et al. [58] on imaging the growing process of plants. This scenario often occurs in mechanical engineering and material science, where the primary goal is to measure the displacement through of a target structure before and after mechanical loading, in order to retrieve the strain field. To estimate this strain field, Digital Volume Correlation (DVC), essentially a 3D variant of optical flow [59], is widely employed [60, 61, 46, 50, 45].

Periodic Deformations. For medical CT imaging, the deformation time scale is smaller than the scan time, however, the motion is often cyclic, with a period corresponding to a heart beat or respiratory cycle. Based on this observation, several approaches proposed a 4D reconstruction of the heart or thorax [37, 38], by regrouping the scans into different phases of the cycle and then reconstructing each phase

independently, using only the relevant x-ray projections. In order to get a good reconstruction for all phases, the total number of scans should be sufficiently large, which mandates a longer irradiation time and a higher radiation dose for the patient. To address this issue, Mory et al. [22] proposed a 4D iterative reconstruction with spatial and temporal total variation regularization. This technique is applied on all projections together. The quality of the yielded reconstruction still suffers from the angular undersampling for each cardiac phase. Another recent approach consists of estimating and iteratively compensating the motion that occurs during a cycle, in order to obtain a 4D reconstruction [42]. The motion estimation is realized using a pair-wise 3D/3D image registration. A major limitation of this approach lies in the low accuracy of the generated 3D volumes representative of each cardiac phase. Finally, one common limitation of all presented 4D CT reconstruction approaches is their reduced applicability to specific deformation phenomenon.

General Deformations. There has been some limited work on handling general deformations during a tomographic scan, and obtaining a true 4D tomographic reconstruction. A first approach consists of compensating for known time-dependent transformations of the scanned object [62]. Another approach is to project the volume densities into a finite element basis, and then track deformations of the finite elements over time [63, 51]. This can be used in digital volume correlation to speed up the distortion estimation between two static frames, but such concepts can also be used as regularizers in continuously deforming objects. Unfortunately, finite elements are very cumbersome to fit to geometries with complex topologies, such as the rose in Figure 1.2. Recent work like e.g [52, 53] first scans a high-quality template geometry, and then requires only a few projections per deformation state to track the motion. This approach will fail if the object is not stationary long enough to obtain the template. Our method, by comparison, is non-parametric and does not require a

template.

Chapter 3

Sparse View and Super-Resolution Tomography

3.1 Introduction

In this chapter, we present a flexible framework for robust computed tomography (CT) reconstruction with a specific emphasis on recovering thin 1D and 2D manifolds embedded in 3D volumes. To reconstruct such structures at resolutions below the Nyquist limit of the CT image sensor, we devise a new 3D structure tensor prior, which can be incorporated as a regularizer into more traditional proximal optimization methods for CT reconstruction. As a second, smaller contribution, we also show that when using such a proximal reconstruction framework, it is beneficial to employ the Simultaneous Algebraic Reconstruction Technique (SART) instead of the commonly used Conjugate Gradient (CG) method in the solution of the data term proximal operator. We show empirically that CG often does not converge to the global optimum for tomography problem even though the underlying problem is convex. We demonstrate that using SART provides better reconstruction results in sparse-view settings using fewer projection images. We provide extensive experimental results for both contributions on both simulated and real data.

The main components of our proximal framework are the regularization term and the data term. We first introduce the STP prior and its proximal operator, then how to use SART as a solver for the data term is presented.

3.2 Structure Tensor Prior (STP)

3.2.1 Structure Tensor

The structure tensor [64] $S_K(\mathbf{f}_i) \in \mathbb{S}_+^3$ for a 3D volume at voxel i is a 3×3 positive semi-definite matrix that captures the local structure around a voxel, and is defined as

$$\begin{aligned} \mathbf{S}_K(\mathbf{f}_i) &= \sum_{j \in \mathcal{N}(q_i)} \mathbf{K}(q_j - q_i) \begin{bmatrix} (\delta_j^1)^2 & \delta_j^1 \delta_j^2 & \delta_j^1 \delta_j^3 \\ \delta_j^1 \delta_j^2 & (\delta_j^2)^2 & \delta_j^2 \delta_j^3 \\ \delta_j^1 \delta_j^3 & \delta_j^2 \delta_j^3 & (\delta_j^3)^2 \end{bmatrix} \\ &= \sum_{j \in \mathcal{N}(q_i)} \mathbf{K}(q_j - q_i) (\nabla \mathbf{f}_j \nabla \mathbf{f}_j^T), \end{aligned} \quad (3.1)$$

where $q_i = [i_1, i_2, i_3]^T \in \mathbb{R}^3$ is the coordinates of voxel i and $\mathbf{f}_i = \mathbf{f}_{i_1, i_2, i_3}$ is the voxel value, $\mathbf{K}(q_j - q_i) : \mathbb{R}^3 \rightarrow \mathbb{R}$ is a 3D rotationally-symmetric smoothing kernel that down-weights the contributions of voxel j in the set $\mathcal{N}(q_i)$ of the l neighbors of the voxel i and $\nabla \mathbf{f}_j \in \mathbb{R}^3$ is the local gradient at voxel j . $\delta_j^k = \nabla^k \mathbf{f}_j$ is its k^{th} component

$$\nabla \mathbf{f}_j = \begin{bmatrix} \nabla^1 \mathbf{f}_j \\ \nabla^2 \mathbf{f}_j \\ \nabla^3 \mathbf{f}_j \end{bmatrix} = \begin{bmatrix} \mathbf{f}_{j_1+1, j_2, j_3} - \mathbf{f}_{j_1, j_2, j_3} \\ \mathbf{f}_{j_1, j_2+1, j_3} - \mathbf{f}_{j_1, j_2, j_3} \\ \mathbf{f}_{j_1, j_2, j_3+1} - \mathbf{f}_{j_1, j_2, j_3} \end{bmatrix}. \quad (3.2)$$

So we can regard the structure tensor as a weighted average of the outer product of the local gradients at the neighborhood of the voxel.

The eigenvalue decomposition of the structure tensor $\mathbf{S}_K(\mathbf{f}_i)$ gives an idea about the neighborhood. Let $\lambda_1 \geq \lambda_2 \geq \lambda_3$ be the three eigenvalues of the structure matrix [65]. We have three cases:

1. $\lambda_1 \gg \lambda_2 \approx \lambda_3$: the area around the voxel is sheet-like (a surface in 3D), in which case we have one large eigenvalue and two small ones.

2. $\lambda_1 \approx \lambda_2 \gg \lambda_3$: the area around the voxel is line-like (or resembles a tube or filament), in which case we have two large eigenvalue and a small one.
3. $\lambda_1 \approx \lambda_2 \approx \lambda_3$: the area around the voxel is isotropic, in which case we have three almost equal eigenvalues. It might be that it is a constant area in which case the eigenvalues are very small, or that the changes are equal in all directions (an isotropic region) in which case they might have larger values.

3.2.2 Definition

The STP regularizer was introduced by [66, 65]. It includes the standard TV as a special case, when the smoothing kernel is a Dirac delta i.e. it is a *local* structure tensor at each voxel [65]. Intuitively, the STP tries to estimate the volume such that its structure tensor is low rank, by minimizing the deviation of voxel values in the region around it. We will introduce the STP and develop its solver by extending it from the case of images in [66, 65] to 3D volumes and by employing more efficient proximal algorithms for its computation.

The STP at a voxel i is defined as the ℓ_p norm of the square roots of the eigenvalues of the structure tensor $\mathbf{S}_K(\mathbf{f}_i)$ defined in Eq. (3.1). Let $\Lambda(\mathbf{S}_K(\mathbf{f}_i)) \in \mathbb{R}^3$ be the vector of eigenvalues of $\mathbf{S}_K(\mathbf{f}_i)$:

$$STP_p(\mathbf{f}_i) = \|\sqrt{\Lambda(\mathbf{S}_K(\mathbf{f}_i))}\|_p = \left(\sum_{j=1}^3 \left(\sqrt{\lambda_j} \right)^p \right)^{\frac{1}{p}}. \quad (3.3)$$

In the case when the kernel is the Dirac delta $K(q) = \delta(q)$, the STP becomes the standard isotropic TV regularizer i.e. the ℓ_2 norm of the gradient vector [65] because the structure tensor simplifies to the outer product of the gradient

$$\mathbf{S}_\delta(\mathbf{f}_i) = \nabla \mathbf{f}_i \nabla \mathbf{f}_i^T$$

which has a rank of 1 and only one non-zero eigenvalue λ_1 whose value equals to the gradient magnitude square (its trace):

$$STP_p(\mathbf{f}_i) = \sqrt{\lambda_1} = \|\nabla \mathbf{f}_i\|_2.$$

In general, however, the structure tensor will aggregate information in the neighborhood of the voxel that will help in having a better regularization of the volume.

Next, we will define how to represent the STP in a form that fits Eq. 1.1 in Chapter 1. Define the “patch-based Jacobian” [65] as a linear map $\mathbf{J}_K : \mathbb{R}^n \rightarrow \mathbb{R}^{nl \times 3}$ between the space of volumes and a set of weighted gradients that are computed from the l -neighborhood of each of the n voxels. We can write the patch-based Jacobian at voxel i as $\mathbf{J}_K(\mathbf{f}_i) \in \mathbb{R}^{l \times 3}$ by stacking the weighted local gradients side-by-side:

$$\mathbf{J}_K(\mathbf{f}_i) = \begin{bmatrix} \kappa_{j_1} \nabla \mathbf{f}_{j_1} & \cdots & \kappa_{j_l} \nabla \mathbf{f}_{j_l} \end{bmatrix}^T \in \mathbb{R}^{l \times 3}, \quad (3.4)$$

where $\{j_1, \dots, j_l\} = \mathcal{N}(q_i)$ denotes the indices of the neighbors of voxel i (including i itself), and $\kappa_{j_k} = \sqrt{K(q_i - q_{j_k})}$. The patch-based Jacobian for the whole volume $\mathbf{J}_K \mathbf{f} \in \mathbb{R}^{nl \times 3}$ is now formed by stacking "local" components $\mathbf{J}_K(\mathbf{f}_i)$ on top of each other

$$\mathbf{J}_K \mathbf{f} = \begin{bmatrix} \mathbf{J}_K(\mathbf{f}_1) \\ \vdots \\ \mathbf{J}_K(\mathbf{f}_n) \end{bmatrix} \in \mathbb{R}^{nl \times 3}. \quad (3.5)$$

Using this linear operator \mathbf{J}_K , Equation 3.1 can be rewritten as follows:

$$\mathbf{S}_K(\mathbf{f}) = \mathbf{J}_K(\mathbf{f})^T \mathbf{J}_K(\mathbf{f}), \quad (3.6)$$

which means that the singular values of $\mathbf{J}_K(\mathbf{f}_i)$ are actually equal to the square root

of the eigenvalues of $\mathbf{S}_K(\mathbf{f}_i)$

$$\sigma(\mathbf{J}_K(\mathbf{f}_i)) = \sqrt{\Lambda(\mathbf{S}_K(\mathbf{f}_i))}, \quad (3.7)$$

where $\sigma(\mathbf{J}_K(\mathbf{f}_i)) \in \mathbb{R}_+^3$ is the vector of singular values of patch-based Jacobian $\mathbf{J}_K(\mathbf{f}_i)$.

From Eq. (3.3) and (3.7) we get the definition of STP_p as

$$STP_p(\mathbf{f}) = \sum_{i=1}^n \|\mathbf{J}_K(\mathbf{f}_i)\|_{\mathcal{S}_p}, \quad (3.8)$$

where $\|B\|_{\mathcal{S}_p}$ is the p -Schatten norm of B i.e. the ℓ_p norm of its singular values

$$\|B\|_{\mathcal{S}_p} = \|\sigma(B)\|_p. \quad (3.9)$$

There are usually three options for \mathcal{S}_p [28]:

1. $p = 1$ is equivalent to the *nuclear norm* i.e. the sum of singular values of the matrix:

$$\mathcal{S}_1(B) = \sum_i |\sigma_i(B)| = \|B\|_*.$$

2. $p = 2$ is equivalent to the *Frobenius norm*:

$$\mathcal{S}_2(B) = \sqrt{\sum_i \sigma_i^2(B)} = \|B\|_F.$$

3. $p = \infty$ is equivalent to the *spectral norm* i.e.

$$\mathcal{S}_\infty(B) = \max_i |\sigma_i(B)| = \|B\|_2.$$

Now we can write this regularizer in a more compact *compound norm*:

$$STP_p(\mathbf{f}) = \|\mathbf{J}_K \mathbf{f}\|_{1,p}$$

where the $(1, p)$ norm for a matrix $\mathbf{J} = \mathbf{J}_K \mathbf{x} \in \mathbb{R}^{nl \times 3}$ is defined as

$$\|\mathbf{J}\|_{1,p} = \|\mathbf{J}_K \mathbf{f}\|_{1,p} = \sum_{i=1}^n \|\mathbf{J}_i\|_{\mathcal{S}_p}, \quad (3.10)$$

where $\mathbf{J}_i \in \mathbb{R}^{l \times 3}$ represents the patch-based Jacobian at some voxel i .

Now, we can define the regularizer function $g(\cdot)$ as

$$g(\mathbf{J}_K \mathbf{f}) = \lambda \|\mathbf{J}\|_{1,p}. \quad (3.11)$$

3.2.3 Proximal Operator for STP

To solve the reconstruction problem we need to solve

$$\min_{\mathbf{f}} \|\mathbf{A}\mathbf{f} - \mathbf{p}\|_2^2 + \lambda S T P_p(\mathbf{f}) \equiv \min_{\mathbf{f}} \|\mathbf{A}\mathbf{f} - \mathbf{p}\|_2^2 + \lambda \|\mathbf{J}_K \mathbf{f}\|_{1,p}, \quad (3.12)$$

where

$$f(\mathbf{f}) = \|\mathbf{A}\mathbf{f} - \mathbf{p}\|_2^2 \quad (3.13)$$

represents the ℓ_2 data fidelity term assuming Gaussian measurement noise, and

$$g(\mathbf{K}\mathbf{f}) = \lambda S T P_p(\mathbf{f}) \quad (3.14)$$

is the regularization term where λ is the trade-off parameter with linear mapping $\mathbf{K} = \mathbf{J}_K$. We will first start with the $(1, p)$ norm, whose dual norm is the (∞, q) [67] defined as

$$\|\mathbf{J}\|_{\infty,q} = \max_{i=1 \dots n} \|\mathbf{J}_i\|_q \quad \forall \mathbf{J} \in \mathbb{R}^{nl \times 3} \quad (3.15)$$

with q such that $\frac{1}{p} + \frac{1}{q} = 1$. Now we can write (3.14) as in [68]

$$\|\mathbf{J}\|_{1,p} = \max_{\mathbf{H} \in \mathcal{B}_{\infty,q}} \langle \mathbf{H}, \mathbf{J} \rangle_{\mathbb{R}^{nl \times 3}} \quad (3.16)$$

where

$$\langle \mathbf{H}, \mathbf{J} \rangle_{\mathbb{R}^{nl \times 3}} = \sum_i \text{tr}(\mathbf{H}_i^T \mathbf{J}_i) \quad (3.17)$$

is the inner product in $\mathbb{R}^{nl \times 3}$ that induces the norm

$$\|\mathbf{H}\|_{\mathbb{R}^{nl \times 3}}^2 = \sqrt{\langle \mathbf{H}, \mathbf{H} \rangle_{\mathbb{R}^{nl \times 3}}}, \quad (3.18)$$

and $\mathcal{B}_{\infty,q}$ is the unit ball for the dual norm defined as

$$\mathcal{B}_{\infty,q} \triangleq \{\mathbf{H} \in \mathbb{R}^{nl \times 3} : \|\mathbf{H}_i\|_{\mathcal{S}_q} \leq 1 \quad \forall i = 1, \dots, n\}. \quad (3.19)$$

Algorithm 1 Tomography with STP regularizer

Require: $\lambda, \eta, \tau, \theta \in \mathbb{R}$, $\mathbf{p} \in \mathbb{R}^m$, $l \in \mathbb{N}$

1: Initialize: $\bar{\mathbf{f}}^{(0)} = 0$

2: **for** $t = 1 \dots T$ **do**

3: Solve

$$\mathbf{Y}^{t+1} = \text{prox}_{\eta g^*}(\mathbf{Y}^t + \eta \mathbf{J}_K \bar{\mathbf{f}}^t) = \Pi_{\lambda \mathcal{B}_{\infty,q}}(\mathbf{Y}^t + \eta \mathbf{J}_K \bar{\mathbf{f}}^t)$$

4: Solve

$$\mathbf{f}^{t+1} = \text{prox}_{\tau f}(\mathbf{f}^t - \tau \mathbf{J}_K^* \mathbf{Y}^{t+1})$$

using Algorithm 2 with input $\mathbf{q} = \mathbf{f}^t - \tau \mathbf{J}_K^* \mathbf{Y}^{t+1}$ and parameter τ .

5: Update

$$\bar{\mathbf{f}}^{t+1} = \mathbf{f}^{t+1} + \theta (\mathbf{f}^{t+1} - \mathbf{f}^t)$$

6: **end for**

return volume reconstruction $\mathbf{f} \in \mathbb{R}^N = \text{argmin}_{\mathbf{f}} \|\mathbf{A}\mathbf{f} - \mathbf{p}\|_2^2 + \lambda S T P_p(\mathbf{f})$.

Using (3.16) we can write (3.14) as

$$\begin{aligned} \lambda \|\mathbf{J}\|_{1,p} &= \lambda \max_{\mathbf{H} \in \mathcal{B}_{\infty,q}} \langle \mathbf{H}, \mathbf{J} \rangle_{\mathbb{R}^{nl \times 3}} \\ &= \max_{\mathbf{H} \in \mathcal{B}_{\infty,q}} \langle \lambda \mathbf{H}, \mathbf{J} \rangle_{\mathbb{R}^{nl \times 3}} \\ &= \max_{\mathbf{V}/\lambda \in \mathcal{B}_{\infty,q}} \langle \mathbf{V}, \mathbf{J} \rangle_{\mathbb{R}^{nl \times 3}} \\ &= \max_{\mathbf{V} \in \lambda \mathcal{B}_{\infty,q}} \langle \mathbf{V}, \mathbf{J} \rangle_{\mathbb{R}^{nl \times 3}} \end{aligned} \quad (3.20)$$

where we defined $\mathbf{V} = \lambda \mathbf{H} \implies \mathbf{H} = \mathbf{V}/\lambda$ and $\lambda \mathcal{B}_{\infty,q}$ is the norm ball of radius λ i.e.

$$\lambda \mathcal{B}_{\infty,q} \triangleq \{\mathbf{H} \in \mathbb{R}^{nl \times 3} : \|\mathbf{H}_i\|_{S_q} \leq \lambda \quad \forall i = 1, \dots, N\}. \quad (3.21)$$

Using (3.20) we can write (3.12) as the following saddle point problem [68]

$$\min_{\mathbf{f}} \max_{\mathbf{H} \in \lambda \mathcal{B}_{\infty,q}} \langle \mathbf{H}, \mathbf{J}_K x \rangle_{\mathbb{R}^{nl \times 3}} + \|\mathbf{A}\mathbf{f} - \mathbf{p}\|_2^2, \quad (3.22)$$

which is equivalent to

$$\min_{\mathbf{f}} \max_{\mathbf{H}} \underbrace{\|\mathbf{A}\mathbf{f} - \mathbf{p}\|_2^2}_{f(\mathbf{f})} + \langle \mathbf{H}, \mathbf{J}_K \mathbf{f} \rangle_{\mathbb{R}^{nl \times 3}} - \underbrace{\iota_{\lambda \mathcal{B}_{\infty,q}}(\mathbf{H})}_{g^*(\mathbf{H})} \quad (3.23)$$

where $\iota_{\lambda \mathcal{B}_{\infty,q}}$ is the indicator function of the norm ball of radius λ

$$\iota_{\lambda \mathcal{B}_{\infty,q}}(\mathbf{H}) = \begin{cases} 0 & H \in \lambda \mathcal{B}_{\infty,q} \\ \infty & \text{otherwise} \end{cases} \quad (3.24)$$

and $g^*(\cdot)$ is the convex conjugate of $g(\cdot)$ [28]

$$g^*(\mathbf{H}) = \max_{\mathbf{J} \in \mathbb{R}^{nl \times 3}} \langle \mathbf{H}, \mathbf{J} \rangle - g(\mathbf{J}).$$

Note that solving Eq. (3.23) is equivalent to solving Eq. (3.14), and we will use the efficient primal-dual CP algorithm [26] to solve it. We will need to define the proximal operators [28, 5] of f and g^* :

- The proximal operator for $f(\mathbf{f})$ is

$$\mathbf{prox}_{\mu f}(\mathbf{q}) = \operatorname{argmin}_{\mathbf{f}} \|\mathbf{A}\mathbf{f} - \mathbf{p}\|_2^2 + \frac{1}{2\mu} \|\mathbf{f} - \mathbf{q}\|_2^2. \quad (3.25)$$

which can be solved directly using the SART proximal operator .

- The proximal operator for $g^*(\mathbf{H})$ is

$$\mathbf{prox}_{\eta g^*}(\mathbf{H}) = \operatorname{argmin}_{\mathbf{J}} \iota_{\lambda \mathcal{B}_{\infty,q}}(\mathbf{J}) + \frac{1}{2\eta} \|\mathbf{J} - \mathbf{H}\|_F^2 \quad (3.26)$$

which is the projection on the convex set $\lambda \mathcal{B}_{\infty,q}$, and decomposes over the n components \mathbf{H}_i of \mathbf{H} . The projection of \mathbf{H}_i is defined as

$$\Pi_{\lambda \mathcal{B}_{\infty,q}}(\mathbf{H}_i) = \mathbf{q} \hat{\Sigma} \mathbf{V}^T \quad (3.27)$$

where $\mathbf{H}_i = \mathbf{q} \hat{\Sigma} \mathbf{V}^T$ is its SVD and $\hat{\Sigma} = \operatorname{diag}(\hat{\sigma}_i)$ with

$$\hat{\sigma} = \Pi_{\lambda \ell_q}(\sigma) \quad (3.28)$$

which is the projection of the vector of singular values of \mathbf{H}_i on the q norm ball with radius λ . For example, when $p = 1$, we have $q = \infty$ and the projection function simplifies to simple truncation of the singular values of \mathbf{H}_i

$$\hat{\sigma} = \Pi_{\lambda \ell_\infty}(\sigma) = \min(\sigma, \lambda). \quad (3.29)$$

The steps to solve the reconstruction problem in Eq. (3.12) are outlined in Algorithm 2.

3.3 SART for the Data Term

3.3.1 Data Term Solver

The proximal operator for the data term $\mathbf{prox}_{\tau_f}(\mathbf{q})$ has traditionally been solved using Conjugate Gradient (CG) [69]. In particular, it can be cast as a least squares problem, and solved using CGLS [70]. However, we find that CG does not in general converge to the global optimum for the tomography data term proximal operator, although it

is a convex problem. These problems can be traced back to two factors, that are both related to the size of the linear system in computed tomography problems:

- CG in general is known to have issues with large systems [71, 72]. Then, it requires a good preconditioner for large and sparse systems. For tomography, preconditioning is usually not an option, since it is infeasible to store the system matrix \mathbf{A} , and CG is instead used in a matrix-free fashion. In fact support for matrix-free operation is one of the primary motivators for using CG in this context, but it limits the choice of preconditioner to e.g. Jacobi preconditioning, which is not very effective for tomography matrices.
- As another consequence of needing to operate in matrix free mode, the matrices themselves are laden with numerical noise. Specifically, solving the least squares problem with a system matrix $\mathbf{A}^T \mathbf{A}$ requires the procedural implementation of two operations: $\mathbf{A} \cdot \mathbf{f}$ (projection) and $\mathbf{A}^T \cdot \mathbf{f}^*$ (backprojection), where \mathbf{f} is a volume and \mathbf{f}^* is the set of projection images. Because of slight numerical discrepancies between the implementations of these two procedural operators, the resulting matrices are not generally exact transposes of each other. CG does tend to be more sensitive to this issue than other solvers.

3.3.2 SART

For the SART algorithm, the update equation for each voxel \mathbf{f}_j in the volume \mathbf{f} is:

$$\mathbf{f}_j^{(t+1)} = \mathbf{f}_j^{(t)} + \alpha \frac{\sum_{i \in \mathcal{S}} c_i^{(t)} a_{ij}}{\sum_{i \in \mathcal{S}} a_{ij}}, \quad (3.30)$$

where

$$c_i^{(t)} = \frac{b_i - \hat{b}_i^{(t)}}{\sum_k a_{ik}} \quad (3.31)$$

is the normalized correction factor for ray i that measures the residual between the

measured projection value b_i and the current estimate at iteration t :

$$\hat{b}_i^{(t)} = \sum_k a_{ik} \mathbf{f}_k^{(t)}, \quad (3.32)$$

α is a relaxation parameter usually $0 < \alpha < 2$, \mathcal{S} is a set of projection rays under consideration, and a_{ij} is the element in row i and column j of the system matrix \mathbf{A} and defines the contribution to ray sum i from voxel j . Basically the update operations in the equation can be decomposed into three steps [73]:

1. **Forward projection:** computes the estimated projection $\hat{b}_i^{(t)}$ for each ray i from the current volume $\mathbf{f}^{(t)}$ (Eq.(3.32)). This corresponds to a volume rendering operation.
2. **Correction:** computes $c_i^{(t)}$, the normalized deviation of this estimate from the true projection b_i , where the correction is normalized by the contribution of this ray to all the voxels it goes through (Eq.(3.31)).
3. **Backprojection:** where this correction factor is distributed back to all the voxels that contribute to this ray sum (Eq.(3.30)).

3.3.3 Definition

We now show how to use the SART algorithm to solve the data term proximal operator $\text{prox}_{\lambda f}(\mathbf{q})$. In particular, we want to solve:

$$\text{prox}_{\lambda f}(\mathbf{q}) = \underset{\mathbf{f}}{\operatorname{argmin}} \|\mathbf{Af} - \mathbf{p}\|_2^2 + \frac{1}{2\lambda} \|\mathbf{f} - \mathbf{q}\|_2^2. \quad (3.33)$$

Recall that SART solves a minimum norm problem. By introducing new variables: $\mathbf{y} = \sqrt{2\lambda}(\mathbf{p} - \mathbf{Af})$ and $\mathbf{z} = \mathbf{f} - \mathbf{q}$, and after further manipulations, it can be shown

that solving the optimization problem in Equation (3.33) is equivalent to solving:

$$\min_{\mathbf{y}, \mathbf{z}} \left\| \begin{bmatrix} \mathbf{y} \\ \mathbf{z} \end{bmatrix} \right\|_2^2 \quad \text{subject to: } \begin{bmatrix} \mathbf{I} & \sqrt{2\lambda}\mathbf{A} \end{bmatrix} \begin{bmatrix} \mathbf{y} \\ \mathbf{z} \end{bmatrix} = \sqrt{2\lambda}(\mathbf{p} - \mathbf{A}\mathbf{q}), \quad (3.34)$$

which can be written as:

$$\min_{\tilde{\mathbf{f}}} \left\| \tilde{\mathbf{f}} \right\|_2^2 \quad \text{subject to: } \tilde{\mathbf{A}}\tilde{\mathbf{f}} = \tilde{\mathbf{p}}, \quad (3.35)$$

where $\tilde{\mathbf{f}} \in \mathbb{R}^{m+n}$, $\tilde{\mathbf{A}} \in \mathbb{R}^{m \times m+n}$, and $\tilde{\mathbf{p}} \in \mathbb{R}^m$. This is now an under-determined linear system, and can be solved using SART.

Algorithm 2 summarizes the steps for the modified SART to solve the proximal operator.

Algorithm 2 SART For Solving The Data Term

Require: $\mathbf{A} \in \mathbb{R}^{m \times n}$, $\mathbf{q} \in \mathbb{R}^n$, $\lambda \in \mathbb{R}$, $\alpha \in \mathbb{R}$, $\mathbf{p} \in \mathbb{R}^m$

```

1:  $\mathbf{p} = \sqrt{2\lambda}\mathbf{p}$ ,  $\mathbf{A} = \sqrt{2\lambda}\mathbf{A}$ 
2: Initialize:  $\mathbf{y}^{(0)} = 0$ ,  $\mathbf{f}^{(0)} = \mathbf{q}$ 
3: for  $t = 1 \dots T$  do
4:   for projections  $\mathcal{S} \in \mathcal{S}_1 \dots \mathcal{S}_N$  do
       $\mathbf{y}_j^{(t+1)} = \mathbf{y}_j^{(t)} + \alpha \mathbf{c}_j^{(t)}$  for  $j \in \mathcal{S}$ 
       $\hat{\mathbf{b}}_i^{(t+1)} = \sum_k a_{ik} \mathbf{f}_k^{(t)} + \mathbf{y}_i^{(t)}$ 
       $\mathbf{c}_i^{(t+1)} = \frac{\mathbf{b}_i - \hat{\mathbf{b}}_i^{(t+1)}}{\sum_k a_{ik} + 1}$ 
       $\mathbf{f}_j^{(t+1)} = \mathbf{f}_j^{(t)} + \alpha \frac{\sum_{i \in \mathcal{S}} \mathbf{c}_i^{(t+1)} \mathbf{a}_{ij}}{\sum_{i \in \mathcal{S}} \mathbf{a}_{ij}}$  for  $j = 1 \dots n$ 
    5: end for
  6: end for
  return volume reconstruction  $\mathbf{f} \in \mathbb{R}^n$ 


---



```

3.3.4 Algorithm

Although we introduced new variables \mathbf{y} and \mathbf{z} and increased the dimensionality of the problem from n to $n+m$, we can solve the modified SART efficiently with very

little computational overhead. Instead of solving SART explicitly for the optimal \mathbf{y}^* and \mathbf{z}^* , we can manipulate the algorithm to solve it directly for the optimal \mathbf{f}^* . In particular, the Eq. (3.30) for the augmented system $\tilde{\mathbf{A}}\tilde{\mathbf{f}} = \tilde{\mathbf{p}}$ becomes (by substituting all variables)

$$\begin{aligned}\tilde{\mathbf{f}}_j^{(0)} &= \mathbf{0}, \\ \tilde{\mathbf{f}}_j^{(t+1)} &= \tilde{\mathbf{f}}_j^{(t)} + \alpha \frac{\sum_{i \in \mathcal{S}} \frac{\tilde{b}_i - \sum_k \tilde{a}_{ik} \tilde{\mathbf{f}}_k^{(t)}}{\sum_k \tilde{a}_{ik}} \tilde{a}_{ij}}{\sum_{i \in \mathcal{S}} \tilde{a}_{ij}},\end{aligned}\quad (3.36)$$

which can be expanded in terms of y , z , and A as

$$\begin{aligned}\mathbf{y}_j^{(t+1)} &= \mathbf{y}_j^{(t)} + \frac{\alpha \sum_{i \in \mathcal{S}} \frac{\tilde{b}_i - \sqrt{2\lambda} \sum_k a_{ik} z_k^{(t)} - \mathbf{y}_i^{(t)}}{\sqrt{2\lambda} \sum_k a_{ik} + 1} \delta_{ij}}{1}, \\ \mathbf{z}_j^{(t+1)} &= \mathbf{z}_j^{(t)} + \alpha \frac{\sum_{i \in \mathcal{S}} \frac{\tilde{b}_i - \sqrt{2\lambda} \sum_k a_{ik} z_k^{(t)} - \mathbf{y}_i^{(t)}}{\sqrt{2\lambda} \sum_k a_{ik} + 1} \sqrt{2\lambda} a_{ij}}{\sqrt{2\lambda} \sum_{i \in \mathcal{S}} a_{ij}},\end{aligned}$$

where $\delta_{ij} = 1$ if $i = j$ and 0 otherwise. Using the fact that $\mathbf{z} = \mathbf{f} - \mathbf{q}$ and simplifying we get

$$\begin{aligned}\mathbf{y}_j^{(t+1)} &= \mathbf{y}_j^{(t)} + \alpha \sum_{i \in \mathcal{S}} \frac{\sqrt{2\lambda} b_i - \sqrt{2\lambda} \sum_k a_{ik} \mathbf{f}_k^{(t)} - \mathbf{y}_i^{(t)}}{\sqrt{2\lambda} \sum_k a_{ik} + 1} \delta_{ij}, \\ \mathbf{f}_j^{(t+1)} &= \mathbf{f}_j^{(t)} + \alpha \frac{\sum_{i \in \mathcal{S}} \frac{\sqrt{2\lambda} b_i - \sqrt{2\lambda} \sum_k a_{ik} \mathbf{f}_k^{(t)} - \mathbf{y}_i^{(t)}}{\sqrt{2\lambda} \sum_k a_{ik} + 1} \sqrt{2\lambda} a_{ij}}{\sqrt{2\lambda} \sum_{i \in \mathcal{S}} a_{ij}}.\end{aligned}$$

Algorithm 2 summarizes the steps for the modified version of the SART algorithm used to solve the proximal operator. We note the following:

1. The initialization is different since we need to initialize \mathbf{y} and \mathbf{f} .
2. The update for \mathbf{y} is very fast because only one index \mathbf{y}_j is updated for every projection pixel $i = j$.
3. The update for \mathbf{f} is very similar to standard version of the SART with the

exception of the term $\mathbf{y}_i^{(t)}$ in the formula for $\mathbf{c}_i^{(t)}$.

3.4 Implementation Details

Chambolle-Pock Primal-Dual algorithm. We calculate the norm of matrix [74] \mathbf{K} to set the optimal values for μ and η in our PSART framework for faster convergence. The parameter θ is set to 1 in all experiments.

STP regularizer. To compute the STP, we used a truncated 3D Gaussian kernel with support of $3 \times 3 \times 3$ voxels (i.e. $l = 27$) and standard deviation $\sigma = 0.5$ voxels. The linear mapping \mathbf{J}_K that computes the patch-based Jacobian is not stored explicitly, and is computed on the fly using discrete forward differences and scaling. In particular, we can decompose $\mathbf{J}_K \in \mathbb{R}^{3nl \times n}$ into two operators

$$\mathbf{J}_K = \mathbf{CD}$$

where $\mathbf{D} \in \mathbb{R}^{3n \times n}$ is a discrete forward-difference matrix that computes local gradients for the voxels and $\mathbf{C} \in \mathbb{R}^{3nl \times 3n}$ extracts the patch-based Jacobian for each voxel over its neighborhood and scales them appropriately using the kernel $\mathbf{K}(\cdot)$. The adjoint $\mathbf{J}_K^* = \mathbf{D}^* \mathbf{C}^*$ is also computed on the fly.

However, the output of the application of the linear map \mathbf{J}_K to the volume \mathbf{f} , i.e. the patch-based Jacobian $\mathbf{J}_K(\mathbf{f})$, needs to be stored in the memory. In particular, for every voxel \mathbf{f}_i , we need to store a matrix $\mathbf{J}_K(\mathbf{f}_i) \in \mathbb{R}^{l \times 3}$ that has the weighted local gradients at its $l - 1$ neighbors and itself (e.g. $l = 27$ for a neighborhood of 3 pixels in each dimension). This means that we need a storage of 82 times the size of the volume to be reconstructed (81 for $\mathbf{J}_K(\mathbf{f})$ plus 1 for the volume itself). Moreover, the temporary and slack variables in Algorithm 2 have also to be stored. In the experiments, the largest volume reconstructed has $690 \times 668 \times 776$ voxels, and the memory required for storing the volume and the patch-based Jacobian is 109 GB

Table 3.1: Parameters for the X-ray CT data.

	zone plate	Artificial rose	Plumeria	Toothbrush
SID (mm)	1800	536.9627	697.0378	243.1662
SDD (mm)	5000	983	983	983
Detector pixel	1024×1024	1916×1536	1916×1536	1916×1536
Detector pixel size (mm)	1	0.127	0.127	0.127
Input pixel	160	168×135	120×96	1916×1536
Input pixel size (mm)	6.4	1.4484	2.027	0.127
Image downsampled factor	$6.4 \times$	$11.4 \times$	$16 \times$	$1 \times$
X-ray penetration (kV)	NA	31	25	32
X-ray intensity (μA)	NA	725	860	421
Voxel size (mm)	0.5	0.2775	0.3605	0.0314
Number of projections	180	120	180	180

using single-precision (4-byte) floating point numbers.

3.5 Experiments

The experiments were run on a machine with two Intel Xeon E5-2697 processors (56 cores overall) and 128 GB of RAM. We present two kinds of experiments:

1. focusing on sparse view reconstruction using the 3D Shepp-Logan phantom and the scans of the rose in Fig. 1.1(c).
2. focusing on super resolution using a simulated 3D Fresnel zone plate, scans of the artificial rose, the plumeria flower, and the toothbrush((a),(b), and (d) in Fig. 1.1, respectively).

3.5.1 Sparse-View Reconstruction

We first validate our choice of SART as the solver for the data term in Eq. (1.1). We run experiments comparing SART head-to-head with Conjugate Gradient (CG) in a sparse-view setting, using the TV regularizer in both cases. In particular, we show the reconstruction quality, measured in PSNR and SSIM, as a function of the number of projections available. We use the implementation provided in RTK and

Table 3.2: Parameters used with each method. Time elapsed with each method denotes the computing time for each main loop iteration.

		Zone plate	Artificial rose	Plumeria	Toothbrush
FDK	Time elapsed (s)	22.2	8.4	-	-
SART	Iter. No.	15	20	-	-
	Relax. param. α	0.3	0.1	-	-
	Time elapsed (s)	150	44	-	-
PCG-TV	Main loop iter.	40	30	25	20
	Nested CG iter.	6	4	4	6
	Nested TV iter.	1	1	1	1
	Prior parameters α	0.8	55	35	0.1
	ADMM param. β	500	8	8	1
	Time elapsed (s)	242	36	33	1158
PSART-TV	Main loop iter.	20	25	15	15
	Nested SART iter.	1	1	1	1
	Nested TV iter.	1	1	1	1
	Prior parameters λ	0.003	0.08	0.03	0.1
	CP parameters μ , ($\mu = \tau$)	0.15	0.1	0.1	0.1
	Time elapsed (s)	98	35	33	1182
PSART-SAD	Main loop iter.	20	25	15	15
	Nested SART iter.	1	1	1	1
	Nested SAD iter.	1	1	1	1
	Prior parameters λ	0.003	0.08	0.03	0.1
	CP parameters μ , ($\mu = \tau$)	0.15	0.1	0.1	0.1
	Time elapsed (s)	114	42	38	1228
PSART-STP	Main loop iter.	25	16	28	25
	Nested SART iter.	1	1	1	1
	Nested STP iter.	1	1	1	1
	Prior parameters λ	0.03	0.5	0.03	5.5
	CP parameters μ ($\mu = \tau$)	0.3	0.3	0.3	0.3
	Time elapsed (s)	326	142	95	2347

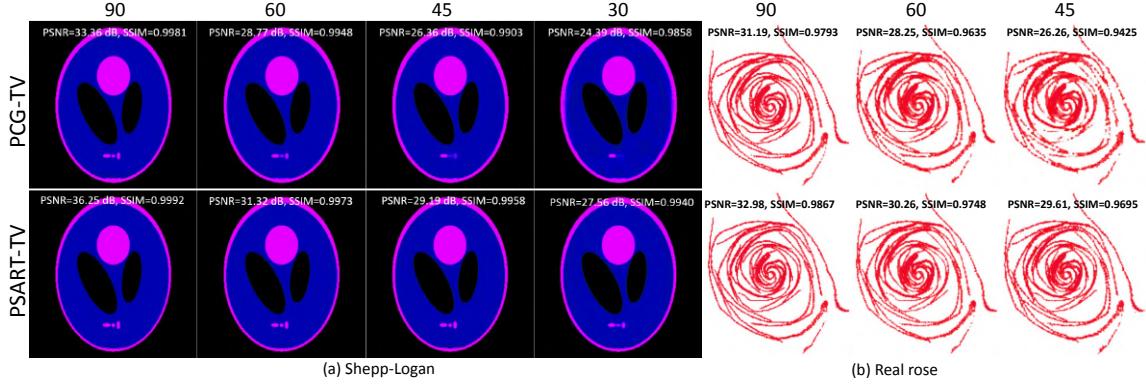


Figure 3.1: A slice sample with different number of projections from a 3D Shepp-Logan (a) and scanned rose (b). The PSNR and SSIM values are shown at the top of each image. For each data, we compare PCG-TV (top) with our proposed PSART-TV method (bottom). For Shepp-Logan data, 90, 60, 45, and 30 projections as input. For the real scanned rose, 90, 60, 45 were used projections as input.

compare to our framework using the SART proximal operator solver. The size of the 3D Shepp-Logan volume is $300 \times 300 \times 300$ with voxel size of $1 \times 1 \times 1 \text{ mm}$, while the volume size of the rose is $436 \times 300 \times 365$ with voxel size of $0.3 \times 0.3 \times 0.3 \text{ mm}$. More details for the parameters used in the experiments are shown in Table 3.1 and Table 3.2.

As Fig. 3.1 shows, SART as a solver for the data term provides better quality than CG, which is expected given the known limitations of CG, whereby it is prone to overfitting the projection noise in the data, which becomes even more pronounced when the number of projections is smaller.

Additional comparison between PCG and PSART results in sparse view tomographic reconstruction scenario is shown in Figure 3.2. Interestingly, we found that the proximal operator for the data term leads to a strongly convex optimization problem and therefore should have a unique optimal solution. However, CG in practice struggles to find it. This is quite well known in the tomography community (which is why SART and other methods remain popular for this application). For example the Figure 3.2 shows results for comparing different solvers for the proximal operator. We can see that in all settings all methods seem to converge against a similar

result (albeit at different speeds), except for CG (grey), which stalls at a much lower value. Note that this behavior is observed across different implementations of CG and CGLS. For example in our submission we use the CG implementation from RTK [75] (as well as the projection/backprojection implementation from the same source), in the figure we show results from the ASTRA toolkit [76], and we also have results from our own CG implementation. So the issue for CG is quite reasonable.

3.5.2 Super Resolution Experiments

Now we run experiments to compare the new regularizer in a super-resolution setting. We chose the following algorithms for our comparison:

- PSART-STP: this is our complete framework using the Structure Tensor prior.
- PSART-SAD: this is our framework with the previously used SAD (Sum of Absolute Differences) prior [77]. It was shown before [21] that SAD performs better than TV, and so we chose it as the best alternative prior to compare to STP.

We compare results from our framework to state-of-the-art algorithms and comparable implementations in RTK, namely:

- Cone Beam Filtered Back Projection (FDK) [34], as the FDK algorithm is still the most commonly used method in practical CT scanners [78].
- Plain SART with no priors (SART).
- ADMM with ATV prior (PCG-TV) using Conjugate Gradient (CG) [6].

The initial volume for all methods is set to 0. For choosing the hyper parameters in all the algorithms, we experiment with a range of values and pick the ones with the best performance.

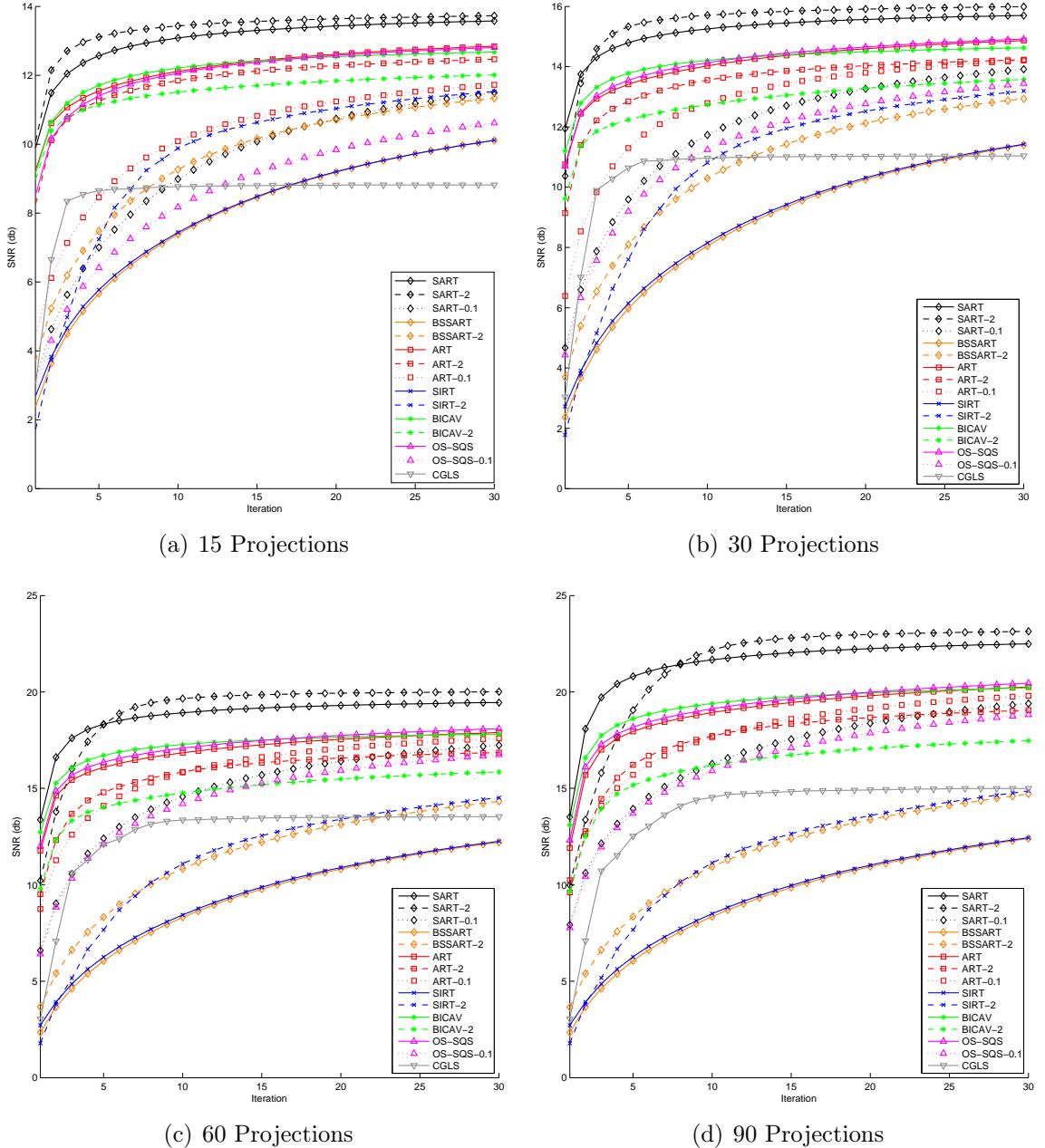


Figure 3.2: Comparison of iterative solvers. Plots show SNR per iteration. Relaxation parameter $\alpha = 1, 2.0$, and 0.1 are represented with solid, dashed, and dotted lines.

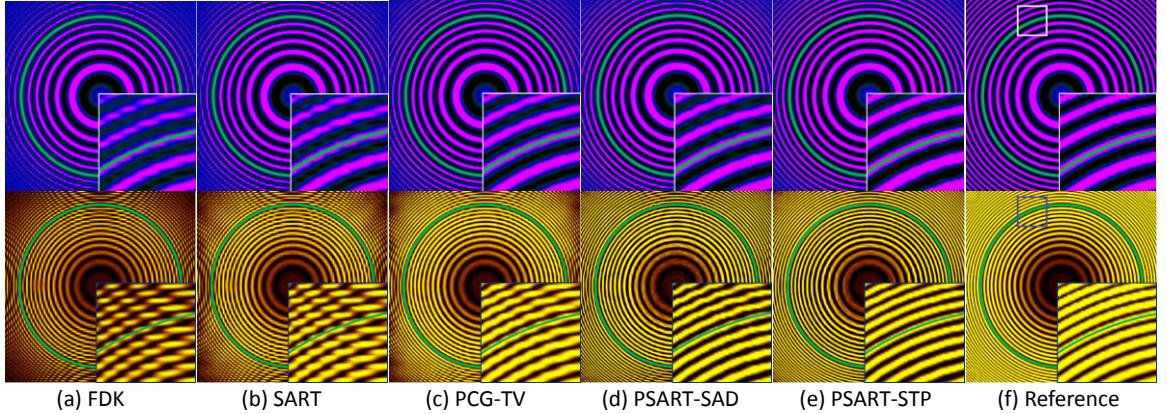


Figure 3.3: 2D slice from the reconstructed 3D Fresnel zone plate (top) and its Sobel filtered visualization (bottom). The green ring in each image represents the smallest feature we can extract according to the Nyquist limit. PSNR and SSIM of slice images (top) from (a) to (e): FDK (17.5978, 0.9354), SART (19.5440, 0.9582), PCG-TV (22.0659, 0.9756), PSART-SAD (**22.6293, 0.9781**), PSART-STP (**24.8331, 0.9864**), Reference volume. The display window is [0, 0.8]. For Sobel filtered images, smoother features in the superresolution frequencies for the PSART results indicate a better suitability for post-processing tasks such as segmentation.

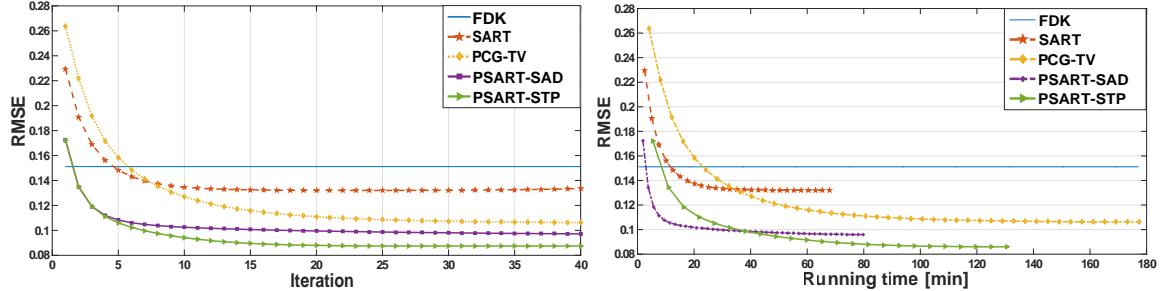


Figure 3.4: RMSE of the reconstructed volume as a function of iteration (left) and running time (right) for the various methods.

First, we use a synthetic volume dataset to demonstrate the super-resolution capabilities of the PSART framework. Specifically, we show a cone-beam tomographic reconstruction of a 3D version of the Fresnel zone “plate” (a 2D cross-section is shown in Fig. 3.3(f)). After adding Gaussian noise with standard deviation $\sigma = 2$, the projection images are downsampled with the scale factor as 6.4 using bicubic interpolation, which are the input for our experiments. We run the SART algorithm with 180 projections with the original size until convergence (15 iterations), and considered the resulting reconstruction the reference volume for numerical comparisons.

Fig. 3.3 (top) shows a visual comparison of the different reconstruction methods, together with the obtained PSNR and SSIM values. As can be seen, the PSART framework, with SART as a solver for the data fidelity term outperforms the other state-of-the-art methods, even when used in combination with the SAD regularizer. The use of the STP provides an additional quality boost. In particular, we note the improved reconstruction quality for frequencies above the Nyquist limit for the 2D pixel sampling rate (green circle).

These results are further confirmed in Fig. 3.3 (bottom). Since tomographic reconstruction is often just the first step in an image analysis pipeline, we tested how robust and reliable the super-resolution information is for further processing such as image segmentation. As a stand-in for more sophisticated segmentation methods, we applied a 3D variant of the Sobel filter [79] to extract the boundaries between the rings. Smoother results from the Sobel filter indicate that it will be easier to trace thin structures through the volume in a segmentation process. We can again see that PSART generates significant super-resolution information, with PSART-STP performing best.

Fig. 3.4 shows the evolution of the RMSE plotted against the iteration and running time during the zone plate volume reconstruction for each method. The PSART methods (PSART-SAD and PSART-STP) converge faster than PCG-TV in terms of running time, and PSART-STP converges slower than PSART-SAD but finds a solution with lower RMSE.

We ran another round of experiments on real datasets that were scanned using a Nikon X ray CT, namely artificial flowers, a plumeria flower, and a toothbrush. These objects have the structural features we are interested in modeling i.e. thin sheets and thin tubes.

The reconstructed volume size for **artificial rose** is $415 \times 314 \times 393$. 120 original-size projection images are used as input for PSART-STP and the best reconstructed

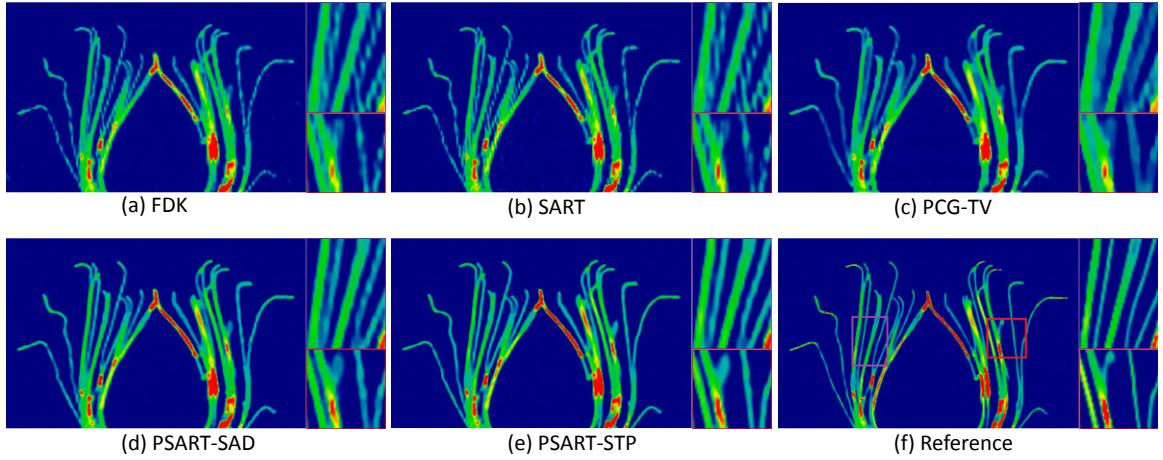


Figure 3.5: a-f: Representative slice visualization in the sagittal plane for the volume and its closeup view for the artificial flower data reconstructed by FDK, SART, PCG-TV, PSART-SAD, PSART-STP, and the reference volume, respectively.

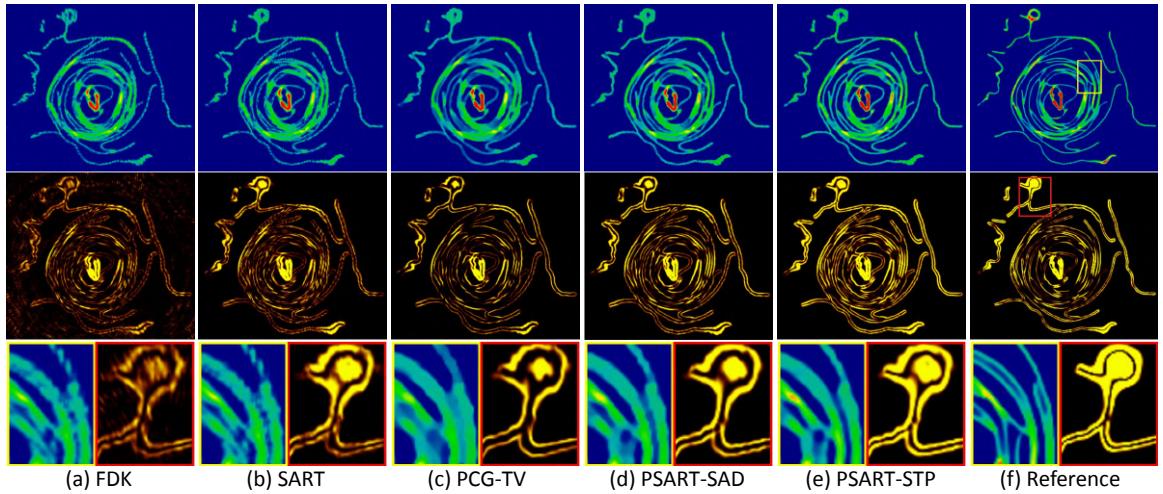


Figure 3.6: Representative slice visualization in the axial plane for (f): the reference volume and (a)-(e): the volumes reconstructed by FDK, SART, PCG-TV, PSART-SAD, and PSART-STP, respectively. From top to bottom: volume visualization, its edge detection, and the closeup views.

result is used as the reference volume for our comparison. Fig. 3.5 shows reconstruction results from different methods in the sagittal plane, and the edge detection results from applying Sobel filter are provided in Fig. 3.9. We can see clearly that our PSART-SAD and PSART-STP achieve better performance than existing methods. Fig. 3.6 shows the results in the axial plane. The reconstructed volume size for the **plumeria** is $406 \times 259 \times 336$. Fig. 3.7 (a) shows the comparison to the state-of-the-art PCG-TV method. For better visualization and comparisons, we generated a reference volume by running the PSART-STP method with 360 original images as input until convergence. The reconstructed volume size for the **toothbrush** is $690 \times 668 \times 776$. Fig. 3.7 (b) shows the comparison between PCG-TV and the proposed PSART-STP. Again, compared to PCG-TV, our method achieves shaper results.

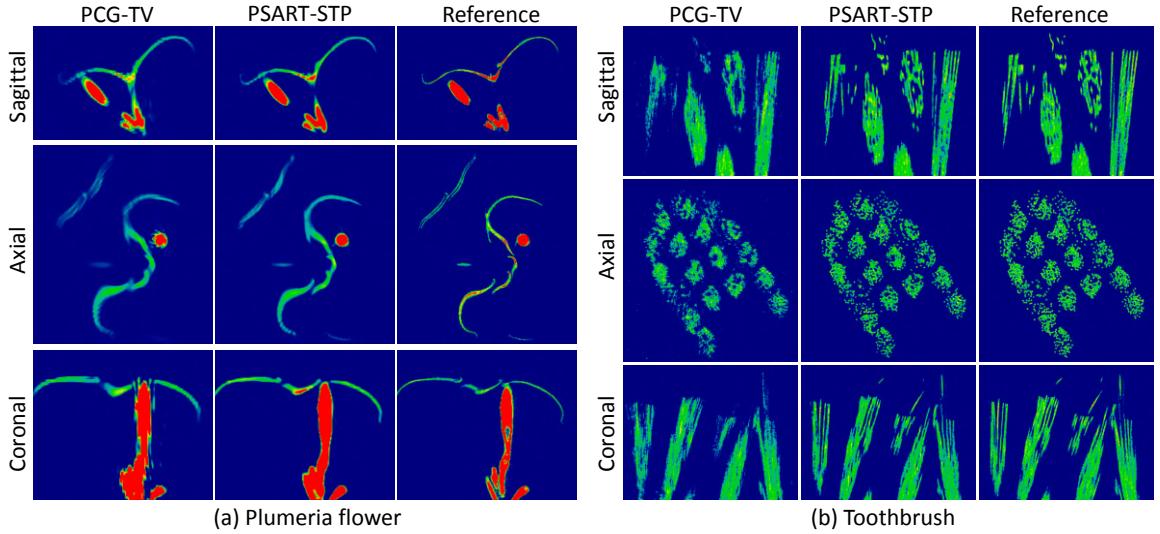


Figure 3.7: Reconstruction results for the real flower (a) and toothbrush (b) in the sagittal, axial, and coronal planes, respectively.

We compared PSART-TV with PCG-TV on three different datasets and verified that SART outperforms CG to solve the data term in the scenario of super resolution. As shown in Figure 3.8, PSART-TV achieves visually better results on an artificial rose, a zone plate pattern, and a real rose.

Figure 3.9 shows the edge detection results from applying Sobel filter in the sagittal

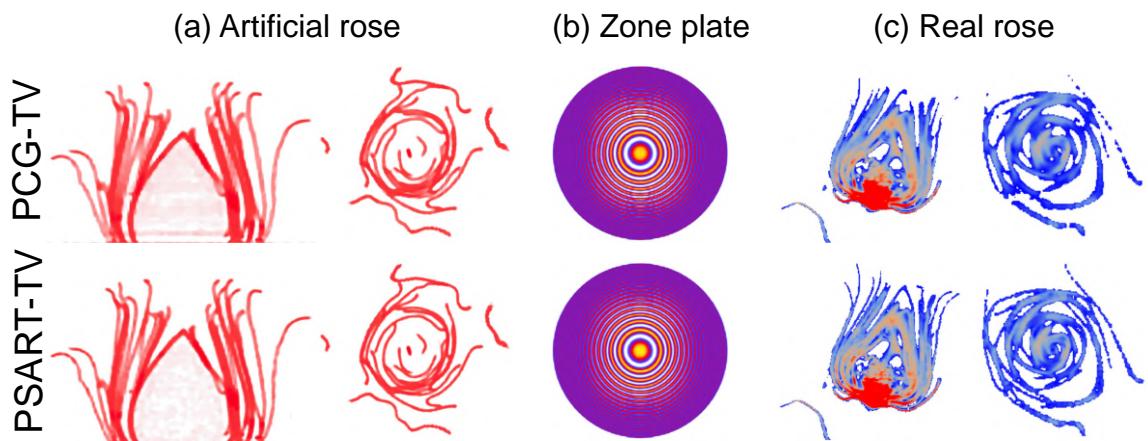


Figure 3.8: Reconstructed results comparison between PSART-TV and PCG-TV.

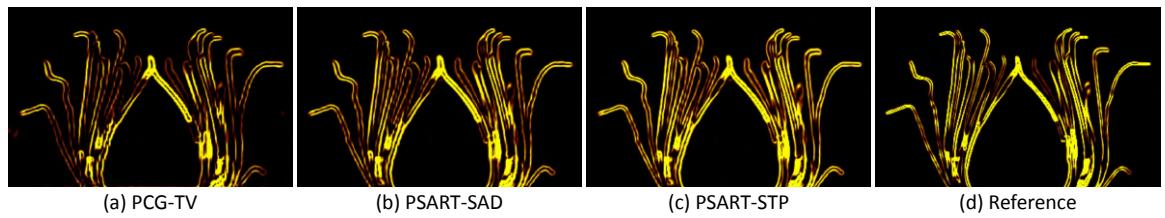


Figure 3.9: a-d: Representative slice visualization in the sagittal plane for edge detection on a volume reconstructed by PCG-TV, PSART-SAD, PSART-STP, and the reference volume, respectively.

Table 3.3: PSNR and SSIM results from different reconstruction methods.

Method	Metric	Zone Plate	Artificial rose	Plumeria	Toothbrush
PCG-TV	PSNR	22.07	26.83	28.73	27.04
	SSIM	0.976	0.958	0.963	0.934
PSART-TV	PSNR	22.18	27.00	28.87	27.09
	SSIM	0.976	0.958	0.964	0.934
PSART-SAD	PSNR	22.63	27.53	28.98	27.62
	SSIM	0.978	0.961	0.969	0.938
PSART-STP	PSNR	24.83	29.93	30.54	31.30
	SSIM	0.986	0.973	0.983	0.973

plane for **artificial rose**. The size of the volume is $415 \times 314 \times 393$. 120 original-size projection images are used as input for PSART-STP and the best reconstructed result is used as the reference volume for the comparison.

In Table 3.3, we show the detailed metrics measured on the reconstructed results using different methods, namely PCG-TV, PSART-TV, PSART-SAD, and PSART-STP. The visualization of the reconstructed results are included in our paper. Specifically, in our measurements, the volumes reconstructed from high resolution projections and a full number of projection images for each data set are treated as ground truth. The metrics PSNR and SSIM are calculated as quantitative performance measure. We can observe that our proposed PSART-STP achieves the best performance compared to the other three methods.

In summary, for both simulated and real scanned data, our PSART reconstructions (PSART-SAD and PSART-STP) consistently give better results than the equivalent PCG-TV. PSART-SAD works better than PCG-TV, confirming earlier results about the SAD regularizer [21, 77]. Our PSART-STP method produces the best results in terms of both quantitative (PSNR and SSIM) and qualitative comparisons (visualization of volume and edge detection filter), allowing for super-resolved reconstruction of thin structures shown in Fig. 1.1.

Table 3.4: Segmentations Results, the threshold parameter is adjusted so that the results from PCG-TV and PSART-SAD are close to the Nyquist limit.

Threshold	Nyquist Limit	PCG -TV	PSART -SAD	PSART -STP	Ground Truth
0.55	6	7	7	11	17
0.6	6	5	5	10	17
0.65	6	4	4	8	17

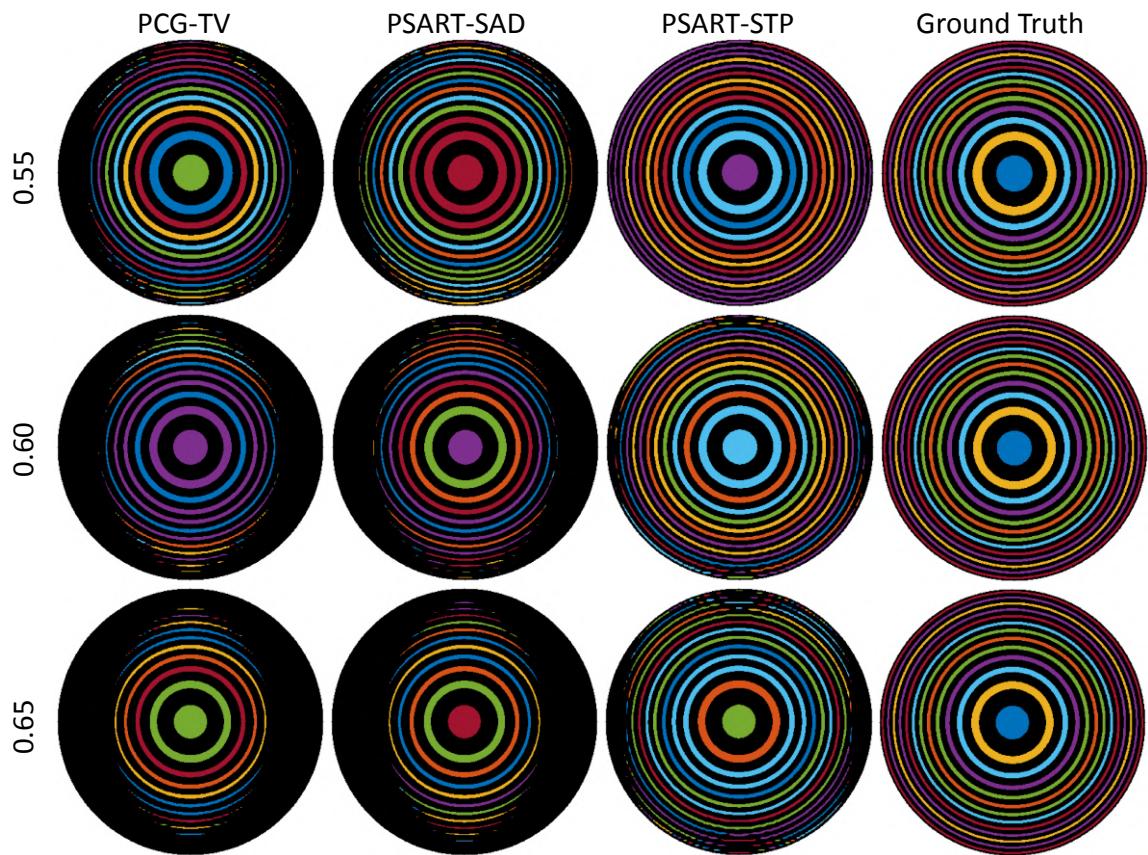


Figure 3.10: Segmentation results from different methods. From top to bottom: the threshold parameters are 0.55, 0.60, and 0.65, respectively, making the reconstructed results of PCG-TV and PSART-SAD closed to the Nyquist limit. From left to right: different reconstructed methods: PCG-TV, PSART-SAD, and PSART-STP, respectively.

3.5.3 Segmentation Results

In this part, we show an additional quantitative evaluation on the zone plate data set. We compare the proposed PSART-STP with PSART-SAD and PCG-TV. We use standard image processing algorithms to segment the data. We simply threshold the result and run a connected component algorithm. We check how many rings are correctly reconstructed. A ring is correctly reconstructed if it has no gaps, is not broken into multiple pieces, and if it does not merge with adjacent rings. We report the number of correctly extracted rings in Table 3.4 and visualize the results in Fig. 3.10 (The five outmost rings are cut away for better visualization). We can make the following observations:

1. Our method is the best for all threshold values.
2. For higher threshold values, the performance of all methods degrades.
3. We can extract rings over the Nyquist limit.

3.6 Conclusions

We have presented a flexible proximal framework for robust 3D cone beam reconstruction of super-resolved thin features. Our two main contributions are (a) introduction of the 3D structure tensor as a regularizer for the tomographic reconstruction problem, and (b) the use of SART for the data-fidelity subproblem in the proximal framework. We have experimentally demonstrated that the 3D structure tensor prior is best suited for reconstructing specific structural features such as thin sheets and filaments, and that using SART provides better reconstructions than other solvers, especially in the case of under-determined tomographic reconstruction from a small number of projections.

We have experimentally compared our framework with the popular RTK open-source software toolkit, both on real and simulated datasets, using different state-of-

the-art priors. We showed the robustness of our algorithms in terms of reconstruction quality.

In the future, we plan to extend our framework by adding a GPU version providing a higher level of parallelism.

Chapter 4

Space-Time Tomography

4.1 Introduction

X-ray computed tomography (CT) is a valuable tool for analyzing objects with interesting internal structure or complex geometries that are not accessible with optical means. Unfortunately, tomographic reconstruction of complex shapes requires a multitude (often hundreds or thousands) of projections from different viewpoints. Such a large number of projections can only be acquired in a time-sequential fashion. This significantly limits the ability to use x-ray tomography for either objects that undergo uncontrolled shape change at the time scale of a scan, or else for analyzing dynamic phenomena, where the motion itself is under investigation.

In this chapter, we present a non-parametric space-time tomographic method for tackling such dynamic settings. Through a combination of a new CT image acquisition strategy, a space-time tomographic image formation model, and an alternating, multi-scale solver, we achieve a general approach that can be used to analyze a wide range of dynamic phenomena. We demonstrate our method with extensive experiments on both real and simulated data.

4.2 Low Discrepancy View Sampling

Computed tomography requires obtaining x-ray projection images of the target under different observation angles. In cone-beam tomography, the most common sampling strategy for obtaining these projections is that the target undergoes a full circle of

rotation relative to the x-ray source and sensor, and images are taken in regular intervals. That is, the angles θ_k of the k^{th} projection are given as

$$\theta_k = \theta_0 + k \cdot 2\pi/N_p, \quad k = 0 \dots N - 1, \quad (4.1)$$

where N_p is the total number of projections, and θ_0 is some starting angle. In the following, we will call such a scan sequence a *linear scan round*, or just *round* for short.

While this simple strategy works well for static targets, we seek a more uniform distribution of angles vs. time in the case of dynamic targets. Several acquisition strategies were proposed in the literature with this goal in mind. The closest to our approach is the interlaced view sampling approach by Mohan et al. [80], who proposed to divide the total number of projections N_p into N_{sf} groups consisting of N_θ projections each (Figure 4.1, right). Each group of projections is used to reconstruct a different volume, corresponding to a time frame in an animation of the deforming object.

Within each frame, the projection angles are distributed according to a low discrepancy sequence, which considerably improves the sampling of angle vs. time compared to the simple linear strategy. However, this approach still has several downsides that we would like to address in this thesis:

- In Mohan et al.'s approach [80] the tradeoff between N_{sf} and N_θ is a parameter that needs to be set according to the speed of the deformation (many frames with a small number of projections for fast motions vs. few frames with a large number of projections for slow motion). This might mean having to repeat scans until the best tradeoff is found. In comparison, we seek a strategy that is independent of the speed of motion, and only depends on the capabilities of the hardware.

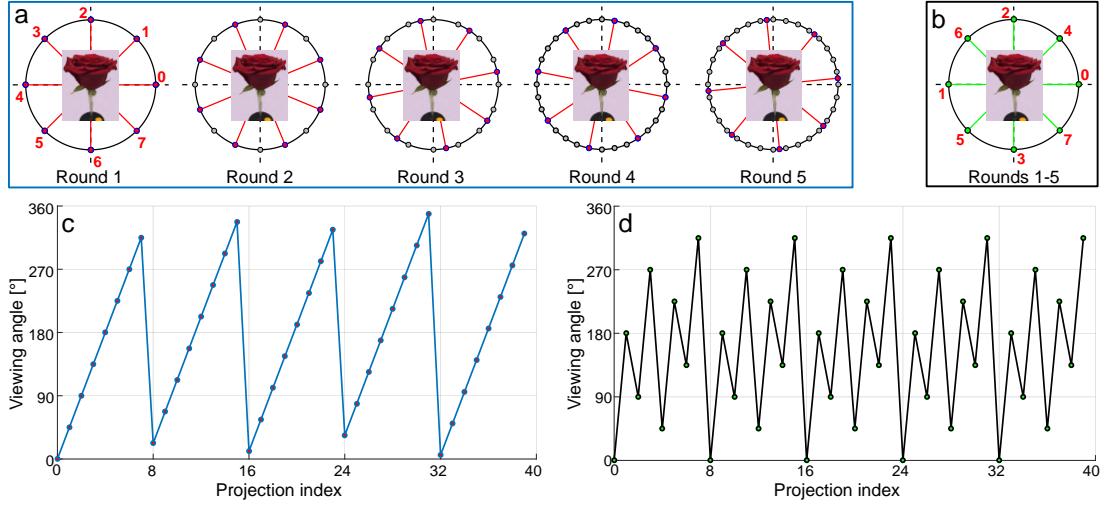


Figure 4.1: Comparison between our acquisition strategy (a and c) and interlaced view sampling method [80] (b and d). The parameters taken for this comparison are: $N_s = N_\theta = 8$ and $N_{sf} = 4$. The acquired viewing angles for each round are represented by red circles (a) or green circles (b). The grey circles correspond to the previously acquired angles. The plots (c and d) represent the viewing angles vs. projection index for the first five rounds.

- In interlaced view sampling, the sequence of angles repeats from frame to frame, so that static or slow moving parts of the geometry do not benefit from additional information as the number of frames increases. Since the number of projections per frame can be small, reconstruction quality will suffer as a result. Instead, we seek an approach where all of the N_p projection angles are unique in order to extract the maximal amount of information from each additional projection.
- The interlaced view sampling strategy cannot be implemented on all CT hardware without changes to control software. For example, the scanner used in our experiments only supports linear scan rounds as discussed above, with a minimum number of $N_{\min} = 10$ projections per round. We aim for a sampling strategy that takes such restrictions into account by decomposing naturally into linear scan rounds.

For our space-time tomography method, we propose to split the total sequence of

N_p projections into N_r rounds, where each round consists of N_{\min} projections that are spaced $\Delta\theta = 2\pi/N_{\min}$ apart. Note that in our approach, the rounds should be as short as possible on a given CT hardware, and the number of rounds should not be confused with the number of frames in the space-time reconstruction. We choose the starting angle $\theta_{0,i}$ for each round i according to a low discrepancy distribution over the range $[0, \Delta\theta]$ to uniformly cover all directions over time (see Figure 4.1, left). Specifically we choose the base-2 Van der Corput sequence [81], which for a given index $i \geq 0$ is defined as

$$h_2(i) = \sum_{j=0}^{\infty} \frac{a_j}{2^{j+1}} \quad \text{with} \quad i = \sum_{j=0}^{\infty} a_j \cdot 2^j, \quad (4.2)$$

where (a_j) is the binary representation of i . $h_2(i)$ is always in the interval $[0, 1]$. The starting angle for the i^{th} round is then given as $\theta_{0,i} = h_2(i) \cdot 2\pi/N_{\min}$, and the angle of the k^{th} projection in the i^{th} round is given as

$$\theta_{k,i} = \theta_{0,i} + k \cdot 2\pi/N_{\min}. \quad (4.3)$$

We note that this strategy simplifies to just the Van der Corput sequence over the full circle if the CT scanner supports “rounds” with just a single image. However, even if the scanner imposes a minimum length for rounds, with our scheme it can still be used to image deformable phenomena, so long as the motion is small over N_{\min} successive frames. All results in this thesis are obtained for $N_{\min} = 10$. We also note that the sampling strategy is completely decoupled from the magnitude of the deformation, and that in particular the number of reconstructed frames in the space-time reconstruction is independent of the number of rounds and can be chosen post-capture. Moreover, since all projection angles are unique, each additional image provides extra information for a high-quality reconstruction, even in the case of stationary or partially stationary geometry.

4.3 Image Formation Model

Our image formation model considers a 3D object undergoing continuous deformation over time. We represent this object as a continuous 4D density function $h(x, y, z, t)$. This density function can be discretized into a sequence of N_k frames, where each frame is represented by a voxel grid with N_v voxels. We choose $N_k \ll N_p$ and assume that each frame represents an approximately static geometry that can be reconstructed at least at low spatial resolution and low quality by a short sequence of N_θ successively captured projections. In our implementation, we choose $N_\theta = N_p/N_k$, however one could also choose larger values of N_θ , so that projections contribute to multiple frames.

After regrouping all voxels in each frame into a column vector, the 4D density volume is described by a vector in $\mathbb{R}^{N_v N_k}$: $\mathbf{f} = \{\mathbf{f}_0, \mathbf{f}_1, \dots, \mathbf{f}_{N_k-1}\}$, where $\mathbf{f}_t \in \mathbb{R}^{N_v}$ represents the 3D volume at the t^{th} time step. With these assumptions, the basic tomography problem on the sequence of frames can be expressed as

$$\begin{pmatrix} \mathbf{A}_1 & & & \\ & \ddots & & \\ & & \mathbf{A}_t & \\ & & & \ddots \\ & & & & \mathbf{A}_{N_k} \end{pmatrix} \cdot \underbrace{\begin{pmatrix} \mathbf{f}_1 \\ \vdots \\ \mathbf{f}_t \\ \vdots \\ \mathbf{f}_{N_k} \end{pmatrix}}_{\mathbf{f}} = \underbrace{\begin{pmatrix} \mathbf{p}_1 \\ \vdots \\ \mathbf{p}_t \\ \vdots \\ \mathbf{p}_{N_k} \end{pmatrix}}_{\mathbf{p}}, \quad (4.4)$$

where $\mathbf{A}_t \in \mathbb{R}^{N_\theta M \times N_v}$ is the matrix that models the Radon transform operator for the N_θ projections contributing to the t^{th} frame. M is the number of pixels for a projection image. $\mathbf{p}_t \in \mathbb{R}^{N_\theta M}$ is the vector of the measured data (sinogram).

Joint optimization framework. In general, this computed tomography reconstruction is an ill-posed inverse problem, especially since only a few projections are used

for the reconstruction of each frame (for the results in this thesis, N_θ ranges from 10 to 60). This problem can be alleviated by aggregating information across frames, using the regularized optimization framework shown in Equation 5.6. This framework jointly reconstructs the 4D volume \mathbf{f} as well as the deformation fields $\mathbf{u} = \{\mathbf{u}_0, \mathbf{u}_1, \dots, \mathbf{u}_{N_k-2}\}$ between successive time frames. Recently, this joint motion estimation and image reconstruction approach has successfully been used in other applications, like in dynamic image reconstruction [82] and fluid flow estimation [14, 83].

$$\begin{aligned}
(\mathbf{f}^*, \mathbf{u}^*) &= \underset{\mathbf{f}, \mathbf{u}}{\operatorname{argmin}} \sum_{t=1}^{N_k} \|\mathbf{A}_t \mathbf{f}_t - \mathbf{p}_t\|_2^2 \\
&+ \kappa_3 \sum_{t=1}^{N_k-1} \|\nabla_T \mathbf{f}_t + \nabla_S \mathbf{f}_t \cdot \mathbf{u}_t\|_1 \\
&+ \kappa_1 \sum_{t=1}^{N_k} \|\nabla_S \mathbf{f}_t\|_{H_\epsilon} + \kappa_2 \sum_{t=1}^{N_k} \|\nabla_T \mathbf{f}_t\|_2^2 \\
&+ \kappa_4 \sum_{t=1}^{N_k-1} \sum_{i=x,y,z} \|\nabla_S \mathbf{u}_{t,i}\|_{H_\tau}
\end{aligned} \tag{4.5}$$

Here, the operators ∇_S and ∇_T represent respectively the spatial and temporal discrete gradient, \mathbf{u}_t represents the deformation vector field at time t .

The first term in Equation 5.6 corresponds to the least-square data fitting term, derived from Equation 5.1. The second term is an L_1 volume correlation term, similar to a 3D version of the Horn-Schunck style brightness constancy term [84] in optical flow. As discussed in Section 4.4 below, we use a multi-scale version of optical flow [85] which does not have a closed form representation, but is capable of dealing with larger deformations than basic Horn-Schunck.

The remaining terms are regularizers for the volume and the deformation field, respectively. On the third line of Equation 5.6 are two terms for regularizing the spatial and the time dimensions of the reconstructed volumes. For the spatial regularization we use the Huber penalty [86] on the spatial gradient over the reconstructed volumes,

which is parameterized by a positive parameter ϵ . The idea of this term is similar to a total variation (TV) prior, except that the Huber penalty prevents the usual staircase artifacts introduced by a TV prior by not penalizing small gradients. Finally, on the last line we introduce a spatial regularization of the deformation field, also utilizing the Huber penalty.

4.4 Solver

Solving the joint optimization problem in Equation 5.6 is challenging, especially because of the optical flow prior. Indeed, this term is neither linear nor convex. Moreover, it involves both variables \mathbf{f} and \mathbf{u} . We address these issues by solving the optimization problem in an alternating way as shown in Algorithm 6.

We first initialize the 4D volume (\mathbf{f}) by applying a variant of the SART algorithm [7] independently for each frame, where the individual projections are processed in a random order to improve convergence [15]. Then, our algorithm alternates between estimating the deformation field (\mathbf{u}) for each pair of successive frames, and refining the reconstruction of \mathbf{f} .

In addition, our alternating joint optimization framework, given in Algorithm 6, follows a multi-scale coarse-to-fine scheme [85]. The deformation field is estimated first for the coarsest scale, then the estimation is propagated step-by-step to the finest levels. For each scale (s), the deformation field is initialized by up-sampling the previous result obtained with a coarser scale. This multi-scale strategy aims to take into account the large deformations in the optical flow prior. In Algorithm 6, the finest and the coarsest scales are respectively obtained for $s = 1$ and $s = N_{scales}$. The operators \uparrow and \downarrow respectively perform cubic up- and down-sampling by a factor of ρ , and the down-sampling operator also includes a Gaussian smoothing.

Detailed explanations of the deformation field estimation and the volume reconstruction and the associated operators $F_{\mathbf{h}}$, $G_{\mathbf{h}}$, $F_{\mathbf{u}}$, and $G_{\mathbf{u}}$ are provided in the fol-

Algorithm 3 Space-Time tomography

```

1: procedure ST-TOMOGRAPHY( $F_{\mathbf{h}}, G_{\mathbf{h}}, F_{\mathbf{u}}, G_{\mathbf{u}}, \rho$ )
2:   Initialize  $\mathbf{u} = \mathbf{0}$ , and  $\mathbf{f}$  by using the SART algorithm
3:   for  $i$  from 1 to maximum outer iteration do
4:     // generate multi-scale data
5:      $\mathbf{f}^1 \leftarrow \mathbf{f}$ ,  $\mathbf{u}^1 \leftarrow \mathbf{u}$ 
6:     for  $s$  from 1 to  $N_{scales} - 1$  do
7:        $\mathbf{f}^{s+1} \leftarrow \downarrow \mathbf{f}^s$ 
8:        $\mathbf{u}^{s+1} \leftarrow \rho \downarrow \mathbf{u}^s$ 
9:     end for
10:
11:    // update deformation field variables
12:    // from the coarsest scale to the finest
13:    for  $s$  from  $N_{scales} - 1$  to 1 do
14:       $\mathbf{u}^s \leftarrow \text{ESTIMATEDEFORMATIONS}(F_{\mathbf{h}^s}(\mathbf{u}^s), G_{\mathbf{h}^s}(\mathbf{u}^s))$ 
15:       $\mathbf{u}^{s-1} \leftarrow \frac{1}{\rho} \uparrow \mathbf{u}^s$ 
16:    end for
17:     $\mathbf{u} \leftarrow \mathbf{u}^1$ 
18:
19:    // update 4D volume variables
20:     $\mathbf{f} \leftarrow \text{RECONSTRUCTVOLUMEDENSITY}(F_{\mathbf{u}}, G_{\mathbf{u}})$ 
21:  end for
22:  return  $\mathbf{f}$  and  $\mathbf{u}$ 
23: end procedure

```

lowing.

4.4.1 Deformation Field Estimation.

For estimating the deformation field between two volumes, we adapt the multi-scale optical flow approach introduced by Meinhardt-Llopis et al. [85] to include additional regularizers and to work on 3D volumes rather than 2D images.

At each scale s , we estimate the residual deformation between frame \mathbf{f}_t and a backwards warped version of frame \mathbf{f}_{t+1} , where the warping is performed using the previously estimated low resolution deformation field \mathbf{u}_t^{s+1} . We denote this backward-warped frame as $\text{warp}(\mathbf{f}_{t+1}, \uparrow \mathbf{u}_t^{s+1})$, and define the following function

$$\begin{aligned} G_{\mathbf{h}}(\mathbf{u}^s) = & \kappa_3 \sum_{t=1}^{N_k-1} \| (\text{warp}(\mathbf{f}_{t+1}, \uparrow \mathbf{u}_t^{s+1}) - \mathbf{f}_t) \\ & + \nabla_S \text{warp}(\mathbf{f}_{t+1}, \uparrow \mathbf{u}_t^{s+1}) \cdot (\mathbf{u}_t^s - \uparrow \mathbf{u}_t^{s+1}) \|_1 \\ & + \kappa_4 \sum_{t=1}^{N_k-1} \sum_{i=x,y,z} \| \nabla_S \mathbf{u}_{t,i}^s \|_{H_\tau} \end{aligned} \quad (4.6)$$

Based on Equation 5.6, the optimization problem for each scale is then given as

$$\mathbf{u}^{s,*} = \underset{\mathbf{u}^s}{\operatorname{argmin}} \ G_{\mathbf{h}}(\mathbf{u}^s). \quad (4.7)$$

This approach corresponds to a first-order Taylor approximation of the non-linear warping function, and was first proposed by Meinhardt-Llopis [85]; we refer to their work for a detailed derivation and discussion. In order to ensure high accuracy results, the BFECC method [87] was selected for all our warping operations.

Due to the presence of the L1-norm and the Huber penalty in Equation 6.5 we reformulate this optimization in the first-order primal-dual framework (CP algorithm), introduced by Chambolle and Pock [26]. The strategy proposed by this algorithm,

Algorithm 4 CP-based method for deformation field estimation

```

1: procedure ESTIMATEDEFORMATIONS( $F_{\mathbf{h}}, G_{\mathbf{h}}$ )
2:   while not converged do
3:     // update slack variable
4:      $\mathbf{w}^{j+1} \leftarrow \text{prox}_{\lambda_1 G_{\mathbf{h}}^*}(\mathbf{w}^j + \lambda_1 \mathbf{K}_{\mathbf{h}} \bar{\mathbf{u}}^j)$ 
5:     // update deformation field
6:      $\mathbf{u}^{s,j+1} \leftarrow \text{prox}_{\mu_1 F_{\mathbf{h}}}(\mathbf{u}^{s,j} - \mu_1 \mathbf{K}_{\mathbf{h}}^T \mathbf{w}^{j+1})$ 
7:     // update dual variable
8:      $\bar{\mathbf{u}}^{j+1} \leftarrow 2 \cdot \mathbf{u}^{s,j+1} - \mathbf{u}^{s,j}$ 
9:   end while
10:  return  $\mathbf{u}^s$ 
11: end procedure

```

consists in splitting the optimization problem into different sub-problems that are solved independently in the form of proximal operators. The pseudo-code shown in Algorithm 9, summarizes the CP algorithm used to solve the Equation 6.5, where \mathbf{w} and $\bar{\mathbf{u}}$ are respectively the slack and the dual variables. $\text{prox}_{\lambda_1 G_{\mathbf{h}}^*}$ and $\text{prox}_{\mu_1 F_{\mathbf{h}}}$ are the proximal operators, based on the functions $F_{\mathbf{h}}(\mathbf{u}) = 0$ and $G_{\mathbf{h}}$ as defined above. Since $F_{\mathbf{h}}(\mathbf{u})$ is a constant function, its proximal operator is simply the identity. The derivation of $\text{prox}_{\lambda_1 G_{\mathbf{h}}^*}$ is given below in detail.

First, we simplify the notations, by denoting:

$$\begin{aligned}
\tilde{\mathbf{w}}^j &= \mathbf{w}^j + \lambda_1 \mathbf{K}_{\mathbf{h}} \bar{\mathbf{u}}^j \\
\tilde{\mathbf{g}}^j &= \mathbf{g}^j + \lambda_2 \mathbf{K}_{\mathbf{u}} \bar{\mathbf{h}}^j \\
\tilde{\mathbf{b}}_t &= \text{warp}(\mathbf{f}_{t+1}, \uparrow \mathbf{u}_t^{s+1}) - \mathbf{f}_t - \nabla_S \text{warp}(\mathbf{f}_{t+1}, \uparrow \mathbf{u}_t^{s+1}) \cdot \uparrow \mathbf{u}_t^{s+1}
\end{aligned}$$

we insert whole function into G , thus we have: $F_{\mathbf{h}}(\mathbf{u}) = \mathbf{0}$ and

$$\begin{aligned} G_{\mathbf{h}}(\mathbf{K}_{\mathbf{h}} \mathbf{u}^s) &= \kappa_3 \sum_{t=1}^{N_k-1} \| (\text{warp}(\mathbf{f}_{t+1}, \uparrow \mathbf{u}_t^{s+1}) - \mathbf{f}_t) \\ &\quad + \nabla_S \text{warp}(\mathbf{f}_{t+1}, \uparrow \mathbf{u}_t^{s+1}) \cdot (\mathbf{u}_t^s - \uparrow \mathbf{u}_t^{s+1}) \|_1 \\ &\quad + \kappa_4 \sum_{t=1}^{N_k-1} \sum_{i=x,y,z} \| \nabla_S \mathbf{u}_{t,i}^s \|_{H_\tau} \end{aligned} \quad (4.8)$$

where the operator $\mathbf{K}_{\mathbf{h}}$ is defined as:

$$\mathbf{K}_{\mathbf{h}} = \begin{pmatrix} \nabla_S^T & 0 & 0 & \nabla_x \text{warp}(\mathbf{f}_{t+1}, \uparrow \mathbf{u}_t^{s+1}) \\ 0 & \nabla_S^T & 0 & \nabla_y \text{warp}(\mathbf{f}_{t+1}, \uparrow \mathbf{u}_t^{s+1}) \\ 0 & 0 & \nabla_S^T & \nabla_z \text{warp}(\mathbf{f}_{t+1}, \uparrow \mathbf{u}_t^{s+1}) \end{pmatrix}^T = \begin{pmatrix} \mathbf{K}_{11} \\ \mathbf{K}_{12} \\ \mathbf{K}_{13} \\ \mathbf{K}_{14} \end{pmatrix} \quad (4.9)$$

Hence:

$$\tilde{\mathbf{w}}^j = \mathbf{w}^j + \lambda_1 \begin{pmatrix} \mathbf{K}_{11} \\ \mathbf{K}_{12} \\ \mathbf{K}_{13} \\ \mathbf{K}_{14} \end{pmatrix} \bar{\mathbf{u}}^j = \begin{pmatrix} \tilde{\mathbf{w}}_1^{j+1} \\ \tilde{\mathbf{w}}_2^{j+1} \\ \tilde{\mathbf{w}}_3^{j+1} \\ \tilde{\mathbf{w}}_4^{j+1} \end{pmatrix} \quad (4.10)$$

Now it can be shown that the problem is equal to solve a saddle problem :

$$\min_x \max_y \mathbf{K}_{\mathbf{h}} \cdot \mathbf{y} + \mathbf{0} - G_{\mathbf{h}}^*(\mathbf{y}), \quad (4.11)$$

Incorporating it into CP algorithm [26], we obtain:

$$\mathbf{w}_{1,2,3}^{j+1} = \text{prox}_{\lambda_1 G_{\mathbf{h}}^*} \begin{pmatrix} \tilde{\mathbf{w}}_1^{j+1} \\ \tilde{\mathbf{w}}_2^{j+1} \\ \tilde{\mathbf{w}}_3^{j+1} \end{pmatrix} \quad (4.12)$$

$$\mathbf{w}_4^{j+1} = \text{prox}_{\lambda_1 G_{\mathbf{h}}^*}(\tilde{\mathbf{w}}_4^{j+1}) \quad (4.13)$$

$G_{\mathbf{h}}(\cdot)$ is the Huber penalty function. Therefore, the proximal operator of $G_{\mathbf{h}}^*(\cdot)$ is a point-wise shrinkage operation similar to the case of the TV norm [26] with an additional multiplicative term H_{f1} :

$$H_{f1} = \frac{1}{1 + \lambda_1 \cdot \epsilon_1 \frac{\kappa_3}{\kappa_4}}$$

The first term of $G_{\mathbf{h}}(\cdot)$ in Equation 1 is an affine linear L1 norm and the proximal operator can be solved [5] directly as:

$$\mathbf{w}_4^{j+1} = \min(1, \max(\tilde{\mathbf{w}}_4^{j+1} + \lambda_1(\tilde{\mathbf{b}}_t + \nabla_S \text{warp}(\mathbf{f}_{t+1}, \uparrow \mathbf{u}_t^{s+1}) \cdot \bar{\mathbf{u}}^j), -1))$$

With this, $\text{prox}_{\lambda_1 G^*}(\cdot)$ (line 4 in Algorithm 2) is defined. The second proximal operator in the same algorithm, $\text{prox}_{\mu_1 F_{\mathbf{h}}}(\cdot)$ is simply the identity since $F_{\mathbf{h}}(\mathbf{u}) = \mathbf{0}$.

4.4.2 Volumetric Reconstruction

After the deformation field estimation, the 4D volume density (\mathbf{f}) is reconstructed using the optimization framework given in the Equation 4.14, which follows directly from Equation 5.6.

$$\begin{aligned} \mathbf{f}^* &= \underset{\mathbf{f}}{\operatorname{argmin}} \sum_{t=1}^{N_k} \|\mathbf{A}_t \mathbf{f}_t - \mathbf{p}_t\|_2^2 \\ &+ \kappa_3 \sum_{t=1}^{N_k-1} \|\nabla_T \mathbf{f}_t + \nabla_S \mathbf{f}_t \cdot \mathbf{u}_t\|_1 \\ &+ \kappa_1 \sum_{t=1}^{N_k} \|\nabla_S \mathbf{f}_t\|_{H_\epsilon} + \kappa_2 \sum_{t=1}^{N_k} \|\nabla_T \mathbf{f}_t\|_2^2 \end{aligned} \quad (4.14)$$

In order to deal with large-scale deformations, the implementation of the volume update in practice uses volume warping instead of the Horn-Schunck-style energy

term, just as in the deformation estimation. That is, we approximate

$$\nabla_T \mathbf{f}_t + \nabla_S \mathbf{f}_t \cdot \mathbf{u}_t \approx \text{warp}(\mathbf{f}_{t+1}, \mathbf{u}_t) - \mathbf{f}_t. \quad (4.15)$$

Including this term in the volume reconstruction is what allows for aggregation of information from projections across frames, and is the reason for improved volume reconstruction quality.

Algorithm 5 CP-based method for tomographic reconstruction

```

1: procedure RECONSTRUCTVOLUMEDENSITY( $F_{\mathbf{u}}, G_{\mathbf{u}}$ )
2:   while not converged do
3:     // update slack variable
4:      $\mathbf{g}^{j+1} \leftarrow \text{prox}_{\lambda_2 G_{\mathbf{u}}^*}(\mathbf{g}^j + \lambda_2 \mathbf{K}_{\mathbf{u}} \bar{\mathbf{h}}^j)$ 
5:     // update volume variable
6:      $\mathbf{f}^{j+1} \leftarrow \text{prox}_{\mu_2 F_{\mathbf{u}}}(\mathbf{f}^j - \mu_2 \mathbf{K}_{\mathbf{u}}^T \mathbf{g}^{j+1})$ 
7:     // update dual variable
8:      $\bar{\mathbf{h}}^{j+1} \leftarrow 2 \cdot \mathbf{f}^{j+1} - \mathbf{f}^j$ 
9:   end while
10:  return  $\mathbf{f}$ 
11: end procedure

```

As before, a first-order primal-dual algorithm is employed, since Equation 4.14 also contains the Huber penalty and the L1-norm. The used scheme for solving the Equation 4.14 is provided in Algorithm 7, where \mathbf{g} and $\bar{\mathbf{h}}$ are respectively the slack and the dual variables. $\text{prox}_{\lambda_2 G_{\mathbf{u}}^*}$ and $\text{prox}_{\mu_2 F_{\mathbf{u}}}$ are the proximal operators for the volume reconstruction. The function $F_{\mathbf{u}}$ is defined as the first term of the objective function in Equation 4.14, while $G_{\mathbf{u}}$ consists of the remaining terms of the same objective function.

$$\begin{aligned}
G_{\mathbf{u}}(\mathbf{K}_{\mathbf{u}} \mathbf{f}) &= \sum_{t=1}^{N_k} \|\nabla_S \mathbf{f}_t\|_{H_{\epsilon_1}} + \frac{\kappa_2}{\kappa_1} \sum_{t=1}^{N_k-1} \|\nabla_T \mathbf{f}_t\|_2^2 \\
&\quad + \frac{\kappa_3}{\kappa_1} \sum_{t=1}^{N_k-1} \|\text{warp}(\mathbf{f}_{t+1}, \mathbf{u}_t) - \mathbf{f}_t\|_1
\end{aligned} \quad (4.16)$$

Thus the operator $\mathbf{K}_\mathbf{u}$ is given by

$$\mathbf{K}_\mathbf{u} = \begin{pmatrix} \nabla_S, \nabla_T, \mathbf{W} \end{pmatrix}^T = \begin{pmatrix} \mathbf{K}_{21} \\ \mathbf{K}_{22} \\ \mathbf{K}_{23} \end{pmatrix}$$

and

$$\tilde{\mathbf{g}}^j = \mathbf{g}^j + \lambda_2 \begin{pmatrix} \mathbf{K}_{21} \\ \mathbf{K}_{22} \\ \mathbf{K}_{23} \end{pmatrix} \bar{\mathbf{h}}^j = \begin{pmatrix} \tilde{\mathbf{g}}_1^j \\ \tilde{\mathbf{g}}_2^j \\ \tilde{\mathbf{g}}_3^j \end{pmatrix} \quad (4.17)$$

where \mathbf{W} is a warping operator representing the volume warping term $\text{warp}(\cdot, \cdot)$.

Note that for the sake of notational simplicity we are abusing the linear operator notation to also describe the non-linear \mathbf{W} . Specifically, we denote the adjoint of the image warping operator as \mathbf{W}^T . If \mathbf{W} is the backward warp from frame $i+1$ to frame i , the \mathbf{W}^T is the forward warp from frame $i-1$ to frame i .

Incorporating these definitions into the CP algorithm [26], we obtain:

$$\begin{aligned} \mathbf{g}_1^{j+1} &= \text{prox}_{\lambda_2 G_\mathbf{u}^*}(\tilde{\mathbf{g}}_1^j) \\ \mathbf{g}_2^{j+1} &= \text{prox}_{\lambda_2 G_\mathbf{u}^*}(\tilde{\mathbf{g}}_2^j) \\ \mathbf{g}_3^{j+1} &= \text{prox}_{\lambda_2 G_\mathbf{u}^*}(\tilde{\mathbf{g}}_3^j) \end{aligned} \quad (4.18)$$

Similar to Equation 4.12, the solution for Equation 4.18 is a point-wise shrinkage operation multiplied by a Huber factor H_{f2} . For Equation 4.18, the solution is:

$$\mathbf{g}_p^{j+1} = \frac{\kappa_2}{\kappa_2 + 2\kappa_1\lambda_2} \tilde{\mathbf{g}}_p^j \quad p = 1, 2, 3 \quad (4.19)$$

The proximal operator for data term $F_{\mathbf{u}}(\cdot)$ is

$$\text{prox}_{\lambda_2 F_{\mathbf{u}}}(u) = \underset{\mathbf{f}}{\operatorname{argmin}} \|\mathbf{Af} - \mathbf{p}\|_2^2 + \frac{1}{2\lambda_2} \|\mathbf{f} - u\|_2^2. \quad (4.20)$$

This is a least squares problem, and we use SART solving this function iteratively in a matrix free manner[77]. The SART algorithm is applied as a solver for the proximal operator of $F_{\mathbf{u}}(\mathbf{f})$ instead of the traditional conjugated gradient (CG) method, since the CG solver usually requires good preconditioners to be stable, and is generally more sensitive to measurement noise. The update equation for each voxel \mathbf{f}'_j in the volume \mathbf{f}' is:

$$\mathbf{f}'^{(t+1)}_j = \mathbf{f}'^{(t)}_j + \varphi \frac{\sum_{i \in S} \frac{\sqrt{2\lambda_2} p_i - \sqrt{2\lambda_2} \sum_k a_{ik} x_k^{(t)} - y_i^{(t)}}{\sqrt{2\lambda_2} \sum_k a_{ik} + 1} a_{ij}}{\sum_{i \in S} a_{ij}}. \quad (4.21)$$

where t is the iteration, φ is a relaxation parameter, S is a set of projection rays under consideration, and a_{ij} is the element in row i and column j of the system matrix A and defines the contribution to ray sum i from voxel j , and p_i is the measured projection value. In practice, φ is set as 0.5, and 3 iterations of plain SART algorithm are applied as initialization for the proposed optimization framework.

4.5 Experiments

In this section, we first quantitatively assess our reconstruction method using a simulated deforming object. Then, we demonstrate its applicability to six selected deformation phenomena: flow of a high-viscosity fluid, wilting of a rose, re-hydration of a dried mushroom, rising of a dough, magnification of lentil/lupin seeds soaked in water, and dissolution of crystal sugar in water.

We perform experiments on a synthetic dataset to obtain quantitative results for comparisons with alternative acquisition schemes and reconstruction methods. The volume is based on a high-quality CT scan of a static copper foam, depicted in Figure 4.2, on which we impose a synthetic deformation. The volume resolution in

these experiments was $125 \times 250 \times 125$.

4.5.1 Quantitative Evaluation

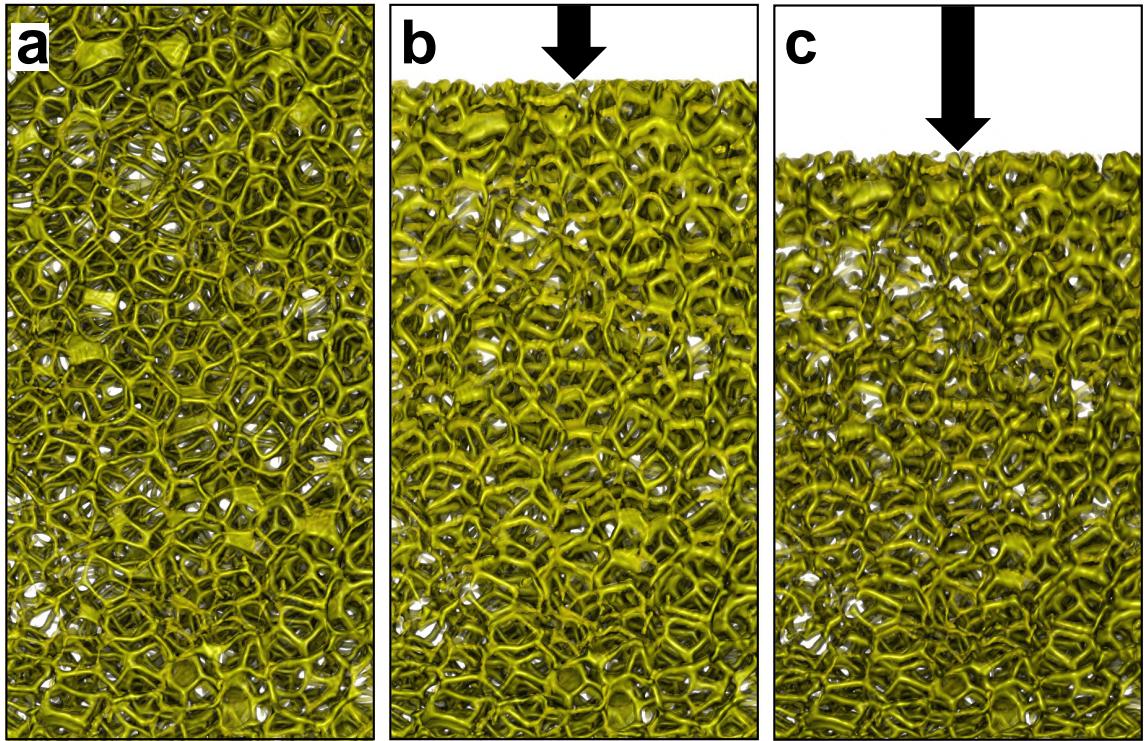


Figure 4.2: Synthetic deformation for a copper foam volume with an initial size of $125 \times 250 \times 125$. (a) Initial volume obtained with real CT acquisition. The deformation is a uniform compression, where the top edge moves downwards at a constant speed, and the bottom edge is stationary, resulting in a linear velocity gradient throughout the volume. Frames (b) and (c) frames 150 and 300 of this sequence.

Uniform compression. The deformation in these experiments is a uniform compression in the vertical direction (Figure 4.2), where the top edge moves downwards by 0.2 voxels between successive projection images. While this is a very simple, non-physical motion, it does allow us to observe a range of different velocities in a single experiment, with $v = 0.2 \text{ voxels}/\Delta t$ at the top edge, $v = 0$ at the bottom edge, and a linear ramp of velocities inbetween. This allows us to collect quality statistics for different horizontal slabs of the volume to analyze the impact of deformation velocity on

the quality of reconstructions for different algorithmic choices and parameter settings. In the following comparisons, we consider the volume generated at the mid-time frame as the ground truth for that time frame.

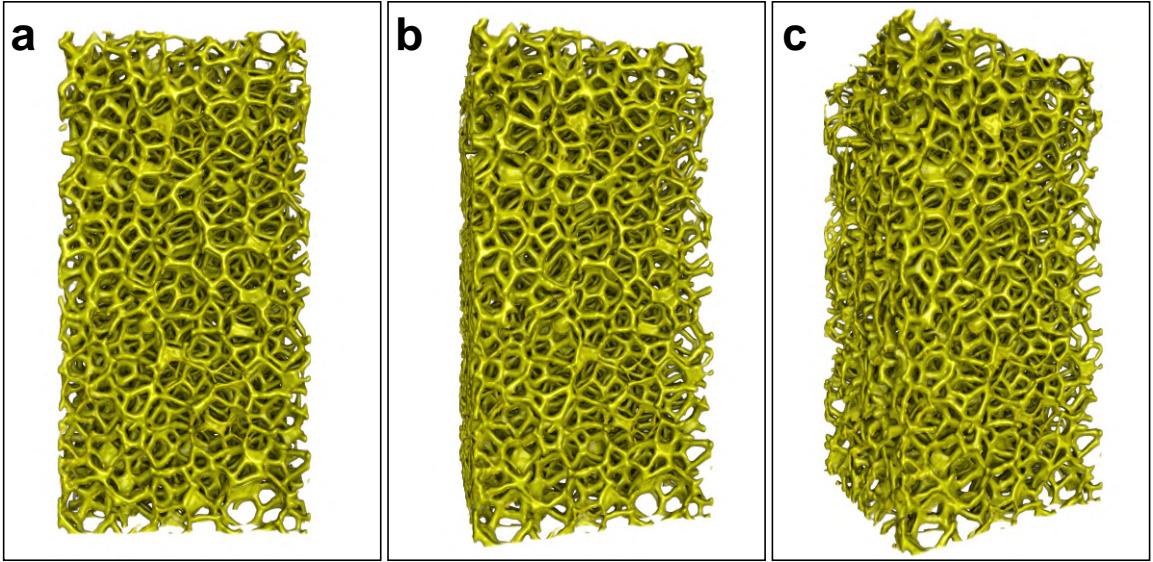


Figure 4.3: Synthetic deformation for a copper foam volume. (a) Initial volume obtained with real CT acquisition. The deformation is a uniform rotation, where the volume rotates from left to right. Frames (b) and (c) frames 150 and 300 of this sequence.

Rotation. We rotated the volume between successive projections with a fixed angle ϕ . Figure 4.3 shows frames 150 and 300 of the sequence with $\phi = 0.1^\circ$. Different values of ϕ were given in the Table 4.1, and demonstrate that the method starts breaking down around values of $\phi > 0.3^\circ$. As stated in the paper, notice that these results can not necessarily be generalized to arbitrary data, since the performance of our method also depends on the amount of local volume structure.

Table 4.1: Calculated PSNR [dB], and SSIM for different rotation velocities [$^\circ/\Delta t$].

Metric	0.1	0.2	0.3	0.4	0.5
PSNR	34.81	30.56	26.15	19.76	16.68
SSIM	0.95	0.88	0.79	0.67	0.54

Impact of the acquisition strategy. First, we study the impact of the view sampling strategy on our space-time reconstruction method. From a total number of $N_p = 160$ projections we reconstruct $N_k = 5$ frames (i.e. we use $N_\theta = 32$ projections per frame). We test three different view sampling strategies: (a) a single **linear** round with $N_p = 160$ projections, (b) the strategy of **Mohan** et al. [80] with $N_\theta = 32$, and (c) **our** low discrepancy sampling, where we use $N_r = 5$ rounds of $N_{\min} = 32$ projections each, since a power of two value for N_{\min} makes our method more directly comparable to Mohan et al.'s method. On all datasets, we attempt our full space-time tomographic reconstruction method.

Table 4.2 shows the numerical results of this experiment separately for five different regions of the volume, corresponding to different velocities of the deformation. Two numerical metrics are collected on the data, namely: Peak SNR (PSNR), and SSIM index. For SSIM comparisons, a binary mask is added to allow us to focus on the geometric comparison. As expected, all the metrics indicate improved reconstruction with decreasing speed of deformation for all sampling strategies. Our low discrepancy strategy offers the best performance at all speeds, with the difference being larger on the fastest moving parts, where the advantages of a good sample distribution in both in space *and* time are the most evident.

Table 4.2: Interaction of the sampling strategy with our reconstruction method. PSNR [dB], and SSIM are applied.

Region	Metric	Linear	Mohan	Our
Top (fastest)	PSNR	18.34	23.83	28.08
	SSIM	0.53	0.75	0.84
	PSNR	21.54	25.27	29.15
	SSIM	0.68	0.80	0.88
Center (medium)	PSNR	25.43	28.17	31.45
	SSIM	0.78	0.87	0.92
	PSNR	28.32	31.44	33.57
	SSIM	0.84	0.93	0.96
Bottom (slowest)	PSNR	30.28	33.23	34.52
	SSIM	0.89	0.96	0.97

Comparison of different reconstruction algorithms. While the previous experiment used our reconstruction method with different sampling strategies, we now compare different reconstruction methods and study in detail the impact of each prior on the reconstructed quality. The methods we compared are:

- plain SART,
- Rudin-Osher-Fatemi total variation denoising model [88], implemented by solving it in a primal dual scheme [26], and using SART as the solver for data term (SART-ROF),
- SART-Huber (SART-H), where the TV term is replaced by the corresponding Huber variant,
- SART-Huber&Temporal smoothing prior (SART-H&T), but no optical flow warping, and
- our full method (Ours).

Considering the structural complexity and relative motion of this simulated data, we choose N_θ as 30, thus we have 10 frames of volumes in total, the numeric results for each metric shown in the Table 4.3 are averaged over all 10 frames. As shown in Table 4.3, the three methods that rely solely on spatial reconstruction (SART, SART-ROF, and SART-H) perform significantly worse than the two methods with using some form of temporal regularization (SART-H&T and Ours). However, even within that latter group, the advantage of our full model including optical flow warping is evident. Analyzing the results for the individual horizontal slices, we note that the errors get worse from bottom to top, but our method degrades much slower than the comparison approaches.

Failure case determination. In order to explore the failure cases of our method, we conduct a compression experiment with larger velocities than before (0.9 voxels/ Δt

Table 4.3: Calculated PSNR [dB], and SSIM for different reconstruction methods: SART, Rudin-Osher-Fatemi (SART+TV), SART+H(uber), SART+H&T(emporal) smoothing, and our full method.

Region	Metric	SART	ROF	H	H&T	Ours
Top (fastest)	PSNR	19.53	21.83	21.88	23.58	28.15
	SSIM	0.65	0.71	0.71	0.76	0.83
	PSNR	21.79	23.13	23.17	24.17	29.03
	SSIM	0.72	0.77	0.77	0.82	0.88
Center (medium)	PSNR	26.67	27.53	27.58	28.69	31.96
	SSIM	0.83	0.86	0.86	0.89	0.93
	PSNR	29.23	30.09	30.07	32.32	33.92
	SSIM	0.89	0.91	0.92	0.93	0.96
Bottom (slowest)	PSNR	30.65	31.07	31.07	32.81	34.35
	SSIM	0.91	0.93	0.93	0.94	0.96

at the top edge of the volume).

Table 4.4: Calculated PSNR [dB], and SSIM for different compression velocities [voxels/ Δt].

Metric	(0.0-0.1)	(0.1-0.3)	(0.3-0.5)	(0.5-0.7)	(0.7-0.9)
PSNR	33.77	27.15	22.52	18.39	15.14
SSIM	0.94	0.81	0.72	0.59	0.48

The Table 4.4 shows how the quality of our reconstruction degrades with faster velocities. For example, if the deformation is larger than 0.5 voxels between successive projections, the PSNR drops below 20. This corresponds to approximately 15 voxels for one time frame of our reconstruction, since we are using 30 projections to reconstruct each time frame. Another experiment was done by rotating the volume between successive projections with a fixed angle ϕ . Different values of ϕ were tested, and demonstrate that the method starts breaking down around values of $\phi > 0.3^\circ$. It must be emphasized that these results cannot necessarily be generalized to arbitrary data, since the performance of our method also depends on the amount of local volume structure. However, overall our synthetic experiments demonstrate that the performance of our method deteriorates gracefully as the speed of motion increases.

4.5.2 Qualitative Evaluation

Table 4.5: Parameters used in the acquisition and for the optimization, and total run times per sub-problem (summed over all outer iterations) for each of the six datasets. The parameters $\kappa_1 = 0.05$ and $\kappa_4 = 1.2$ were fixed for all datasets.

Dataset	Cap. time [h:mm]	N_k	N_θ	Volume size	Vox. pitch [mm]	κ_3	κ_2	# Outer iterations	u-problem [h:mm]	f-problem [h:mm]
rose	9:32	92	60	510×384×456	0.20	0.2	0.3	6	4:48	3:19
fluid	2:12	152	10	384×304×384	0.24	0.2	0.5	6	4:34	1:21
mushrooms	6:05	200	20	240×190×240	0.34	0.1	0.1	6	5:50	1:07
dough	4:27	132	20	276×221×271	0.20	0.1	0.1	8	4:50	0:51
seeds	5:12	60	50	508×332×506	0.17	0.1	0.1	8	2:38	2:02
sugar	0:43	50	10	475×380×475	0.17	0.1	0.1	8	2:09	0:18

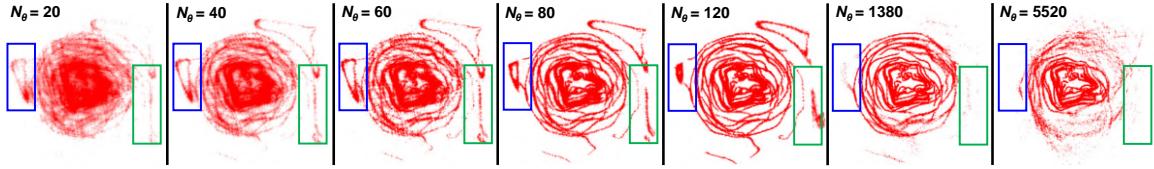


Figure 4.4: Comparison of rose slices obtained after SART reconstruction using a different number of projections (N_θ). The blue and green rectangles highlight the regions where the differences are the most prominent.

To demonstrate the versatility of our proposed approach in studying a variety of dynamic phenomena, as shown in Figure 4.5, we apply it to six real experiments comprising very different types of objects and deformations: (a) a **rose** undergoing significant wilting during the 9.5 hour scan process. The wilting was further accelerated by adding salt to the bottom of the stem. (b) Flow of a high viscosity transparent **fluid** over a 3D-printed mold. This fluid, with 20 million times the viscosity of water, has several included air bubbles that move and pop as the fluid fills the mold. (c) Re-hydration of dried black **mushrooms** from a melting ice cube. (d) Rising **dough**, made from flour and yeast. A hazelnut is placed inside this dough, in order to have additional internal structures. (e) Hydration of a mixture of lentil and lupin **seeds** soaked in water. (f) Dissolution of crystal **sugar** inside water.

Scanning and Optimization Parameters. All of these phenomena were captured on the same CT device, a Nikon XT H 225, where the acceleration voltage of the x-ray

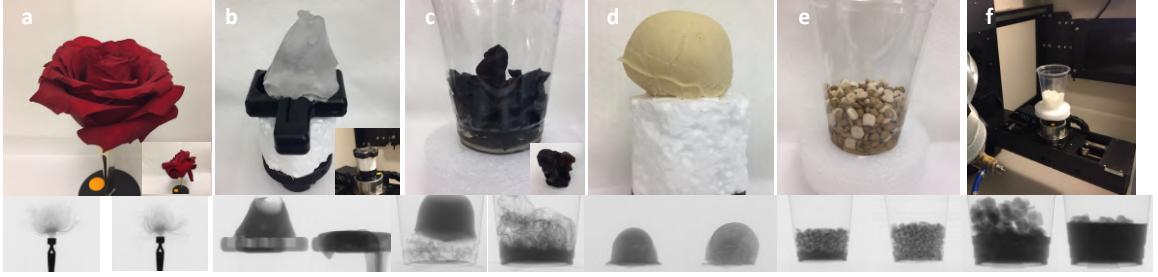


Figure 4.5: Different types of objects and deformations used in the experiments.

tube was in the range of [93 kV, 175 kV], depending on the dataset. The detector of this device has a resolution of 1910×1524 pixels, with a pixel size of 0.127×0.127 mm. Given the different nature of these experiments, their acquisition parameters (capture time, number of frames) are very different. Table 4.5 provides the value of these parameters for all datasets.

In our optimization framework some parameters are common for the six datasets. The weights for the Huber penalty priors are set respectively to: $\kappa_1 = 0.05$ (In practice, κ_1 ranging from 0.01 to 0.1 yields accurate results) and $\kappa_4 = 1.2$. For the multi-scale scheme we choose $N_{scales} = 3$, $\rho = 0.5$, $\sigma = 0.65$ (Algorithm 6). Otherwise, the scales used in the proximal operators are set to: $\lambda_1 = \mu_1 = 0.3$ for the deformation estimation problem (Algorithm 9), and $\lambda_2 = 1.0$, $\mu_2 = 0.1$ for the volume density reconstruction (Algorithm 7). In the inner loops we used two iterations for the SART algorithm and one iteration for the deformation field reconstruction. Other parameters are specific for each dataset (see Table 4.5 for these parameters). κ_2 controls the temporal coherence of each dataset, and usually to get good results κ_2 should be in the range 0.1 to 0.8. Our algorithms were implemented in C++, and were run on a computer with a dual-core 3.00GHz Intel Xeon E5-2687W processor and 512 GB of RAM. The deformation estimation (**u**-problem) and the volume reconstruction (**f**-problem) run times for one complete iteration are also given in the Table 4.5.

The deformation scale and the complexity of the structures are very different from one dataset to another. Consequently, we choose different numbers of projections per

time frame (N_θ) in our reconstructions (see Table 4.5). This parameter is of great importance, and has to be set carefully in order to obtain accurate reconstructions. Intuitively, it should be as high as possible to have a better reconstruction of complex structures in each individual frame, which can then be used for optical flow tracking. On the other hand, N_θ should be small enough to avoid the "motion blur" caused by the deformations. In other words, there is a trade-off between spatial and temporal resolution that requires parameter selection, but unlike in the work of Mohan et al. [80], this parameter choice is strictly post-capture and can be informed by visually analyzing the amount of motion in the raw projection images.

Rose dataset. In Figure 4.4 we illustrate the impact of N_θ on the quality of a standard (3D) SART reconstruction of the wilting rose. This figure shows the same rose slice obtained using reconstructions with different values of N_θ . The central petals of this rose are almost static during the scanning process. Thus, the reconstruction quality of these features improves with an increasing number of projections. This can also be seen in Figure 1.2-(c). However, when the structures are dynamic, using a large number of projections yields blurred reconstructions, causing these features to disappear as their density gets distributed over many voxels. Note that for Figure 1.2-(c), we use a threshold for the rendering, that is why these dynamic structures look less blurred in the 3D rendering, but certain features are missing entirely. The blue and green rectangles in Figure 4.4 illustrate two dynamic features of the rose. One can notice that for N_θ higher than 60, some details of these features are blurred. Therefore, for this dataset we choose to use 60 projections to reconstruct each time step using our space-time method. Moreover, Figure 1.2-(f) shows that the reconstructed rose is quite similar for successive frames (Frames 01-02 and Frames 91-92), which reinforces our choice of N_θ . For other datasets we also set this parameter empirically, by looking at the changes in projections of similar angles.

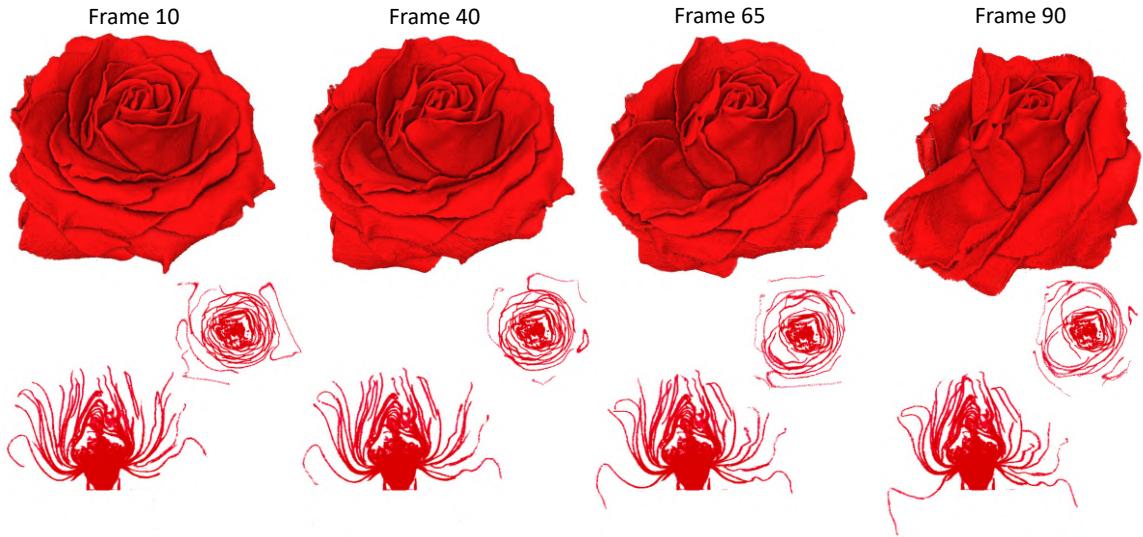


Figure 4.6: Reconstruction obtained for the wilting rose dataset. A 3D rendering of the rose is given at four different time frames (first row). The second and the third rows show respectively a top and a side slices for the same time frames.

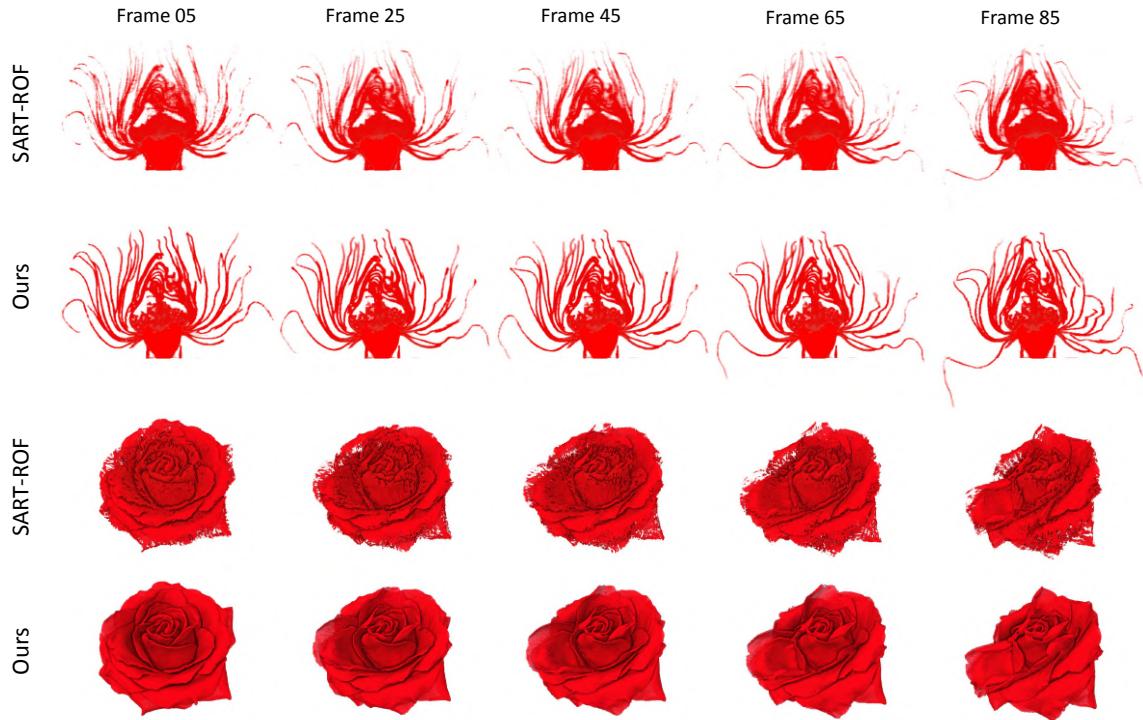


Figure 4.7: Comparison for rose dataset between SART-ROF and our method.

The reconstructions shown in Figure 4.6, and Figure 4.7 demonstrate the accuracy of our approach. Note that the petals curl in on themselves, creating a quite intricate

deformation field. Our non-parametric method can handle this situation well, while fitting an appropriate FEM model to this data would be extremely challenging. Despite the high geometric complexity of this dataset, as well as the low contrast in the projections (see Figure 1.2-(d)), our reconstruction provides sharp details of the rose during the whole wilting process. This makes our approach very suitable for studying botanical processes that happen at time scales of the scan process.

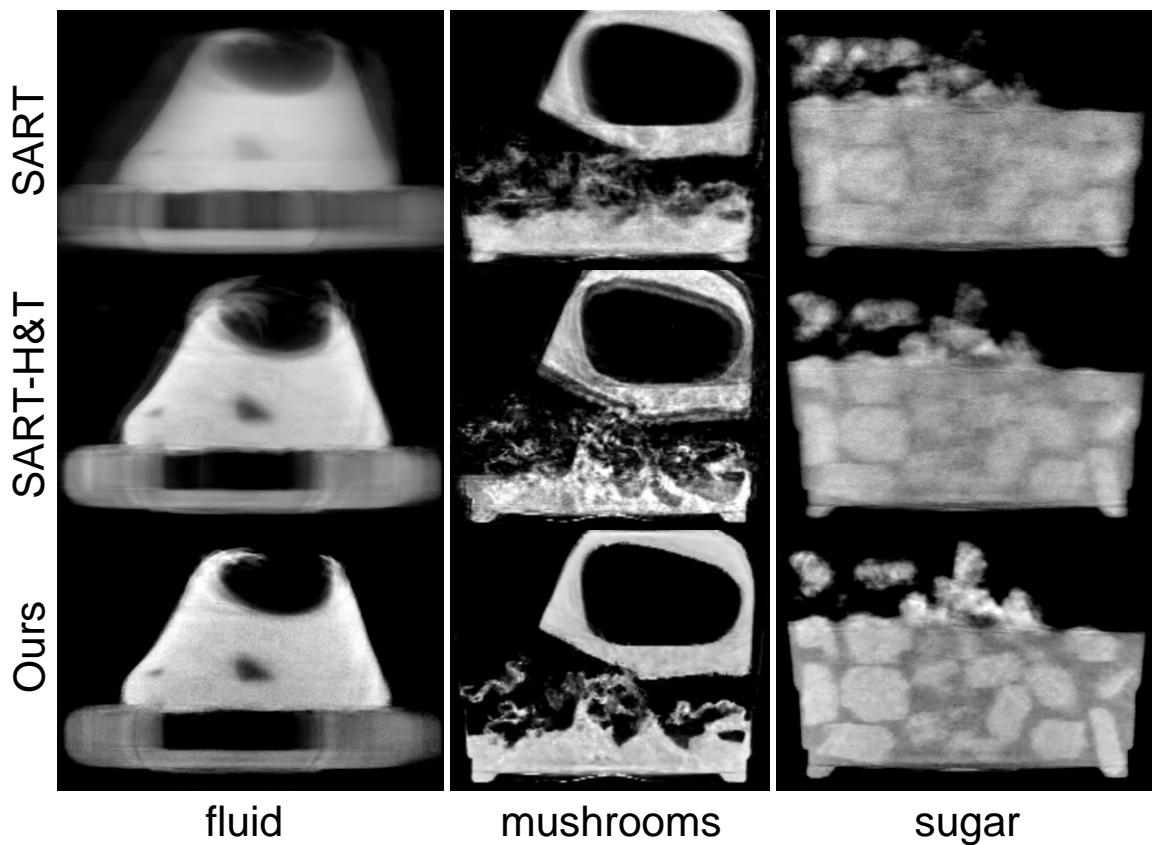


Figure 4.8: Comparison of three reconstruction approaches for different datasets (from left to right: fluid, mushrooms and sugar datasets). The first row corresponds to the SART result. The second row are reconstructions using the SART algorithm with a Huber penalty and a temporal smoothing prior. The last row shows the reconstruction obtained with our method.

Fluid dataset. The second dataset (see Figure 4.12 and Figure 4.13) is composed of a static part (mold) and a dynamic part (high viscosity fluid, (PSF-20,000,000 cSt Pure Silicone Fluid) that contains internal structures (bubbles). Given the relatively

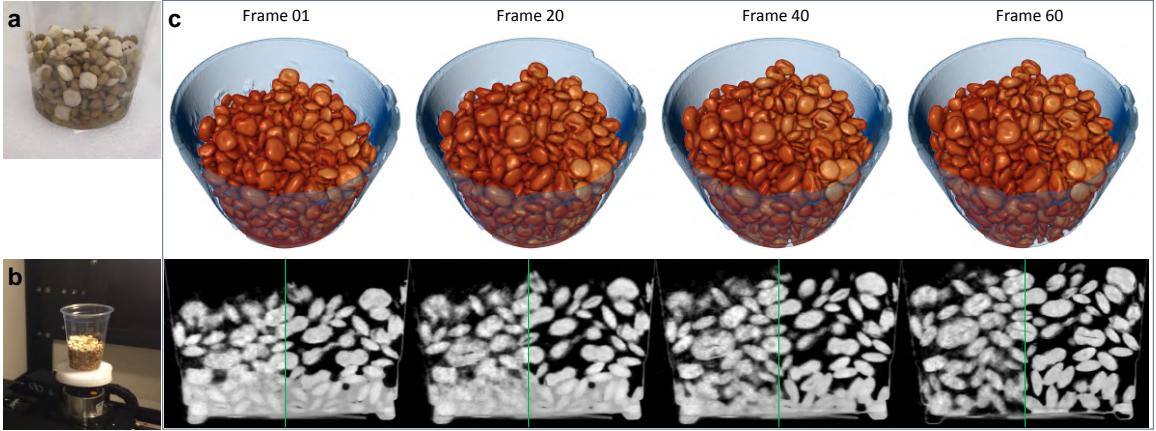


Figure 4.9: Reconstruction obtained for the lentils/lupin seeds dataset. (a) and (b) Captured photographs of the seeds sample before and at the end of the scan. (c) 3D rendering (top) and slices visualization (bottom) of the obtained result for 4 different time frames. For these slices the left half corresponds to the reconstruction using a classical method (SART-ROF); while the right half corresponds to our reconstruction result.

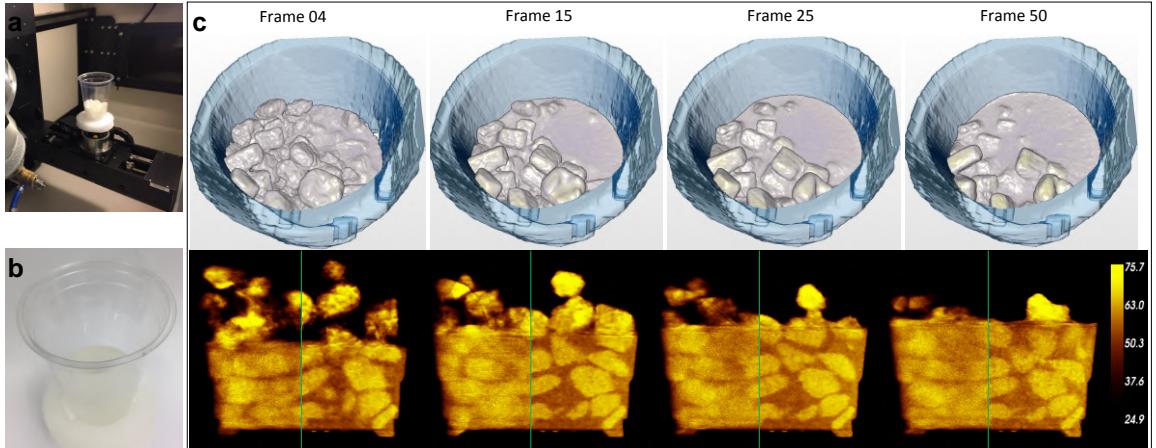


Figure 4.10: Reconstruction obtained for the crystal sugar dataset. On the left side, photographs of the sample before (a) and after (b) the scanning process are given. On the right side (c), a 3D rendering for four time frames is given (top), as well as a slices visualization for the same time frames (bottom). For these slices the left hand part corresponds to the reconstruction using a classical method (SART-ROF); while the right side corresponds to our reconstruction result.

fast motion of the fluid, only ten projections are used to reconstruct each time frame. Thus, when a SART algorithm is applied on this dataset, the volumes are poorly reconstructed (see the first column of Figure 4.8). Adding just the spatial priors – the Nonlinear TV prior (ROF) or the Huber penalty prior – is not sufficient to improve

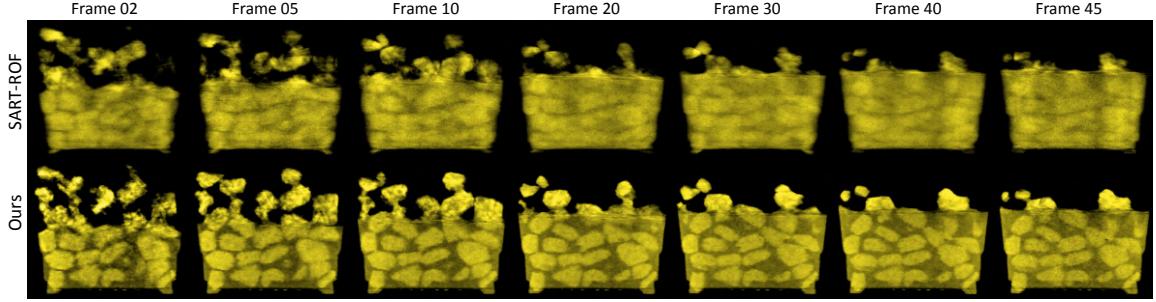


Figure 4.11: The visualization of the sugar dataset, SART-ROF is compared with our method.

the reconstruction quality. Moreover, on the provided video we observe a temporal incoherence between the retrieved volumes at successive time steps. With our reconstruction, this is solved by adding temporal smoothness and optical flow priors. The temporal prior ensures a temporal coherence for the reconstructed volumes. In addition, it allows enriching the angular information, for both the static parts (mold) and the dynamic parts (fluid), since projections used in successive time frames are different. The accuracy of the reconstruction is then improved, especially for static and quasi-static features. In Figure 4.8, we see clearly that the SART-H&T algorithm (SART algorithm + Huber penalty + temporal prior) has a better reconstruction for the mold. But since the top of the fluid has large displacements, we observe some blurring and smearing effects. This is solved in our approach by using the optical flow prior and the joint optimization framework. Figure 4.8 shows the improvement of our method in the reconstruction of the dynamic features of the fluid.

Mushroom dataset. Similar observations can be made for the slices representing the mushrooms dataset shown in the middle column of Figure 4.8. In this data, a partially frozen ice cube with a liquid water core is placed on a bed of dried mushrooms (*Auricularia auricula-judae*). The slice sequence (Figure 4.14-(f)) makes it possible to follow the complete process of re-hydration of the mushrooms. The first image illustrates the presence of a water cavity inside the ice, showing that the cube was

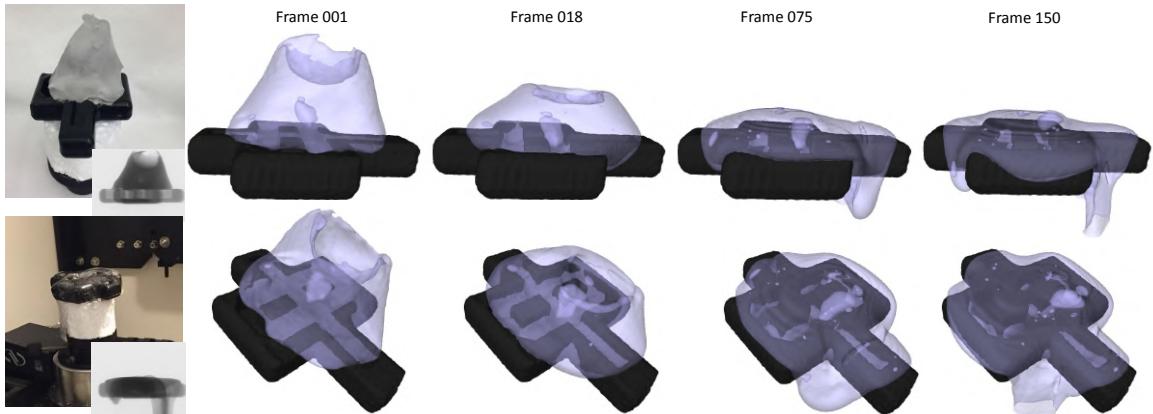


Figure 4.12: Reconstruction obtained for the high viscosity fluid dataset. Two photographs are given for the before and after scanning process (left). These photos are accompanied by thumbnail images of the first and last projections. A 3D rendering is given for four different time frames from two different views (right).

not frozen completely. The water has higher density than ice in the x-ray images, as is to be expected. The second image shows the beginning of the melting of the ice. The thickness of the ice is reduced, and liquid water is now present at the bottom of the container. On the top side of the ice the density is darker than before, while it is brighter on the bottom side. This shows how the liquid water drains from the cavity in the ice cube and is replaced by air. In the third image, all the liquid water leaves the cavity and ends up on the bottom of the container. The remaining four images show the continuous re-hydration of the mushrooms from the melting ice. The final state of this process is illustrated by the photograph in Figure 4.14-(d) and the 3D rendering in Figure 4.14-(e) and Figure 4.15.

As can be seen in Figure 4.8, the SART and the SART-H&T approaches fail in reconstructing sharp features, especially for the cavity inside the melting ice. Figure 4.14-(a),(b) and (c) shows a comparison between the SART, the SART-ROF algorithms and three different iterations of our approach for the last time frame. At this stage all the ice has melted, which accelerates the re-hydration of the mushrooms. The increased rate of deformations in the mushrooms causes a reduction in reconstruction quality for the SART and SART-ROF algorithms. After few iterations of

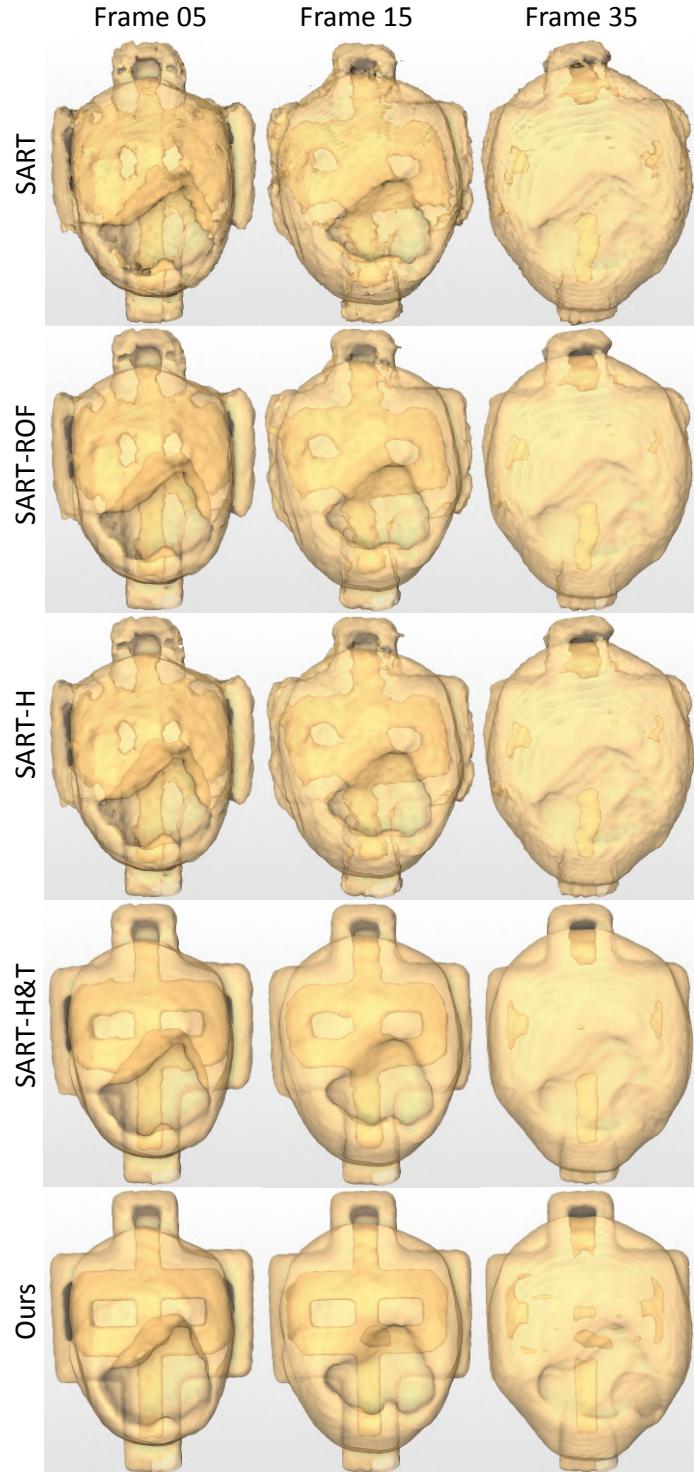


Figure 4.13: Reconstruction obtained for the high viscosity fluid dataset. Two photographs are given for the before and after scanning process (left). These photos are accompanied by thumbnail images of the first and last projections. A 3D rendering is given for four different time frames from two different views (right).

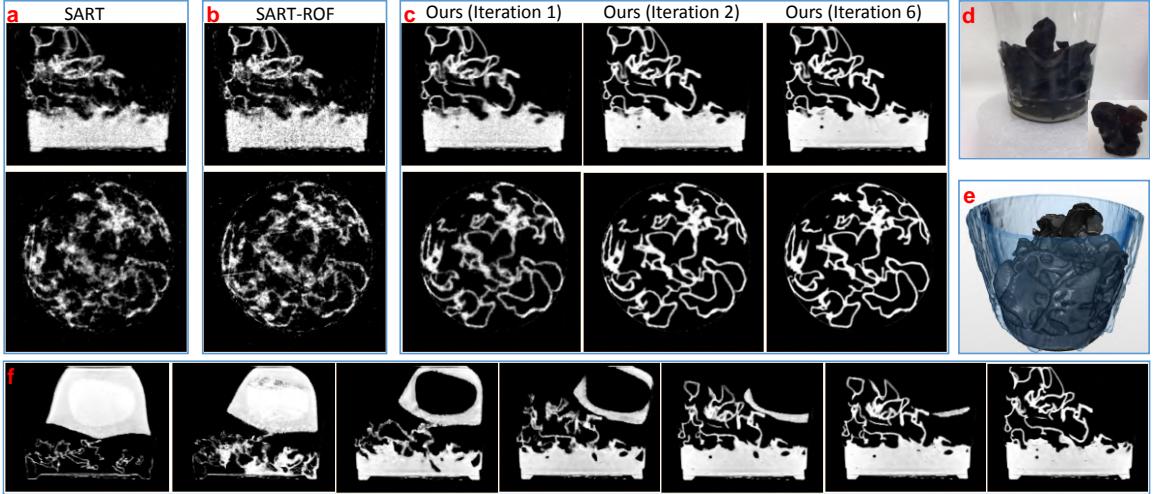


Figure 4.14: Reconstruction obtained for the dried black mushrooms dataset. A comparison of different reconstruction methods at one time frame is shown for 2 slices in (a) SART,(b) SART-ROF and (c) Our method with 1 iteration (left), 2 iterations (middle) and 6 iterations (right). Images (d) and (e) represent respectively a photograph and a 3D rendering of the last time frame. Finally, a time sequence of the same slice illustrates the steps of the melting ice and the re-hydration of the mushrooms in (f).

our algorithm the result is accurate.

Dough dataset. The fourth dataset corresponds to a rising dough shown in the photographs in Figure 4.17-(a) and (b). In comparison to the rose, the projections of the dough have a good contrast (see Figure 4.17-(d)). In addition, the dough initially has a relatively simple overall geometric shape comparing to the rose. Thus, the classical method reconstructs the external shape with good accuracy (see Figure 4.17-(e) and (Y)). However as the dough rises, the yeast creates air bubbles of different sizes. Many air bubbles are too small to be geometrically resolved by the CT scanner. Their impact is visible through a change in the absorption coefficient of the dough. Other bubbles, however, are developing at a macroscopic scale, and show up as internal, sponge-like structures in the geometry. In the classical reconstruction method, these structures are blurred out due to the low projection count. Our space-time reconstruction (see Figure 4.17-(f) 4.17, and (Y')), however, is able to recover

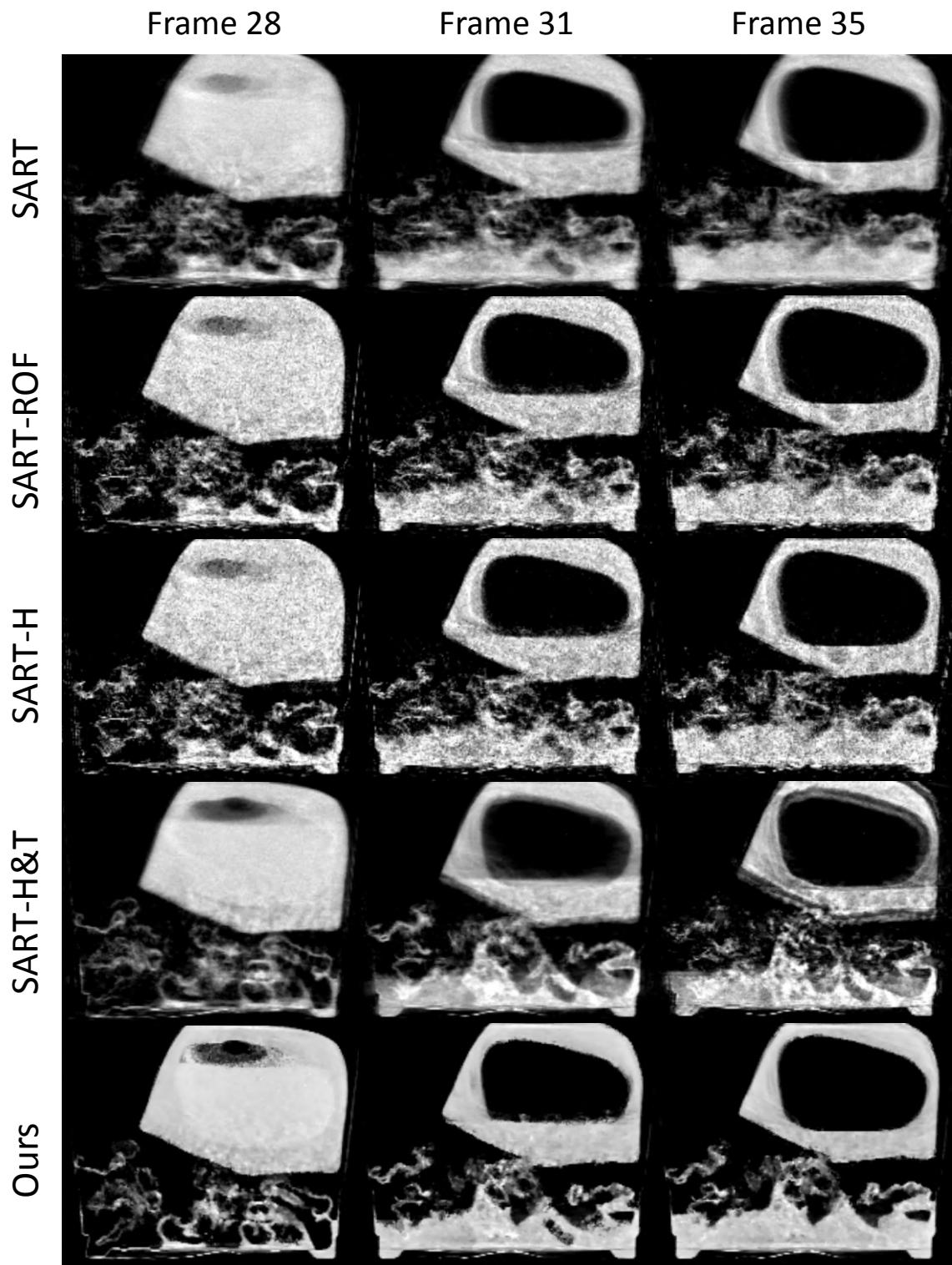


Figure 4.15: Slice visualization and comparison for mushrooms data.

these bubbles quite well.

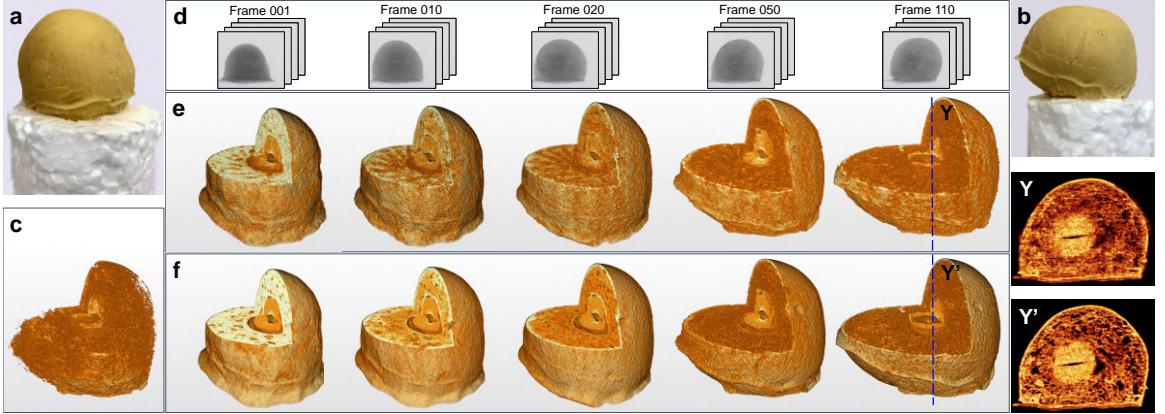


Figure 4.16: Reconstruction obtained for the rising dough dataset. Images (a) and (b) show photographs of the dough directly before and directly after the scanning process. Traditional CT reconstructions from all 132 frames (d) in the scan sequence show significant distortions due to misalignment of features in the x-ray projections (c). While the deformation is gradual enough to be negligible over shorter sequences of 20 successive images, this number of projections is too small for reconstructing accurately the internal structures of the dough (e, \mathbf{Y}). By comparison, our full space-time reconstructions algorithm yields a time sequence of highly detailed volumes for different time steps (f, \mathbf{Y}').

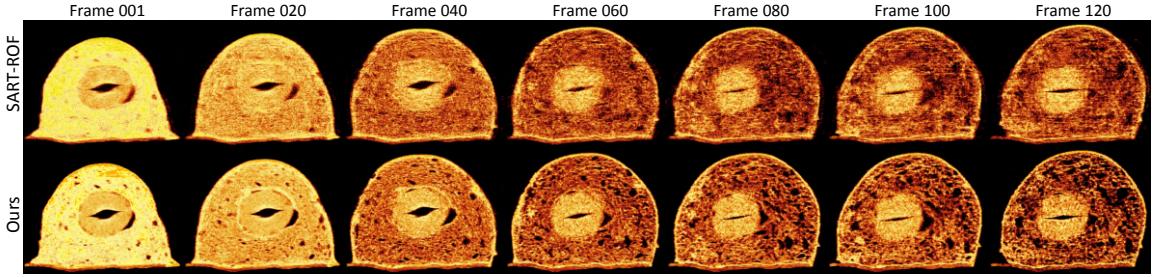


Figure 4.17: The visualization of the dough dataset, SART-ROF is compared with our method.

Seeds dataset. Figure 4.9 illustrates the results obtained for the reconstruction of the seeds dataset. Before the scan, the seeds were soaked in water for several minutes, but the water was then mostly drained for the actual scan. By absorbing water these seeds increase in size, and push the other seeds upward. The slices in Figure 4.9 point up that the magnification of seeds is more important for those in the bottom of the

container. We can notice also that at the last frame, all the water is absorbed.

For this dataset, the impact of the "motion blur" is clearly seen on the SART-ROF reconstruction (left side of the slices). This artifact it is well corrected using our approach.

Sugar dataset. The last experiment consists of imaging the dissolution of sugar crystals in water. A comparison of different reconstruction methods (SART, SART-ROF, SART-H&T and ours) is also given for this dataset in the right column in Figure 4.8, as well as in Figure 4.10. Our reconstruction outperforms the other methods for this dataset as well, by yielding sharper features. When comparing the slices corresponding to Frame 04 and 50 in Figure 4.10, we notice the density of the water increases slightly over time. This can be explained by the presence of dissolved sugar molecules in the water at the Frame 50. Finally, in the provided video we can see some temporal incoherence in the beginning of the sequence of this dataset. For this case, we are out of the scope of our method since the deformation time scale is smaller than the scanning time for ten projections (here $N_\theta = 10$).

4.6 Conclusion

We have presented a method for space-time tomographic reconstruction of objects that undergo significant deformations during the scanning process. We demonstrate our method on a wide variety of input data, ranging from deforming surfaces (e.g. rose, mushrooms) to volumes with changing internal structure (e.g. rising dough).

The success of our method relies on two novel contributions. First, we devise a new sampling strategy for selecting a sequence of viewing angles from which to obtain the x-ray projections. This strategy provides a dense, approximately uniform coverage of angles vs. time, and can be implemented on commercial x-ray CTs without modifications. Furthermore, the sampling strategy does not require any data-dependent

parameter selection and can naturally handle both static objects as well as deformations of various magnitudes and speeds.

The second component of our method is a joint image formation model and optimization method for simultaneously recovering a sequence of shapes over time, as well as the deformation fields between them. By jointly solving for both variables, we successfully transfer information between time steps of different deformation states, and are able to overcome quality and resolution issues that would result from independent reconstructions of each frame. Our method is non-parametric; we do not require projections of either the volumes or the motions into basis functions. This makes it easy to apply our method to a wide range of geometries with varying topologies without manual tweaking.

The limitation of the approach is that we still have to assume a relatively small motion at the scale of a small number of projections (10-60 in our examples). Our method does not work if there already is significant deformation between two successively captured projections. In the next chapter, we will address this situation by moving to a continuous time scale, in which we respect the exact capture time for each individual projection image in the optimization method.

Despite this limitation on deformation speed, our method already provides an efficient and effective means of dealing with deformations in x-ray computed tomography. This allows to analyze time-varying phenomena with interesting changes in internal structure at time scales that could not previously be handled.

Chapter 5

Warp-and-Project Tomography

In this chapter, we show a new warp-and-project approach for dynamic tomographic reconstruction, which is designed for larger deformations even between successive projections. The method not only estimates the volume densities over time, but also the motion field. We also decouple the frame rate of the reconstructed volume sequence from the acquisition times for the captured projections. Finally, the temporal sampling is also adaptive, since additional volumes are reconstructed during periods of rapid motion.

5.1 Introduction

In Chapter 4, we present a non-parametric Space-Time tomographic method (ST-tomography) to scan and analyze deforming objects and dynamic phenomena. While this method resulted in marked improvement of the state of the art, it does suffer from several shortcomings that we address in this chapter: First, ST-tomography was conceived for the case of relatively slow and smooth motion fields, where the deformation is negligible for short sequences of $\approx 10 - 60$ successive frames. Second, the method relies on an explicit trade-off between spatial and temporal reconstruction quality. Finally, the temporal sampling is uniform, resulting in wasted computational effort for slow moving periods, as well as poor reconstruction quality for fast moving periods.

In this chapter, we propose a new warp-and-project approach for dynamic tomographic reconstruction. This new method, inspired by ST-tomography introduced in Chapter 4, relaxes the assumption of slow deformations in order to reconstruct objects with larger motion even between successive projections, where the ST-Tomography fails. This new method not only estimates the volume densities over time, but also the motion field. To this end, we move from a coarsely discretized time axis in ST-tomography [24] to an essentially continuous time axis, where each projection image has its own time stamp, and warping is used to align the keyframes with the individual projections. We would like to highlight that the complexity of a deformation has no impact in our comparison to ST-tomography, only the speed of the motion matters. We also decouple the frame rate of the reconstructed volume sequence from the acquisition times for the captured projections. Finally, the temporal sampling is also adaptive, since additional volumes are reconstructed during periods of rapid motion. These improvements translate into significant improvements in the reconstruction results, as demonstrated by quantitative comparisons on simulated data, as well as qualitative comparisons on real data from a number of different application domains.

In summary, the main contributions of this chapter include:

- a new image formation model for dynamic tomography reconstruction, that takes into account the deformation occurring between successive captured projections.
- a temporal decoupling between the reconstructed key frames and the captured projections.
- a non-uniform temporal up-sampling, which will improve the quality of each reconstruction.
- a matrix-free solver for the proposed optimization algorithm.

- a strong evaluation of our approach both on simulated data, controlled experimental data (where a ground truth can be estimated) and several real data sets highlighting different application scenarios.

5.2 Warp-and-Project Tomography

5.2.1 Image Formation Model

In dynamic x-ray tomography the aim is to reconstruct a 4D volume \mathbf{f} that represents the scanned deforming object, from N_p acquired projections. While previous work [24] assumed that the object motion between a small number of successive frames is negligible, we target situations where the motion can be significant even between two images taken in immediate succession.

Assume that the CT scan consists of N_p projection images $\mathbf{p} = \{\mathbf{p}_1, \dots, \mathbf{p}_j, \dots, \mathbf{p}_{N_p}\}$ taken at times $\{t_1, \dots, t_j, \dots, t_{N_p}\}$. For the description in the following we loosely assume that the projection images are taken at regular intervals, i.e. $t_j = j \cdot \Delta t$, although the framework also works for irregular temporal sampling patterns. Due to the continuous motion of the scan target, the volume is different at each capture time t_j , and is denoted as $\mathbf{f} = \{\mathbf{f}_j\}$. The relationship between the captured projections and the time varying volumes is then described as

$$\underbrace{\begin{pmatrix} \mathbf{A}_1 & & \\ & \ddots & \\ & & \mathbf{A}_j & \\ & & & \ddots & \\ & & & & \mathbf{A}_{N_p} \end{pmatrix}}_{\mathbf{A}} \cdot \underbrace{\begin{pmatrix} \mathbf{f}_1 \\ \vdots \\ \mathbf{f}_j \\ \vdots \\ \mathbf{f}_{N_p} \end{pmatrix}}_{\mathbf{f}} = \underbrace{\begin{pmatrix} \mathbf{p}_1 \\ \vdots \\ \mathbf{p}_j \\ \vdots \\ \mathbf{p}_{N_p} \end{pmatrix}}_{\mathbf{p}}, \quad (5.1)$$

where $\mathbf{A}_j \in \mathbb{R}^{M \times N_v}$ is the projection matrix for *a single* projection, mapping the

volume \mathbf{f}_j with N_v voxels to a projection image \mathbf{p}_j with M pixels. Also see Figure 5.1.

We note that Eq. (5.1) is a heavily ill-posed problem that cannot be solved without additional priors such as motion models, since a single 2D projection image \mathbf{p}_j is insufficient for reconstructing a whole 3D volume \mathbf{f}_j . At the same time, it is often not necessary to reconstruct all N_p volumes to get a good representation of the shape and its deformation, since the changes can still be gradual and smooth. We therefore subsample the volumes \mathbf{f} into a set of N_k key frames $\mathbf{h} = \{\mathbf{h}_1, \dots, \mathbf{h}_k, \dots, \mathbf{h}_{N_k}\}$ corresponding to times T_k . The output of our reconstruction method is this set of key frames, together with a set of motion fields $\mathbf{u} = \{\mathbf{u}_1, \dots, \mathbf{u}_{N_k-1}\}$, where \mathbf{u}_k describes the deformation between \mathbf{h}_k and \mathbf{h}_{k+1} . We choose N_k adaptively, starting with a small number and increasing it until the deformation between key frames is small enough.

Given estimated key frames \mathbf{h}_k and motion fields \mathbf{u}_k , we can approximate the volume at a time t_j with $T_k \leq t_j < T_{k+1}$ by warping (advection) the key frame \mathbf{h}_k *forward* in time:

$$\mathbf{f}_j \approx \tilde{\mathbf{f}}_j = \text{warp}(\mathbf{h}_k, \frac{t_j - T_k}{T_{k+1} - T_k} \mathbf{u}_k). \quad (5.2)$$

Similarly, we can obtain another estimate by *backward* warping the next key frame:

$$\mathbf{f}_j \approx \tilde{\mathbf{f}}_j = \text{warp}(\mathbf{h}_{k+1}, \frac{t_j - T_{k+1}}{T_{k+1} - T_k} \mathbf{u}_k). \quad (5.3)$$

For ease of notation, we introduce warping operators \mathbf{W}_j^f and \mathbf{W}_j^b that respectively perform forward and backward warping to create two estimates of the intermediate frame $\tilde{\mathbf{f}}_j$ from the previous (resp. next) key frame, i.e. $\tilde{\mathbf{f}}_j = \mathbf{W}_j^f(\mathbf{h}_k)$ and $\tilde{\mathbf{f}}_j = \mathbf{W}_j^b(\mathbf{h}_{k+1})$. The image formation model from Eq. (5.1) then corresponds to two separate data terms that can be utilized in an optimization-based reconstruction:

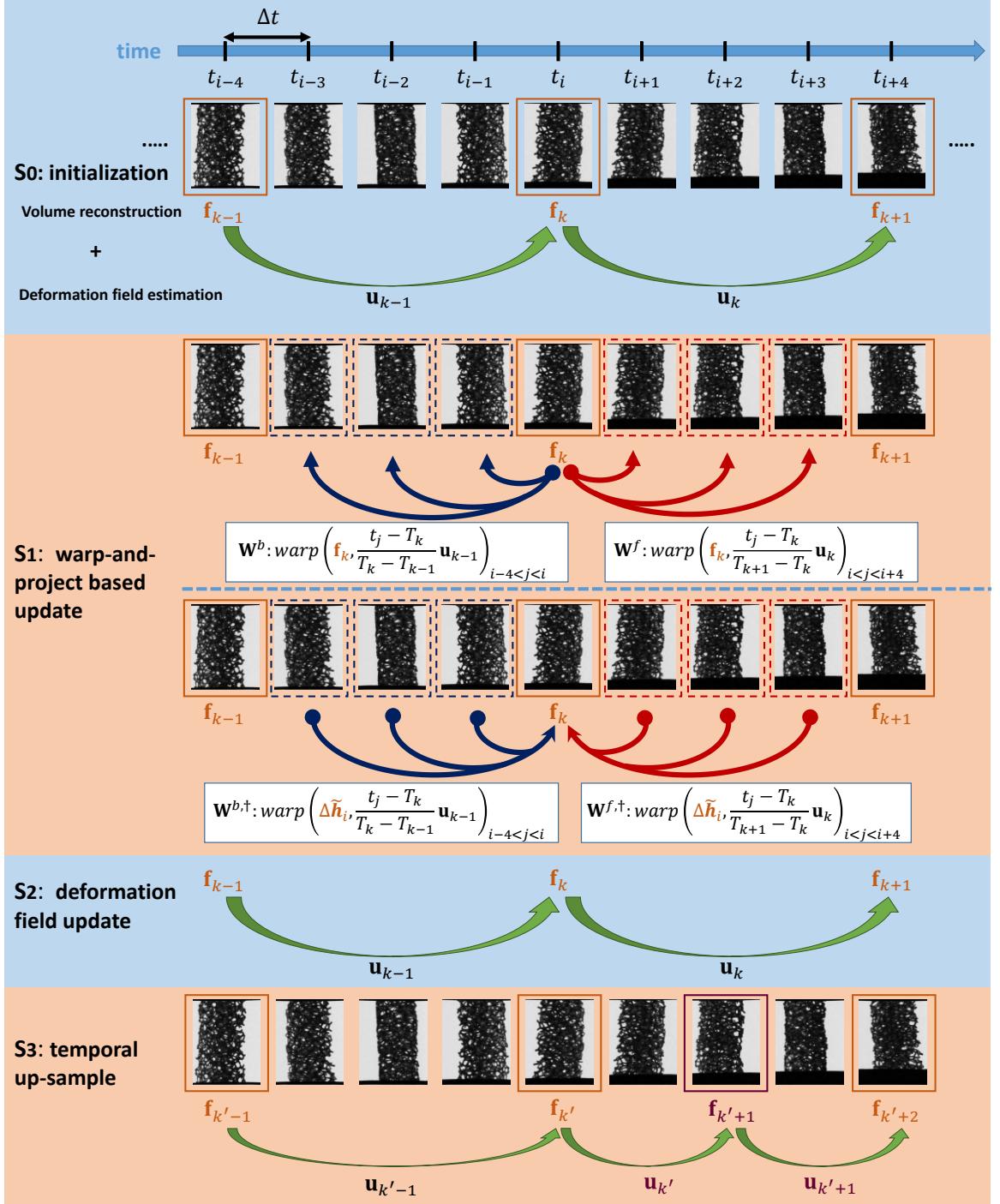


Figure 5.1: Diagram of our method. S0: the key frames and the motion field between the key frames are initialized to 0. S1: volume reconstruction. The current volume estimate for a key frame is warped to the time of each X-ray image, and a residual image is computed by comparing the X-ray Image with the projection of this warped volume. The residual is then back-projected into 3D, and warped back to the time of the key frame to update the volume estimate. S2: flow (velocity) reconstruction between subsequent key frames. S3: temporal up-sampling. New key frames are inserted where the motion is fast.

$$\mathbf{A} \cdot \mathbf{W}^f(\mathbf{h}) = \mathbf{p} \text{ and} \quad (5.4)$$

$$\mathbf{A} \cdot \mathbf{W}^b(\mathbf{h}) = \mathbf{p} \quad (5.5)$$

For implementing these data terms in an optimization approach, we also require the adjoint operators $\mathbf{W}^{f,\dagger}$ and $\mathbf{W}^{b,\dagger}$. However, these are easily implemented as the corresponding warps in the opposite direction, followed by an averaging of all warped volumes that contribute to a single keyframe.

Figure 5.1 shows a diagram of this warping-based interpolation of the intermediate volumes from the neighboring key frames, which is the key distinguishing characteristic of our method compared to the existing state of the art. Without this warping-based approach, all projections used in the reconstruction of a key frame are implicitly assumed to have been taken at the same time (i.e. representing the same shape). This is the approach taken by Chapter 4, and it results in blurred reconstructions for faster motions.

5.2.2 Full Optimization Problem

Given the two data terms from above, we can now formulate an objective function for reconstructing the deforming geometry jointly with the motion field. Due to the ill-posed nature of this problem, this requires additional regularizers for both the key frames and the deformation field, which we adopt from the work of [24]:

$$\begin{aligned}
& \min_{\mathbf{h}, \mathbf{u}} \sum_{j=1}^{N_p} \left\| \mathbf{A}_j \mathbf{W}_j^f(\mathbf{h}_{j-}) - \mathbf{p}_j \right\|_2^2 + \sum_{j=1}^{N_p} \left\| \mathbf{A}_j \mathbf{W}_j^b(\mathbf{h}_{j+}) - \mathbf{p}_j \right\|_2^2 \\
& + w_1 \sum_{k=1}^{N_k-1} \left\| \nabla_T \mathbf{h}_k + \nabla_S \mathbf{h}_k \cdot \mathbf{u}_k \right\|_1 \\
& + \sum_{k=1}^{N_k} [w_2 \left\| \nabla_S \mathbf{h}_k \right\|_{H_\epsilon} + w_3 \left\| \nabla_T \mathbf{h}_k \right\|_2^2] \\
& + \sum_{k=1}^{N_k-1} \sum_{i=x,y,z} [w_4 \left\| \nabla_S \mathbf{u}_{k,i} \right\|_{H_\tau} + w_5 \left\| \nabla_T \mathbf{u}_{k,i} \right\|_2^2],
\end{aligned} \tag{5.6}$$

where \mathbf{h}_{j-} and \mathbf{h}_{j+} refer to the key frames immediately before and immediately after projection \mathbf{p}_j . Here, w_1, w_2, w_3, w_4 and w_5 are weights of the different terms of the objective function. The operators ∇_T and ∇_S correspond to the temporal and spatial discrete gradient, implemented as one-side divided differences. The first two terms correspond to the two warping-based data terms. The second line in this objective function corresponds to a 3D version of the brightness constancy term in the Horn-Schunck optical flow [84]. In order to deal with large deformations we opted for a multi-scale implementation of the optical flow [85]. The next two terms in the third line correspond respectively to the spatial and temporal regularizations of the density volumes. A Huber penalty [86] is used on the spatial gradient with a positive parameter ϵ , while we favor smooth behavior in the time domain with an L2-norm. Similar regularizations are also used for the deformation field in the two terms of the fourth line. τ is the positive parameter of the Huber penalty on the spatial gradient of the deformation field.

The framework that we propose in this chapter is presented in Figure 5.1. A detailed description of this framework is also given as pseudo-code in Algorithm 6. First, an initialization of the density volumes and the deformation fields is performed (Step S0). Then, a Warp-and-Project update scheme (step S1) is applied to improve

Algorithm 6 Warp-and-Project Tomography

```

1: procedure WP-TOMOGRAPHY( $F_h^2, G_h^2, F_u^2, G_u^2, F_u^W, G_u^W, \rho$ )
2:   // Step S0: initialize the key frames and deformation fields.
3:    $\mathbf{h} \leftarrow 0; \mathbf{u} \leftarrow 0;$ 
4:   while not converged do
5:     // Step S1: update volumes by comparing the projections
6:     // of warped volumes with captured projections
7:      $\mathbf{h} \leftarrow \text{WARPANDPROJECT}(F_u^W, G_u^W)$ 
8:
9:     // Step S2: update deformation field between key frames
10:    // from the coarsest scale to the finest
11:    // generate multi-scale data
12:     $\mathbf{h}^1 \leftarrow \mathbf{h}, \mathbf{u}^1 \leftarrow \mathbf{u}$ 
13:    for  $s$  from 1 to  $N_{scales} - 1$  do
14:       $\mathbf{h}^{s+1} \leftarrow \downarrow \mathbf{h}^s$ 
15:       $\mathbf{u}^{s+1} \leftarrow \frac{1}{\rho} \downarrow \mathbf{u}^s$ 
16:    end for
17:    for  $s$  from  $N_{scales} - 1$  to 1 do
18:       $\mathbf{u}^s \leftarrow \text{ESTIMATEDEFORMATIONS}(F_{h^{2s}}(\mathbf{u}^s), G_{h^{2s}}(\mathbf{u}^s))$ 
19:       $\mathbf{u}^{s-1} \leftarrow \frac{1}{\rho} \uparrow \mathbf{u}^s$ 
20:    end for
21:     $\mathbf{u} \leftarrow \mathbf{u}^1$ 
22:  end while
23:
24:  // Step S3: temporal up-sample
25:  if not converged and motion too fast then
26:     $\mathbf{h}, \mathbf{u} \leftarrow \text{TEMPORALUPSAMPLE}(\mathbf{h}, \mathbf{u})$ 
27:  end if
28:
29:  return  $\mathbf{h}$  and  $\mathbf{u}$ 
30: end procedure
  
```

the quality of the reconstruction of the density volumes. Given the updated volumes, we re-estimate the deformation fields (Step S2). Finally, new intermediate key frames may be introduced (Step S3) in order to improve the temporal resolution and the spatial accuracy. The estimation of the deformation fields is done according to a multi-scale coarse-to-fine scheme [85]. The deformation fields are first estimated for the coarsest level ($s = N_{scales}$). Then, these fields are up-sampled scale-by-scale using the operator \uparrow and re-estimated at each given scale. The output of this step are the estimated deformation fields at the finest scale ($s = 1$). The up-sampling \uparrow and the down-sampling \downarrow are done by a factor of ρ , using a cubic interpolation. The steps (S1, S2 and S3) are repeated in a loop until convergence. For the last iteration, only the steps S1 and S3 are performed in order to get the best reconstruction at the last temporal sampling. The operators $F_{\mathbf{h}}^2, G_{\mathbf{h}}^2, F_{\mathbf{u}}^2, G_{\mathbf{u}}^2, F_{\mathbf{u}}^{\mathbf{W}}$ and $G_{\mathbf{u}}^{\mathbf{W}}$ will be explained below.

Step S0: Initialization

In the initialization of the algorithm, we set the deformation fields $\mathbf{u}_k = 0$. As a result, the warping operators simplify to identity operators, which means that in the first iteration the volume reconstruction (Step S1 below) is static volume reconstruction without the warp-and-project approach.

We also initialize the number of key frames N_k to be small, giving a coarse temporal resolution, and causing the initial volume estimates to be reconstructed from a large number of projections.

Step S1: Warp-and-Project based volume update

To solve the joint optimization problem in the Equation 5.6, we split it into two sub-problems that we solve alternately. The density volumes reconstruction sub-problem is solved during this current step of our framework. It is described by the following

minimization problem:

$$\begin{aligned}
\mathbf{h}^* = \operatorname{argmin}_{\mathbf{h}} & \sum_{j=1}^{N_p} \left\| \mathbf{A}_j \mathbf{W}_j^f(\mathbf{h}_{j^-}) - \mathbf{p}_j \right\|_2^2 + \sum_{j=1}^{N_p} \left\| \mathbf{A}_j \mathbf{W}_j^b(\mathbf{h}_{j^+}) - \mathbf{p}_j \right\|_2^2 \\
& + w_1 \sum_{k=1}^{N_k-1} \|\nabla_T \mathbf{h}_k + \nabla_S \mathbf{h}_k \cdot \mathbf{u}_k\|_1 \\
& + \sum_{k=1}^{N_k} [w_2 \|\nabla_S \mathbf{h}_k\|_{H_\epsilon} + w_3 \|\nabla_T \mathbf{h}_k\|_2^2]
\end{aligned} \tag{5.7}$$

With the new data fitting term used in this objective function, the comparison of the captured \mathbf{p}_k and the simulated projection of the forward-warped key frame, $\mathbf{A}_k \mathbf{W}_k^f \mathbf{h}_k$ is performed between pairs of projections corresponding to the same time t_i , and hence represent the same geometric configuration of the scan target. The same holds for the backward-warped key frame in the second data term.

In the following, we will drop the f and b superscript, and simply refer to the warping operator as \mathbf{W} . To simplify the notation, we will further use the non-linear \mathbf{W} like a matrix, and refer to its adjoint operator as \mathbf{W}^T .

To solve the optimization problem in Equation (5.7), we use a first-order primal-dual framework [26] as shown in Algorithm 7. This optimization problem is then split into two sub-problems, that we solve alternately using proximal operators. The first sub-problem contains only the least squares data fitting terms of this equation, that we denote as $F_{\mathbf{u}}^{\mathbf{W}}$. The remaining terms denoted as $G_{\mathbf{u}}^{\mathbf{W}}$, are grouped together into the second sub-problem.

The Algorithm 7 shows the pseudo-code used to solve the Warp-and-Project strategy, which is based on the CP-algorithm. The notations \mathbf{g} , $\bar{\mathbf{h}}$, $\mathbf{prox}_{\lambda_1 G_{\mathbf{u}}^{\mathbf{W}*}}$ and $\mathbf{prox}_{\mu_1 F_{\mathbf{u}}^{\mathbf{W}}}$ are used respectively for the slack and dual variables, and the proximal

Algorithm 7 CP-based method for tomographic reconstruction

```

1: procedure WARPANDPROJECT( $F_{\mathbf{u}}^{\mathbf{W}}$ ,  $G_{\mathbf{u}}^{\mathbf{W}}$ )
2:   while not converged do
3:     // update slack variable
4:      $\mathbf{g}^{j+1} \leftarrow \text{prox}_{\lambda_1 G_{\mathbf{u}}^{\mathbf{W}*}}(\mathbf{g}^j + \lambda_1 w_2 \bar{\mathbf{h}}^j)$ 
5:     // update volume variable
6:      $\mathbf{h}^{j+1} \leftarrow \text{prox}_{\mu_1 F_{\mathbf{u}}^{\mathbf{W}}}(\mathbf{h}^j - \mu_1 w_2 \mathbf{g}^{j+1})$ 
7:     // update dual variable
8:      $\bar{\mathbf{h}}^{j+1} \leftarrow 2 \cdot \mathbf{h}^{j+1} - \mathbf{h}^j$ 
9:   end while
10:  return  $\mathbf{h}$ 
11: end procedure

```

operators of the functions $G_{\mathbf{u}}^{\mathbf{W}*}$ and $F_{\mathbf{u}}^{\mathbf{W}}$. The operator w_2 is defined by:

$$w_2 = \begin{pmatrix} \nabla_S, \nabla_T, \mathbf{W} \end{pmatrix}^T \quad (5.8)$$

Since we are using the same priors for the density volumes as [24], the proximal operator $\text{prox}_{\lambda_1 G_{\mathbf{u}}^{\mathbf{W}*}}$ is also the same, and we refer to their paper for the details.

On the other hand, we proposed a new derivation to solve the proximal operator $\text{prox}_{\mu_1 F_{\mathbf{u}}^{\mathbf{W}}}$, given by:

$$\text{prox}_{\mu_1 F_{\mathbf{u}}^{\mathbf{W}}}(\mathbf{h}_v) = \underset{\mathbf{h}}{\operatorname{argmin}} \|\mathbf{A}\mathbf{W}\mathbf{h} - \mathbf{p}\|_2^2 + \frac{1}{2\mu_1} \|\mathbf{h} - \mathbf{h}_v\|_2^2. \quad (5.9)$$

This proximal operator problem is equivalent to solving the following minimization problem:

$$\begin{aligned} & \underset{\widehat{\mathbf{X}}}{\operatorname{argmin}} \|\widehat{\mathbf{X}}\|_2^2 \\ \text{subject to: } & \widehat{\mathbf{A}} \cdot \widehat{\mathbf{X}} = \widehat{\mathbf{p}}. \end{aligned} \quad (5.10)$$

where:

$$\begin{aligned}\hat{\mathbf{p}} &= \sqrt{2\mu_1} (\mathbf{p} - \mathbf{AW} \cdot \mathbf{h}_v) \\ \hat{\mathbf{A}} &= \begin{bmatrix} \mathbb{I} & \sqrt{2\mu_1} \mathbf{AW} \end{bmatrix} \\ \hat{\mathbf{X}} &= \begin{bmatrix} \sqrt{2\mu_1} (\mathbf{p} - \mathbf{AWh}) & \mathbf{h} - \mathbf{h}_v \end{bmatrix}^T\end{aligned}\quad (5.11)$$

In the pseudo-code detailed in the Algorithm 8, we present the solver that we used for the proximal operator in Equation 5.9. For each key frame k , we warp the density volume \mathbf{h}_k to the times t_j of its neighboring projections. Then, we project the obtained volume $\tilde{\mathbf{f}}_j$ using the corresponding viewing angle θ_j , and we compute the residual image with respect to the captured projection \mathbf{p}_j . A correction volume $\Delta\tilde{\mathbf{f}}_j$ is then computed by back-projecting the residual image Δp . Finally, this correction volume is warped from the time t_j of the j^{th} projection to the time T_k and used to update the key frame \mathbf{h}_k .

One can notice that the proximal operator in Equation 5.9 is very similar to the case of a linear least squares problem [27] where the solution is given by:

$$\text{prox}_{\mu_1 F_{\mathbf{u}}^{\mathbf{W}}}(\mathbf{h}_v) = \left(2\mu_1 (\mathbf{AW})^T \mathbf{AW} + \mathbb{I} \right)^{-1} \left(2\mu_1 (\mathbf{AW})^T \mathbf{p} + \mathbf{h}_v \right) \quad (5.12)$$

Actually, our proposed solver is also quite similar to this solution, except that the warping operator \mathbf{W} and its adjoint are non-linear operators that are implemented procedurally.

Deformation field estimation (S2)

After reconstructing volume estimates with the warp-and-project strategy, the deformation fields between successive key frames have to be updated. For this purpose, we solve the deformation field estimation sub-problem derived from the Equation 5.6.

Algorithm 8 Solver for the proximal operator in Equation 5.9

```

1: procedure WARPING-CORRECTION( $\mathbf{u}, \mathbf{h}_v$ )
2:   initialize:  $\alpha \in \mathbb{R}$ ,  $\mathbf{h} = \mathbf{h}_v$  and  $\mathbf{q} = \mathbf{h} - \mathbf{h}_v = 0$ 
3:   for  $k$  from 1 to  $N_k$  do
4:     while not converged do
5:       for all  $j$  with  $T_{k-1} \leq t_j \leq T_{k+1}$  do
6:         // warp the key frame to a given time (Eqs. 5.2, 5.3)
7:          $\tilde{\mathbf{f}}_j = \mathbf{W}\mathbf{h}_k$ 
8:         // residual in image space
9:          $\hat{\mathbf{p}}_j = \sqrt{2\mu_1} (\mathbf{p}_j - \mathbf{A}\tilde{\mathbf{f}}_j)$ 
10:         $\hat{\mathbf{X}}_j = \begin{bmatrix} \sqrt{2\mu_1} (\mathbf{p}_j - \mathbf{A}\tilde{\mathbf{f}}_j) & \mathbf{q}_k \end{bmatrix}^T$ 
11:         $\Delta p_j = \hat{\mathbf{p}}_j - \hat{\mathbf{A}} \cdot \hat{\mathbf{X}}_j$ 
12:        // compute the correction volume
13:         $\Delta \tilde{\mathbf{f}}_j = \hat{\mathbf{A}}^T \cdot \Delta p_j$ 
14:        // warp the residual to key frame and update
15:         $\mathbf{h}_k = \mathbf{h}_k + \alpha \mathbf{W}^T \Delta \tilde{\mathbf{f}}_j$ 
16:         $\mathbf{q}_k = \mathbf{q}_k + \alpha \Delta p_j$ 
17:      end for
18:    end while
19:  end for
20:  return  $\mathbf{h}$ 
21: end procedure
  
```

This optimization sub-problem is described as follows:

$$\begin{aligned} \mathbf{u}^* &= \underset{\mathbf{u}}{\operatorname{argmin}} w_1 \sum_{k=1}^{N_k-1} \|\nabla_T \mathbf{h}_k + \nabla_S \mathbf{h}_k \cdot \mathbf{u}_k\|_1 \\ &+ \sum_{k=1}^{N_k-1} \sum_{i=x,y,z} [w_4 \|\nabla_S \mathbf{u}_{k,i}\|_{H_\tau} + w_5 \|\nabla_T \mathbf{u}_{k,i}\|_2^2] \end{aligned} \quad (5.13)$$

The adopted approach to solve this optimization is the same as in the ST-tomography framework [24].

Temporal up-sampling (S3)

In the state-of-the-art 4D CT reconstruction methods, the temporal sampling is dependent on the number of captured projections N_θ used to reconstruct the density volume for one key frame. However, one of the advantages of the proposed warp-and-project strategy is to decouple the key frames from their association to these sets of N_θ captured projections. This allows us to adjust the temporal sampling by choosing the number of key frames, without taking into account the number of captured projections N_θ needed to initialize the reconstruction of each key frame. Then, we can on one hand increase the temporal resolution of our reconstruction. On the other hand, we will increase the accuracy of our reconstruction as we will show in the experiments section. In addition, the combination between the warp-and-project strategy and the temporal up-sampling opens the way to the reconstruction with a good accuracy of fast phenomena, for which the assumption of slow deformation is not valid.

The temporal up-sampling is performed only if a deformation field \mathbf{u}_k exceeds at a given threshold velocity. In this case, we introduce an intermediate key frame between the key frames \mathbf{h}_k and \mathbf{h}_{k+1} . The density volume for this new key frame $\mathbf{h}_{k'+1}$ is computed by warping \mathbf{h}_k to the time $T_{k'+1}$ using the deformation field \mathbf{u}_k . Moreover, the deformation field \mathbf{u}_k , is split linearly into two parts, each covering half the motion.

For the last outer iteration of our framework (see Algorithm 6), the temporal up-sampling is skipped. This is to ensure the final density volumes reconstruction result has been updated by the Warp-and-Project strategy. The obtained temporal sampling for the key frames is mostly non-uniform. But, with this sampling choice, we reduce the memory cost for saving the density volumes as well as the deformation fields.

5.3 Experiments

In the following, we first validate our proposed approach by quantitative comparisons on both simulated data (fluid flow) as well as real data where ground truth is available from high resolution scanning using a stop-motion approach (copper foam). Then we show the results of our reconstruction on six different data set, where the deformation is relatively fast. All these data sets use the low-discrepancy view sequence proposed in Chapter 4.

In the experiments shown in this section, many parameters are common. For the multi-scale optical flow calculation, we use $N_{scales} = 3$, $\sigma = 0.65$, and $\rho = 0.5$. The weights for the Huber penalty priors are set to $w_2 = 0.08$ and $w_4 = 1.2$. For volume density reconstruction, $w_1 = 0.25$, the relaxation parameter α for SART is set to 0.3, and two inner SART iterations are applied for all methods. For the proximal framework, $\lambda_1 = \mu_1 = 0.3$. Our algorithm was implemented in C++. The run times for density volume reconstruction and flow field estimation are given in the Table 5.2. Comparisons are made with two reference methods. The first is **SART-ROF**, a state-of-the-art static tomographic reconstruction method that incorporates the Rudin-Osher-Fatemi Total Variation prior [88]. The resulting optimization problem per frame is solved using a primal dual scheme is then chosen for this optimization problem, and the SART algorithm is used as the solver for the data term. The second reference method is the Space-Time Tomography **ST-Tomography** introduced in Chapter 4.

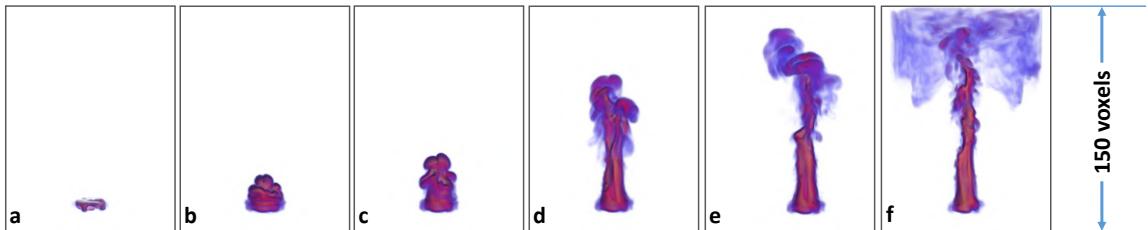


Figure 5.2: Several frames from the fastest fluid animation.

Synthetic plume data.

Our first validation experiment is done using synthetic fluid flow data, generated by the mantaflow code [89]. The purpose of this data set is to quantify the quality of the estimated deformation field, since simulation is the only way to generate dense ground truth motion. The quality of the volume reconstructions is assessed with real scans later in this section.

The simulation domain (resolution $100 \times 150 \times 100$) contains a cylindrical source emitting a density in a non-uniform way both spatially and temporally. The aim of this non-uniformity is to introduce some textures inside the plume density. Then, the emitted density is transported through the domain by a 3D incompressible flow over 300 time steps. For this experiment, the velocity of the density transport is controlled by the buoyancy parameter in the domain. We used 5 different values for this parameter ($v = 0.1, 0.41, 0.73, 1.28$ and 3.0 voxels/ Δt). Frames from the fastest animation are shown in Figure 5.2. The leftmost image shows the starting configuration, which is the same for all simulations.

Only the volumes having an odd index have been employed for the reconstruction of the plumes dynamic, but all 300 simulated volumes are used later for the comparison between the different obtained reconstructions and the ground truth. From each volume with an odd index a projection is computed with view points according to the low discrepancy view sampling strategy [24]. The reconstruction methods (ST-tomography and our method) are then performed using these 150 projections with

different strategies to obtain different number of key frames. For St-tomography, we combine 10 projections together to reconstruct each key frame. For our method, 5 levels of uniform temporal up-sampling are performed to obtain 15, 30, 60, 120 or 150 key frames. Where the level 1 is the level with the smallest number of reconstructed key frames.

Table 5.1: Numerical comparisons with ground truth data for different algorithms: average end-point error (EE) in voxels and average angular error (AE) in degrees.

	Speed (voxels/ Δt)	ST- tomo.	Ours				
			L1	L2	L3	L4	L5
EE	3.0	0.58	0.49	0.42	0.36	0.33	0.32
	1.28	0.44	0.37	0.32	0.27	0.25	0.21
	0.73	0.37	0.32	0.28	0.23	0.21	0.19
	0.41	0.31	0.27	0.24	0.20	0.18	0.17
	0.1	0.27	0.24	0.21	0.18	0.16	0.15
AE	3.0	17.5	17.1	16.2	14.6	13.9	13.8
	1.28	15.8	15.4	14.5	13.3	11.9	11.5
	0.73	14.3	14.1	13.3	12.2	11.3	10.9
	0.41	13.3	12.9	12.0	11.2	10.7	10.5
	0.1	12.7	12.0	11.2	10.7	10.3	10.2

Table 5.1 shows comparisons of the Average Endpoint Error (EE) and the Mean Angular Error (AE) for the competing methods. EE and AE are standard error metrics for assessing optical flow methods. Shown are ST-tomography and several levels of our method. We can see that our method dominates ST-tomography even after just one iteration, and then continues to improve over the next few iterations as the adaptive temporal sampling and the warp-and-project method improve the motion estimates. Furthermore, the method degrades gracefully with increased speed.

In-situ transformations of a copper foam.

To perform quantitative evaluation of the volume reconstructions, we turn to real X-ray images of copper foam. This dataset is inspired by composite material analysis in mechanical engineering, where the deformation characteristics of such metal foams

and similar kinds of materials are being investigated. Two controlled transformations (translation and compression) were conducted using a CT5000 5kN compression load stage (Deben UK Ltd., Suffolk, UK). This stage contains two parallel surfaces: a fixed one (top) and a moving one (bottom). The latter is controlled with high precision using a software interface. This setup allows us to capture high quality scans of many poses of the deformation using a stop-motion approach.

Vertical translation of the copper foam. For this experiment, a thin slice of copper foam (dimensions: $8.69 \times 8.42 \times 1.73 \text{ mm}^3$) was set on the moving part of the compression stage. The fixed part of the stage has been removed for this experiment. The sample was scanned for 42 different positions, using 90 view angles each time. Between successive position the sample was translated by 0.1 mm.

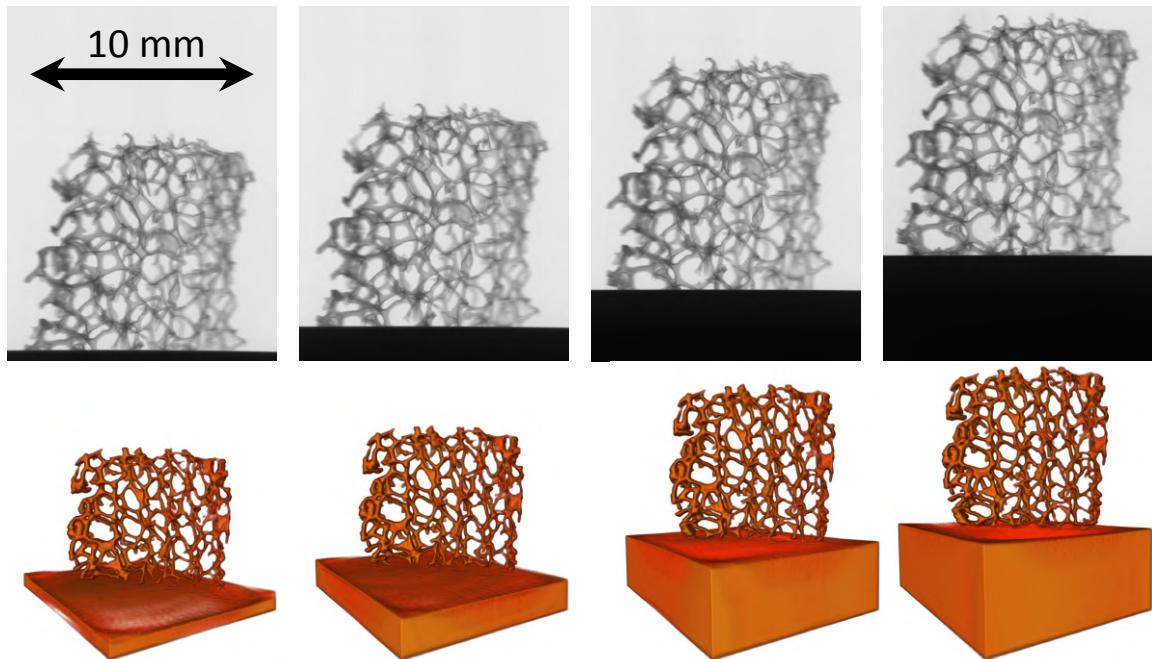


Figure 5.3: The translation of the copper foam. First row: captured projections, second row: reconstructed volumes at time frame 2, 12, 30, 40.

The Figure 5.3 illustrates 4 positions of the sample during the translation. While the first row shows projections of the sample at the same viewing angle, the second

row shows the obtained reconstruction of the copper foam using the SART-ROF algorithm. This reconstruction is very accurate, since the object is static for each position, and we use a large number of projections. For the following comparison of the different reconstruction methods these reconstructed volumes are considered as the ground truth volumes for the copper foam at each position.

For the dynamic reconstructions, from each position only 6 projections are used. Moreover, the projections from each 5 successive positions are combined in one time frame for the SART-ROF and the ST-tomography as well as for the initialization of our method. After convergence, the SART-ROF and the ST-tomography approaches reconstruct only 8 volumes, but with our method a volume is reconstructed for each position of the copper foam.

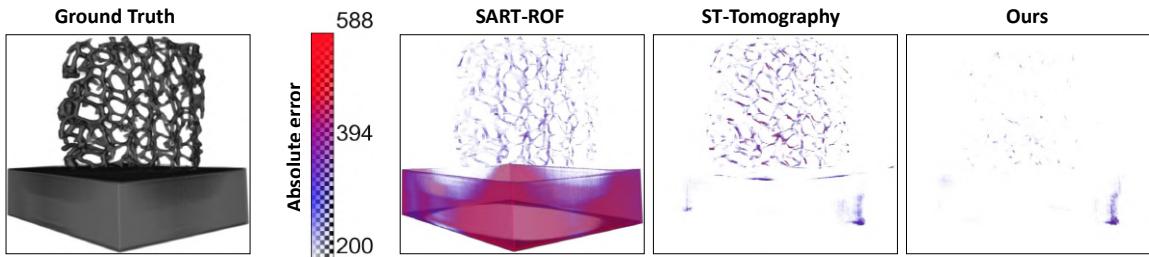


Figure 5.4: Absolute error for SART-ROF, ST-Tomography and ours for time frame 25.

For one intermediate volume the absolute error is given in Figure 5.4 between the ground truth and the different reconstruction methods. This figure shows the improved accuracy of our reconstruction compared to the state-of-the-art methods.

Compression of the copper foam. In this experiment, we scanned the foam crumpling under a compressive force using the setup shown in Figure 5.5. This is a real-world scenario that is of interest in mechanical engineering applications. In order to obtain ground-truth data, we again employed a stop-motion scanning strategy, this time acquiring 192 individual scans with 60 projections each. To test our method against the competing approaches we then selected 1 projection for each

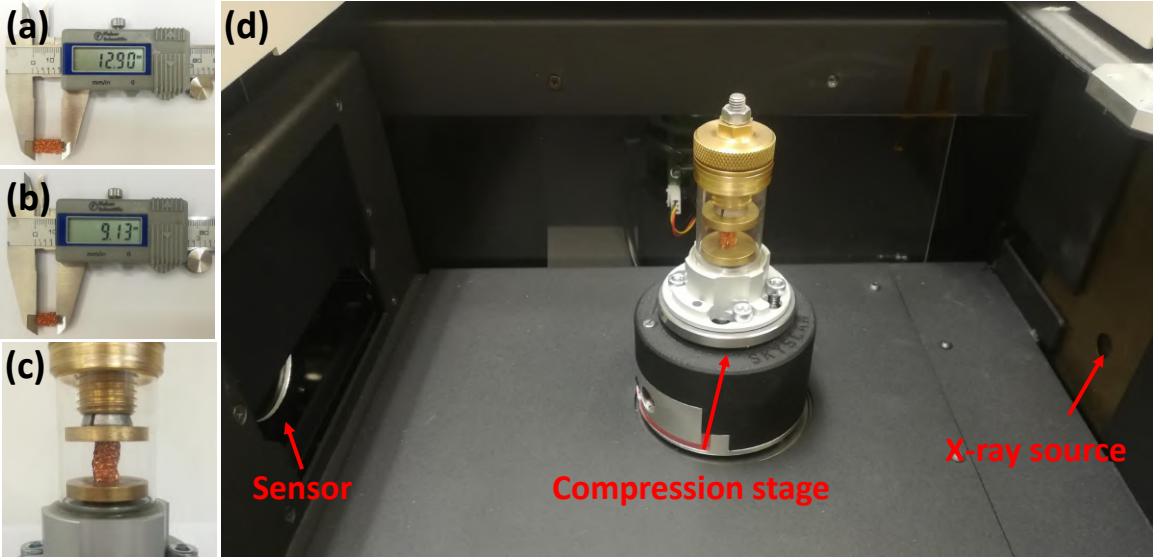


Figure 5.5: (a-b): The height (mm) of the copper foam before and after the compression process, respectively. The total displacement during the compression is 3.77 mm, uniformly distributed over 192 scans. 60 projections are obtained for each scan. (c): the states of the foam after 192 scans. (d) The Micro-CT setup for the in-situ compressions of the foam.

scan according to the low-discrepancy sequence. Results for a single frame are shown in Figure 5.6. The top row shows the reconstructions, while the bottom row shows the error of the different methods. We can see that our new method significantly reduces the reconstruction error compared to both SART-ROF and ST-tomography. Please also refer to the video.

To quantify this effect further, we show numerical results in Figure 5.7 and Table 5.3. Figure 5.7 plots the PSNR of each key frame for SART-ROF, ST-tomography, and several iterations of our method. As expected, the stationary reconstruction by SART-ROF has the worst performance, with ST-tomography generating a 2– dB gain on average. Our warp-and-project method is slightly superior to ST-tomography even in the first, and then continues to improve by 2–3 dB over the next 4 iterations. This is because the adaptive insertion of key frames allows for more accurate estimations of the motion field, which in term allows for better reconstructions of the volume densities with the warp-and-project approach. Table 5.3 shows numerical results of

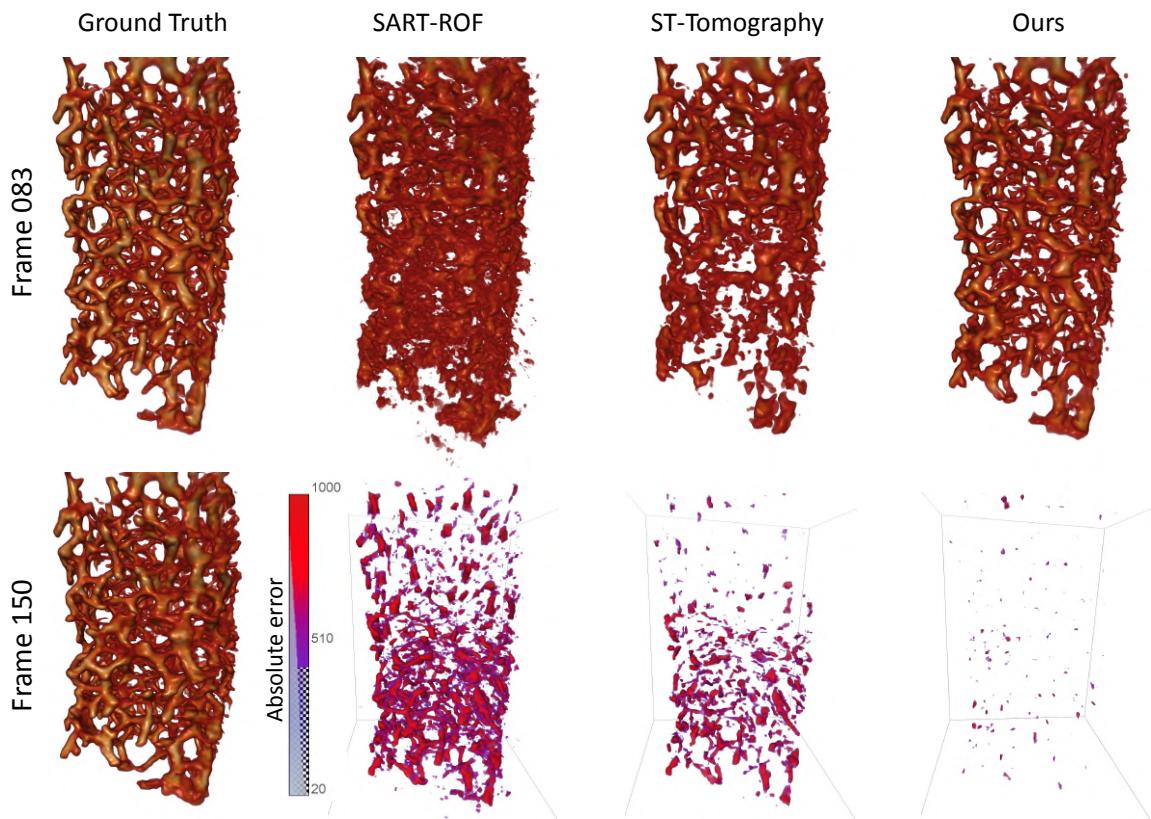


Figure 5.6: Algorithm comparison for the compression of copper foam. First row: results from different reconstruction methods compared to ground truth, second row: the absolute error for time frame 150.

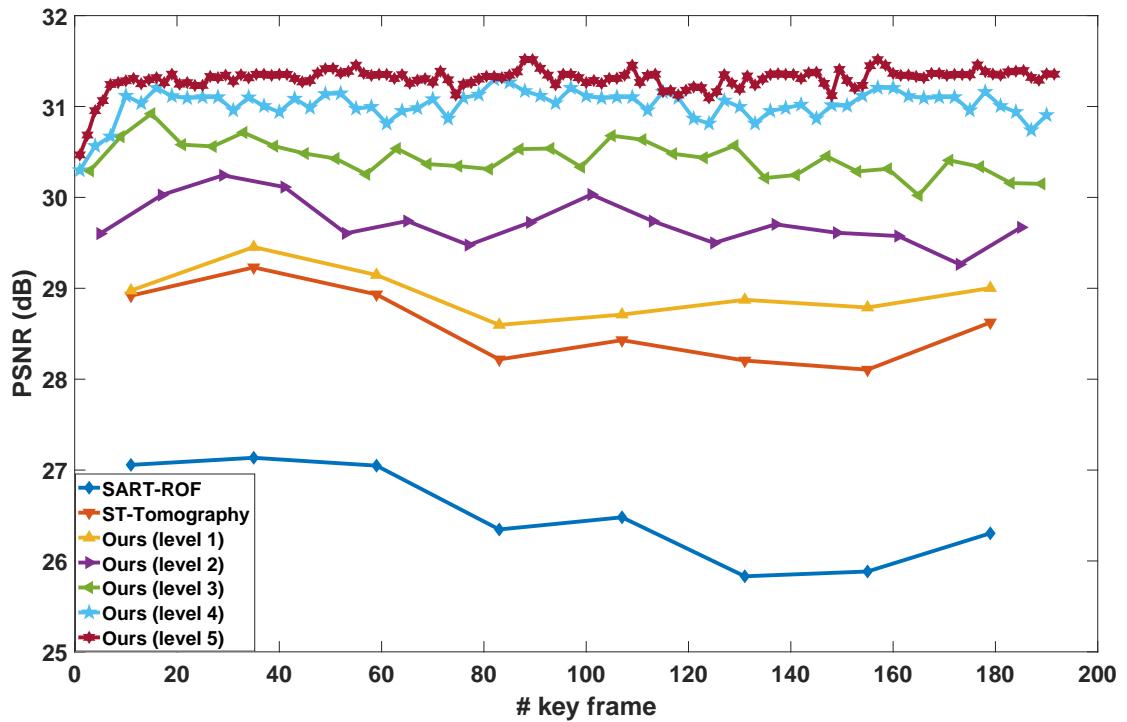


Figure 5.7: PSNR values of the volume reconstructions for each key frame. We compare SART-ROF, ST-tomography, and several iterations of our method. Please see text for details.

the same experiment, aggregated over all frames, but with separate statistics for the top, middle, and bottom parts of the volume. These exhibit different motion speeds since the foam is only compressed from the bottom, while the top edge is stationary. To test multiple virtual scan speeds, we also performed reconstructions in which every second time step is skipped.

Table 5.2: Parameters used in the acquisition and for the tomographic reconstruction.

Dataset	# of proj.	Cap.time [h:mm]	N_θ	Keyframes (Ours)	Vol. size	S1 step [h:mm]	S2 step [h:mm]	# iter.	w_3	w_5
rock	800	1:22	20	80	$400 \times 500 \times 400$	4:42	6:30	5	0.3	0.05
fungus	600	0:38	30	128	$250 \times 300 \times 250$	2:20	4:36	4	0.2	0.03
hydro-gel	640	0:43	16	70	$423 \times 320 \times 423$	3:15	5:20	5	0.3	0.05
liquids	600	0:46	30	120	$180 \times 200 \times 180$	3:44	5:50	4	0.1	0.03
pills	30	0:03	3	30	$400 \times 200 \times 400$	1:40	2:45	5	0.1	0.03

Table 5.3: Calculated PSNR [dB], and SSIM for different reconstruction methods: SART-ROF, ST-Tomography, and our method.

Region	Top (slowest)		Center (medium)		Bottom (fastest)	
	Sampling 1: 192 projections					
Metric	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
SART-ROF	29.14	0.68	26.27	0.56	24.11	0.28
ST-Tomo.	32.11	0.72	27.85	0.64	24.68	0.41
Ours (level 1)	32.28	0.72	28.48	0.65	24.90	0.46
Ours (level 2)	32.72	0.75	29.10	0.67	26.25	0.53
Ours (level 3)	33.32	0.77	29.93	0.68	28.08	0.56
Ours (level 4)	33.74	0.78	30.57	0.70	29.18	0.61
Ours (level 5)	33.78	0.78	30.87	0.72	29.35	0.62
Sampling 2: 96 projections						
SART-ROF	28.55	0.65	23.61	0.24	21.31	0.19
ST-Tomo.	31.41	0.69	24.76	0.47	23.14	0.36
Ours (level 1)	31.51	0.69	25.14	0.50	23.39	0.37
Ours (level 2)	31.86	0.71	26.29	0.56	23.95	0.39
Ours (level 3)	32.43	0.72	27.84	0.60	24.45	0.43
Ours (level 4)	32.59	0.72	28.88	0.63	24.72	0.44
Ours (level 5)	32.88	0.73	29.08	0.64	24.87	0.46

We also provide qualitative results on a number of fast-moving data sets. The algorithm parameters for the individual dataset are shown in Table 5.2 along with the reconstruction times. In addition to the visual results presented here, we also

encourage the reader to consult the video for the full time-varying reconstruction results.

Rising dough. The first data set is a scan of fast rising dough, shown in Figures 5.10, 5.9, and 5.8. As the dough rises due to the yeast secreting CO₂ gas, we see bubbles forming at both microscopic scales and mesoscopic scales. The microscopic bubbles manifest themselves as a change in the density of the dough, while the meso-scale bubbles result in drastic topology changes of the dough that are resolved well with our method, but are blurred out in the comparison method (see slices in Figure 5.9).

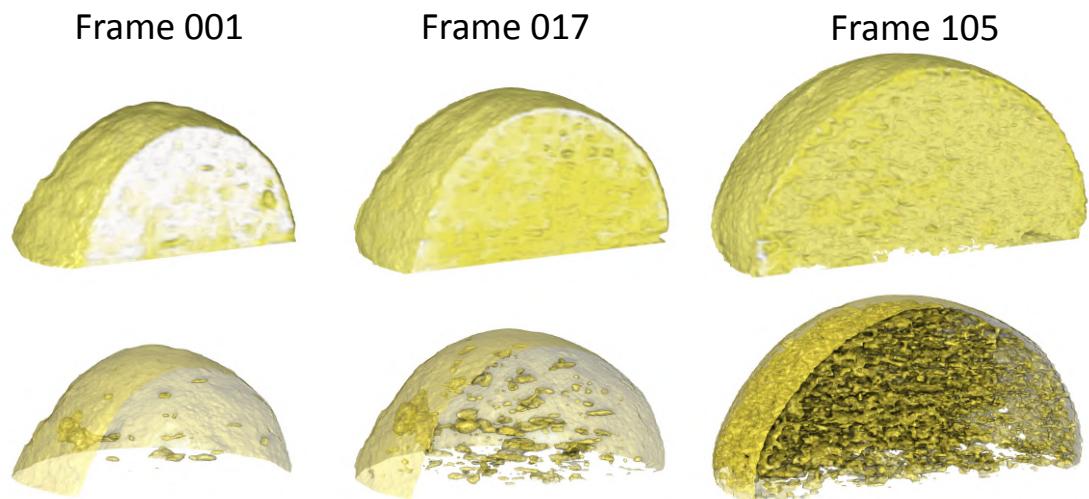


Figure 5.8: Rising dough reconstructed by our method. First row: direct volume rendering. Second row: bubble surfaces.

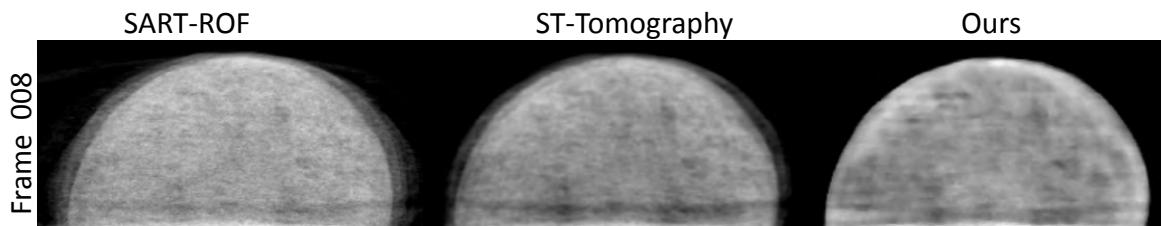


Figure 5.9: Comparison between SART-ROF, ST-tomography and Ours for a slice visualization of the dough dataset.

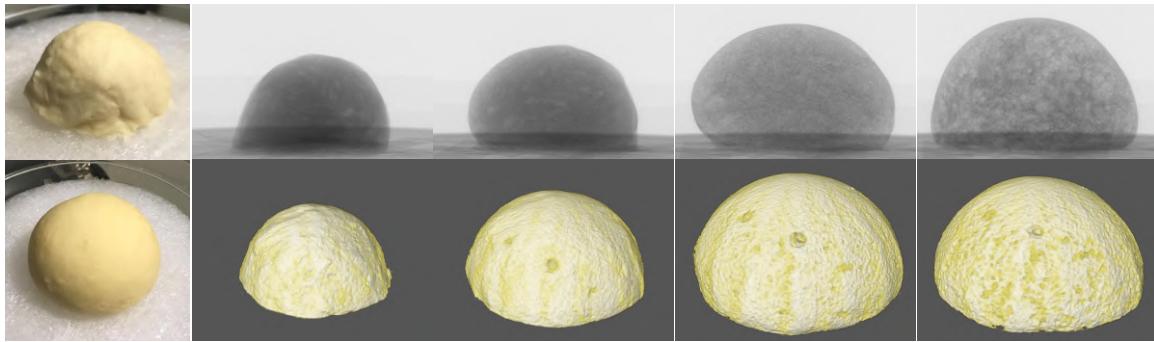


Figure 5.10: 3D reconstructions of the dough dataset, with corresponding 2D X-ray images, as well as before-and-after photographs.

Capillarity effect in porous rock. The next experiment is a standard test deployed in geology, and specifically oil and gas exploration [47]. The goal of such studies is to measure the porosity or the permeability of rock samples in order to quantify the ability of the rock to store oil. Here, a cylindrical sample of the rock is placed in a dish with bottom dipped into a liquid contrast agent. Due to capillary action, the liquid is absorbed into the sample, over the duration of the scan (one hour and 22 minutes). A 3D visualization of this experiment showing the absorbed liquid can be found in Figure 5.11 and Figure 5.12.

Dried snow fungus. Next, we show the re-hydration of dried mushrooms (*tremella fuciformis*, Figure 5.13). A direct comparison with the reference methods shows significantly improved detail and reconstruction quality compared to both SART-ROF and ST-tomography. Note that the first iteration of our algorithm produces approximately similar results to ST-tomography, but later iterations improve the result due to both the added key frames and the warp-and-project method using the estimated motion fields.

High viscosity liquids. In Figure 5.14 we show an experiment where solid objects were dropped into a high viscosity fluid and sink to the bottom under gravity. In this case the motion is fast enough to violate the assumption of negligible deformation

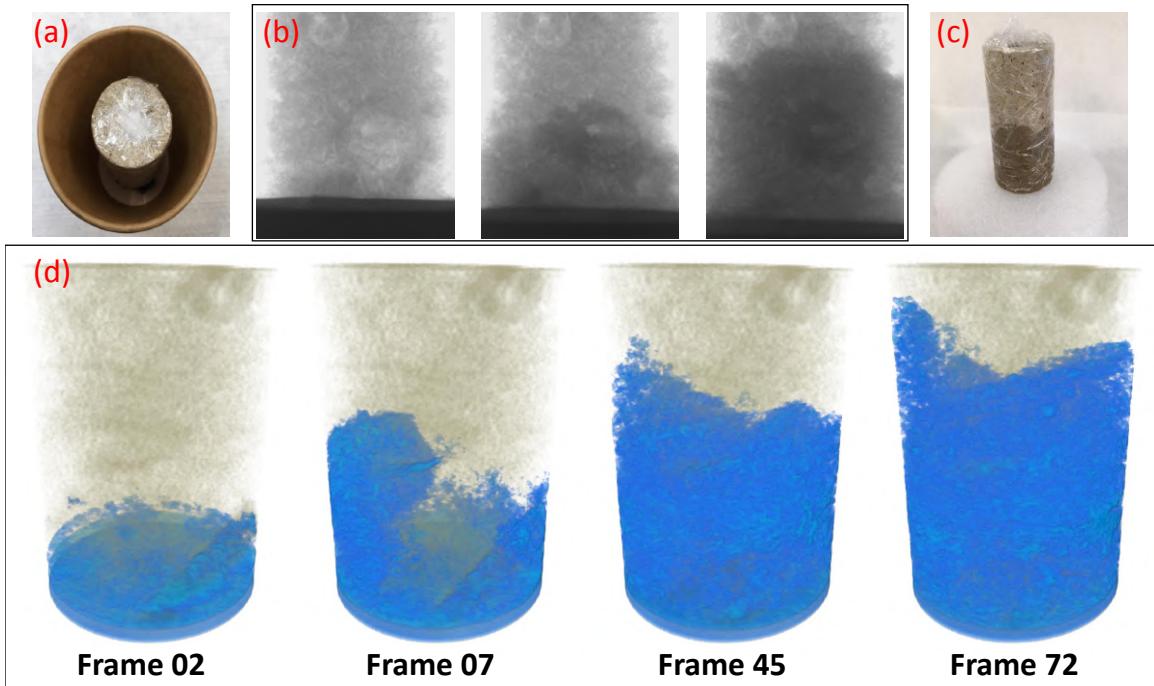


Figure 5.11: Reconstruction results for the porous rock dataset. Images (a) and (c) represent respectively the rock before and after the scan. Some representative CT projections are given in (b). The rendering results in (d) show the absorption of liquid over time.

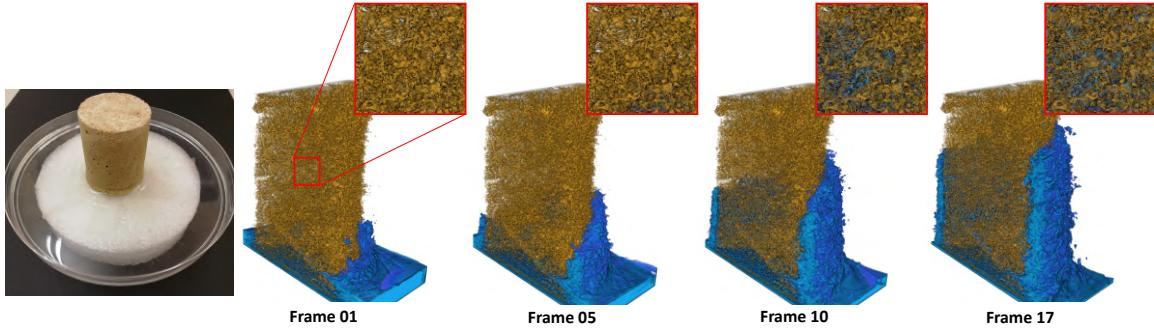


Figure 5.12: Rendering results for the rock from our method.

between successive frames, which ST-tomography is based on. As a result, the shapes are blurred out and fine detail is lost, while our method manages sharp reconstructions of the solid.

Pills dissolved in water. In many medical pills, the drug is coated with a protective hull that dissolves in water, releasing the drug at a designed rate. Figure 5.15

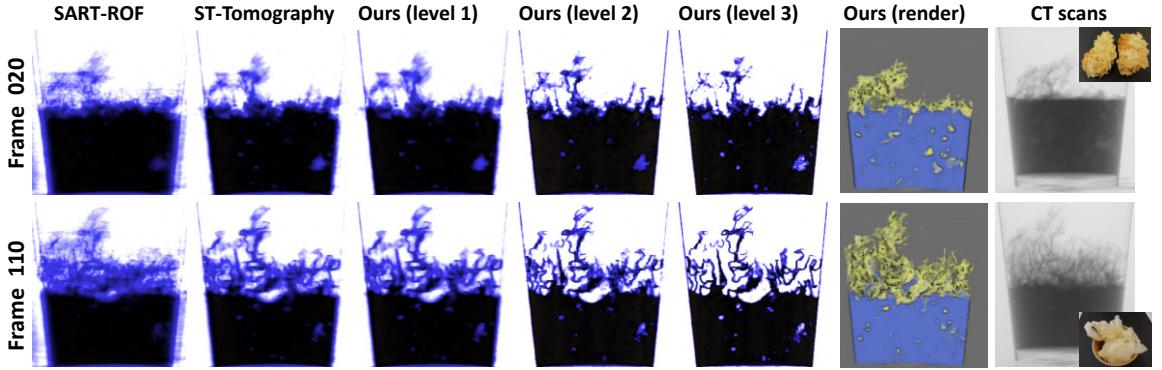


Figure 5.13: Re-hydration of dried *tremella fuciformis*. This process results in a rapid volume expansion, which is not well captured by comparison methods, while our method manages to reconstruct fine detail. From left to right: slice visualizations, 3D renderings, acquired projections during scanning and photographs of the sample.

shows the results of capturing this process in a CT scan. This is a challenging data set, since the motion is initially quite fast, and the drug particles are quite small. The whole sequence is only 3 minutes long and consists of 30 projections. As with the previous datasets, we can see that our method produces crisper, more detailed results than the comparison methods.

Hydro-gel balls. Finally, in Figures 5.16 and 5.17 we show 2D slice visualizations and 3D renderings of Hydro-gel balls (Orbeez) absorbing water. Since the density of the water and the Hydro-gel balls is very similar, there is very little contrast between the two in the X-ray images or in the reconstruction. However, we can again see that the rapid absorption of water and the associated volume change create fast motions that cannot reliably be reconstructed by the competitor methods, whereas our approach produces sharp, clearly defined shapes for the balls.

5.4 Conclusion

In summary, we have presented a new method, warp-and-project tomography, for tomographic reconstruction of deforming objects. We perform quantitative comparisons on simulated data, as well as qualitative comparisons on real data from a number of

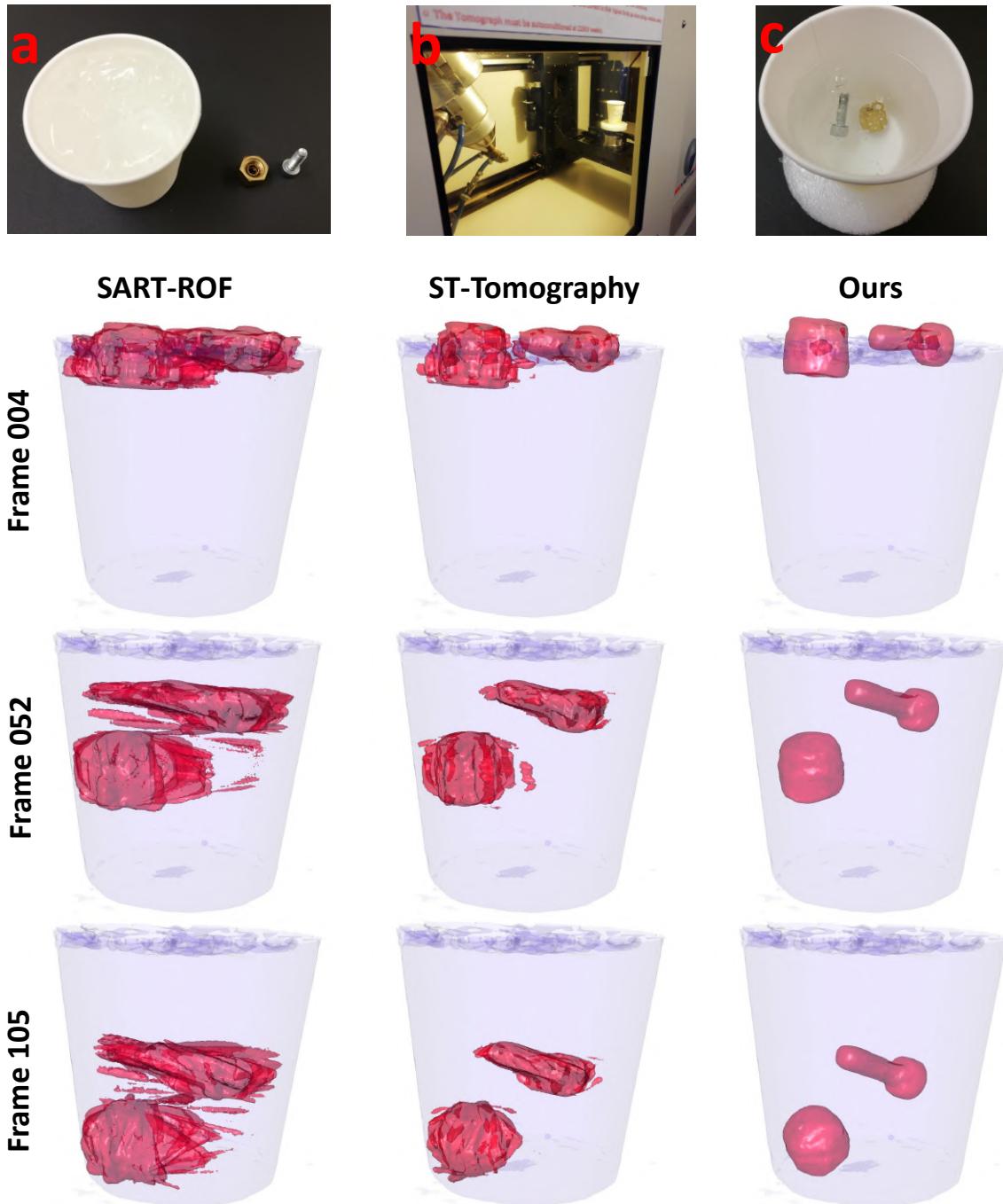


Figure 5.14: Metal objects being dropped into a high viscosity liquid. Our method produces the sharpest reconstructions.

different application domains. These experiments clearly show a significant improvement in the reconstruction quality compared to the state-of-the-art in both static and dynamic tomographic reconstruction.

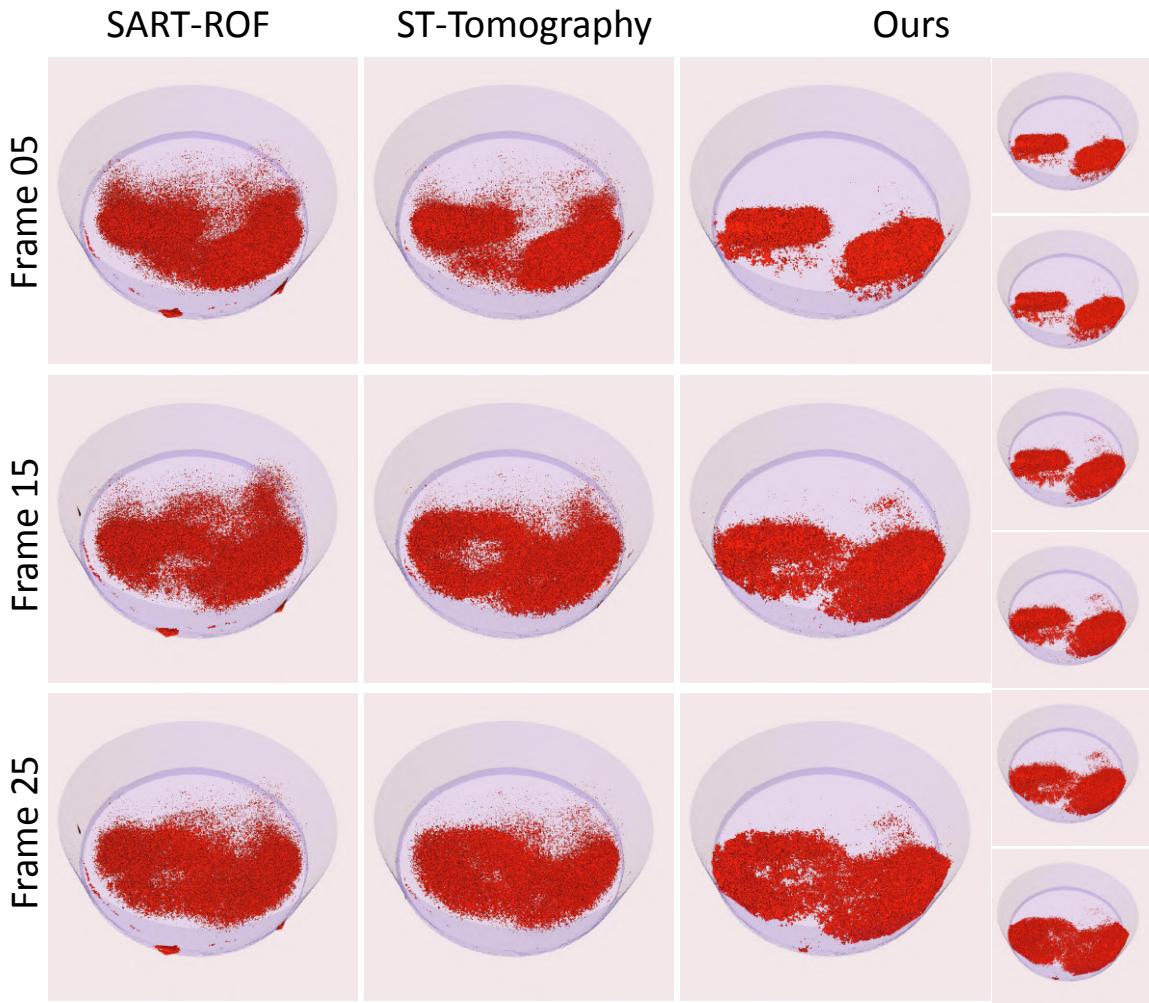


Figure 5.15: Pills dissolving in water. This is our fastest dataset with only 30 projections captured in 3 minutes.

We observed two major limitations of our method. First, the tracking algorithm requires features in order to work. If we are scanning a nearly feature-less volumetric object, the reconstruction of the motion field will only work on the boundary, but not inside the feature-less region. Second, our method will degrade with increasing speed of motion and an increasing geometric complexity of the volume. For example, objects that have a lot of high frequency details generally require more scans in a static reconstruction. For these objects our method can only work with slower motions. Also as a minor limitation, we do not consider motion blur during the acquisition of a single frame.

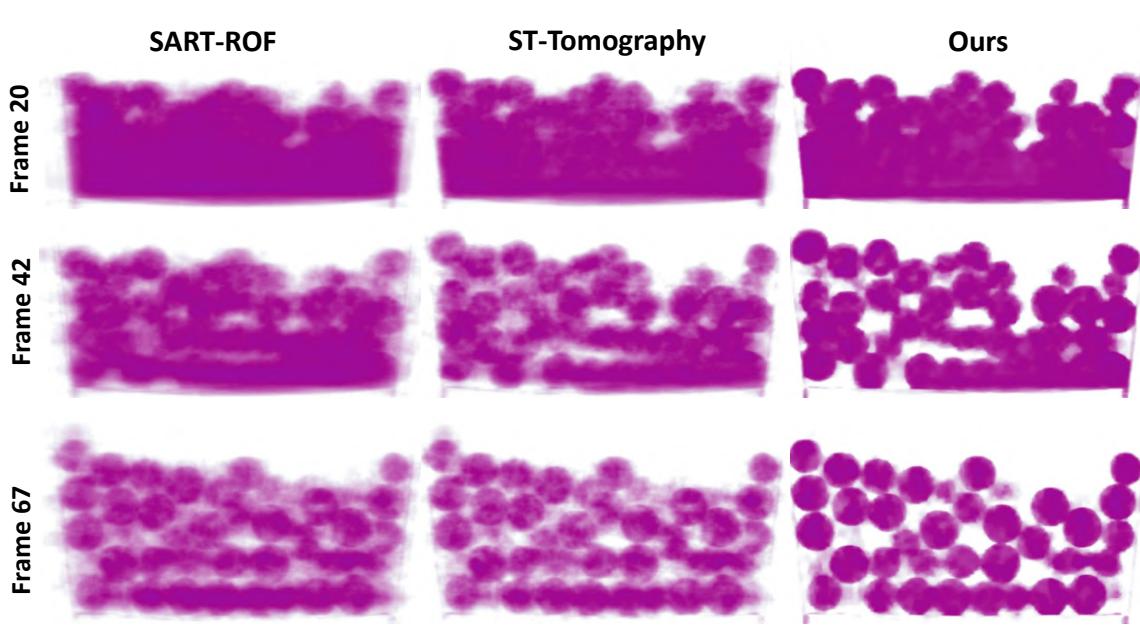


Figure 5.16: Slice rendering of water absorption by Hydro-gel balls.

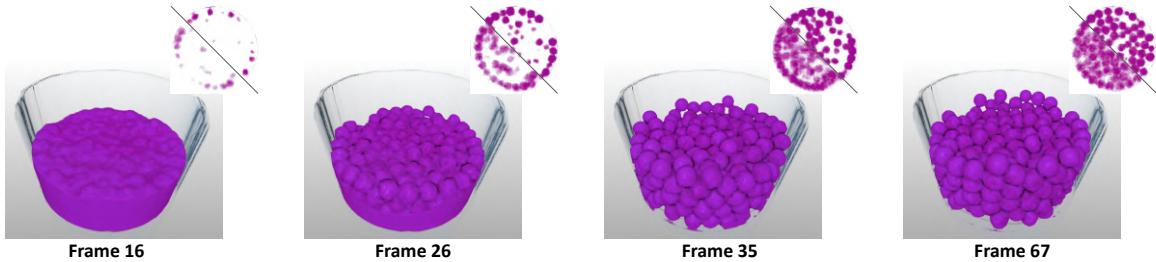


Figure 5.17: 3D rendering of water absorption by Hydro-gel balls.

We believe that the developed method has significant applications in many domains. In fact the metal foam and rock data sets are already starting point of applications in mechanical engineering and geology / oil and gas exploration. In the future we plan on deeper investigations of these applications in collaboration with domain experts. We also intend to combine the method described here with other X-ray tomography techniques such as phase contrast imaging, in order to boost the contrast in certain problematic data sets, such as the hydro-gel balls.

Chapter 6

Applications

In this chapter, some additional applications directly based on the tomographic work introduced in previous sections are given.

One interesting extension from X-ray tomography is the fluid imaging application, in which the images are captured from a few optical cameras (2-3 in our example) with a similar geometry setting to tomography. The soot and fuel data presented in this chapter are captured from the Clean Combustion Research Center (CCRC), while the mixing fluid is imaged by High-Speed Fluids Imaging Laboratory, both at KAUST.

We take the captured soot data as an example for fluid imaging. Three high-speed cameras have surrounded a burner at 0, 45, and 90 degrees, respectively. The cameras are well aligned so they match pixel to pixel and high-speed images of a turbulent flame then be captured. After imaging the soot field, the extinction measurements from the 3 angles are obtained. To validate the accuracy of reconstruction quality later, a 2D "slice" of soot volume fractions (SVF), located at 127 degrees, for every 180th frame is also provided. Figure 6.1 and Figure 6.2 are the input images (after post-processing) at some time step from the captured videos.

Considering the specific properties of fluid imaging, such as the flow is always incompressible, the reconstructed quality can be further improved by incorporating such natural properties into the optimization framework. We show how the incompressible flow could be interpreted as a prior and by leveraging it in our optimization framework, the unprecedented results can be obtained in this section.

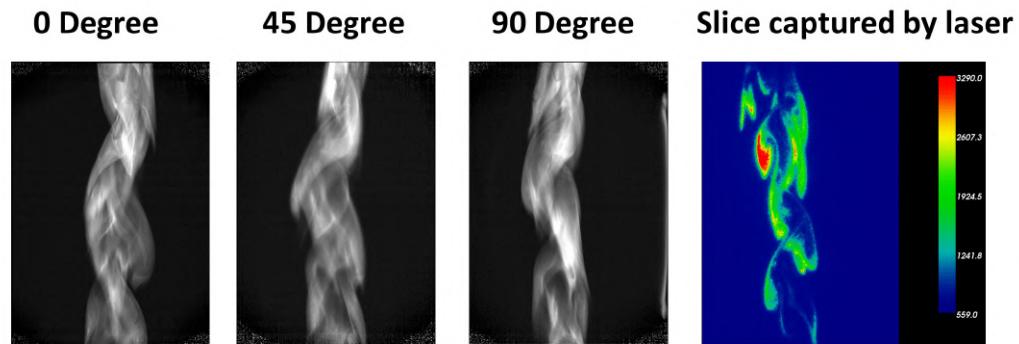


Figure 6.1: From left to right: captured projection for low density soot data at 0, 45, and 90 degree. Rightmost is obtained SVF slice.

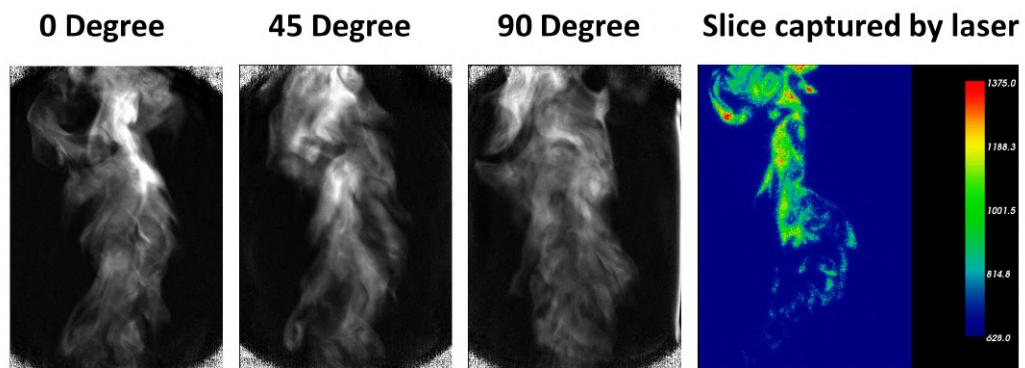


Figure 6.2: From left to right: captured projection for high density soot data at 0, 45, and 90 degree. Rightmost is obtained SVF slice.

6.1 Pipeline overview

From the video sequence obtained by the three cameras (Fig. 6.1 and 6.2), 3D reconstruction of the extinction field and 4D reconstruction of the velocity field was accomplished using a variant of the Space-Time tomography (ST-Tomography) algorithms shown in Chapter 4 and 5 . This 4D tomography framework is based on a variational approach that jointly estimates the density field (extinction field) and the velocity field. In this method, an objective function is defined that combines a data-fitting term with different spatial and temporal regularizers. The data-fitting term ensures a good fit of the reconstructed density field with the extinction measurement video. The first regularizer is a 3D version of the Horn-Schunck optical flow model [84], Next, the proposed divergence free prior is applied on the flow estimation of the soot. This prior combines both the density and velocity fields to ensure density consistency between successive frames. At the same time, it allows 3D tracking of density gradients and therefore provides a powerful tool to estimate the velocity field. In this chapter, we assume negligible soot formation or oxidation between successive frames since soot formation and oxidation is a process that typically takes several milliseconds whereas the frame rate was several orders of magnitude faster. A multi-scale scheme of the optical flow is applied, to deal with large deformations discussed in [24]. Furthermore, we implemented spatial smoothness penalties on both the extinction and velocity fields to retrieve a more plausible solution, especially in regions of uniform extinction where the optical flow priors are less efficient. Next, a temporal smoothness filter was added to the density field to improve the quality of the result.

The framework begins by initializing each time step using the simultaneous algebraic reconstruction technique (SART) algorithm to construct a 3D extinction field [7]. Following initialization, the framework estimates the flow field by calculating the frame-to-frame deformation in the extinction field. The framework then iterates between refinement of the optical density reconstruction and the frame-to-

frame flow field. This variational approach for estimating velocity fields was recently introduced in PIV measurements with encouraging results [83].

Divergence-free constraint. In fluid dynamics, the advection theory for a conserved quantity, which is described mathematically by continuity equation, presents the relations between scalar field \mathbf{f} , time step t , and flow velocity \mathbf{u} as:

$$f_t + \nabla \cdot (f\mathbf{u}) = 0$$

with a simple product rule operation, we have:

$$f_t + \nabla f \cdot \mathbf{u} + f \nabla \cdot \mathbf{u} = 0 \quad (6.1)$$

where $\nabla \cdot$ is the divergence operator. We call Equation 6.1 the energy-conserving (or mass-conserving) equation.

If the flow is incompressible (divergence-free), then we have

$$f_t + \mathbf{u} \cdot \nabla f = 0 \quad (6.2)$$

$$\text{w.r.t } \nabla \cdot \mathbf{u} = 0 \quad (6.3)$$

Interestingly, we note that Equation 6.2 is exactly the partial differential equation form of optical flow under the common brightness constancy constraint [84]. In our framework, we apply a 3D variant version of optical flow motion estimation, with additional sparse prior and pressure projection prior (i.e. divergence-free constraint). Details are provided in the next section.

Optimization framework. The optimization framework for dynamic fluid reconstruction is quite similar to the framework with [24, 25], except the flow motion

estimation part, in which physical-based divergence-free prior is applied:

$$\begin{aligned}
G_{\mathbf{h}}^3(\mathbf{u}^s) = & \sum_{t=1}^{N_k-1} \| (\text{warp}(\mathbf{f}_{t+1}, \uparrow \mathbf{u}_t^{s+1}) - \mathbf{f}_t) \\
& + \nabla_S \text{warp}(\mathbf{f}_{t+1}, \uparrow \mathbf{u}_t^{s+1}) \cdot (\mathbf{u}_t^s - \uparrow \mathbf{u}_t^{s+1}) \|_1 \\
& + l_1 \sum_{t=1}^{N_k-1} \sum_{i=x,y,z} \| \nabla_S \mathbf{u}_{t,i}^s \|_{H_\tau} + l_2 \text{DIV}(\mathbf{u}^s)
\end{aligned} \tag{6.4}$$

The optimization problem for each scale is then given as

$$\mathbf{u}^{s,*} = \underset{\mathbf{u}^s}{\operatorname{argmin}} G_{\mathbf{h}}^3(\mathbf{u}^s). \tag{6.5}$$

Due to the presence of the L1-norm and the Huber penalty in Equation 6.5 we reformulate this optimization in the first-order primal-dual framework (CP algorithm), introduced by Chambolle and Pock. The strategy proposed by this algorithm, consists in splitting the optimization problem into different sub-problems that are solved independently in the form of proximal operators. The pseudo-code shown in Algorithm 9, summarizes the CP algorithm used to solve the Equation 6.5, where \mathbf{w} and $\bar{\mathbf{u}}$ are respectively the slack and the dual variables. $\text{prox}_{\lambda_1 G_{\mathbf{h}}^*}$ and $\text{prox}_{\mu_1 F_{\mathbf{h}}^3}$ are the proximal operators, based on the functions :

$$\begin{aligned}
G_{\mathbf{h}}^3(\mathbf{u}^s) = & \sum_{t=1}^{N_k-1} \| (\text{warp}(\mathbf{f}_{t+1}, \uparrow \mathbf{u}_t^{s+1}) - \mathbf{f}_t) \\
& + \nabla_S \text{warp}(\mathbf{f}_{t+1}, \uparrow \mathbf{u}_t^{s+1}) \cdot (\mathbf{u}_t^s - \uparrow \mathbf{u}_t^{s+1}) \|_1 \\
& + l_1 \sum_{t=1}^{N_k-1} \sum_{i=x,y,z} \| \nabla_S \mathbf{u}_{t,i}^s \|_{H_\tau}
\end{aligned} \tag{6.6}$$

$$F_{\mathbf{h}}^3(\mathbf{u}^s) = \text{DIV}(\mathbf{u}^s) \tag{6.7}$$

Gregson et. al [14] presents how to cast pressure solver that is integral to many fluid simulators as a proximal operator. We utilize this concept and adapt the

Algorithm 9 Primal dual solver for velocity retrieving

```

1: procedure VELOCITYRETRIEVING( $F_h^3, G_h^3$ )
2:   while not converged do
3:     // update slack variable
4:      $\mathbf{w}^{j+1} \leftarrow \text{prox}_{\lambda_1 G_h^{3*}}(\mathbf{w}^j + \lambda_1 \mathbf{K}_h \bar{\mathbf{u}}^j)$ 
5:     // update deformation field
6:      $\mathbf{u}^{s,j+1} \leftarrow \text{prox}_{\mu_1 \text{DIV}}(\mathbf{u}^{s,j} - \mu_1 \mathbf{K}_h^T \mathbf{w}^{j+1})$ 
7:     // update dual variable
8:      $\bar{\mathbf{u}}^{j+1} \leftarrow 2 \cdot \mathbf{u}^{s,j+1} - \mathbf{u}^{s,j}$ 
9:   end while
10:  return  $\mathbf{u}^s$ 
11: end procedure

```

divergence-free term to our primal-dual optimization framework, written as:

$$\begin{aligned}
\text{prox}_{\mu_1 \text{DIV}}(\mathbf{x}^*) &= \underset{x \in \Phi \text{DIV}}{\text{argmin}} \|\mathbf{x} - \mathbf{x}^*\|_2^2 \\
&= \Pi_{\Phi \text{DIV}}(\mathbf{x}^*).
\end{aligned} \tag{6.8}$$

The adopted approach to solve line 4 in Algorithm 9, and scalar densities reconstruction are the same as in the ST-tomography framework [24].

6.2 4D Soot Imaging of a Combustion Process

Figure 6.3 and 6.4 shows the final frame of the reconstructed volume (the same frame as shown in Figure 6.1 and Figure 6.2 respectively). Image visualization for the reconstruction was done using Avizo. In the reconstruction, local extinction (K) values are scaled up by a factor of 100. Following image reconstruction, image noise related to the noise found in the corners of Figure 6.2 was removed by thresholding data outliers found outside of the soot zone. From the results, we can see clearly that the reconstruction quality of low-density soot is better than the higher one. Considering that only 3 views provided, the shown results are still encouraging.

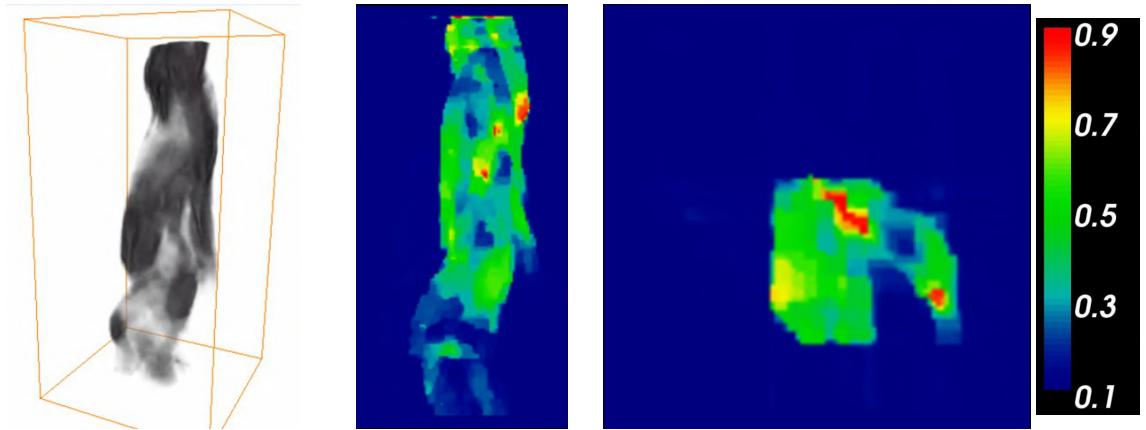


Figure 6.3: Reconstruction results for low density soot by proposed divergence-free based 4D tomography technique. Left: Reconstructed volume. Middle, front slice visualization. Right: Top view slice visualization.

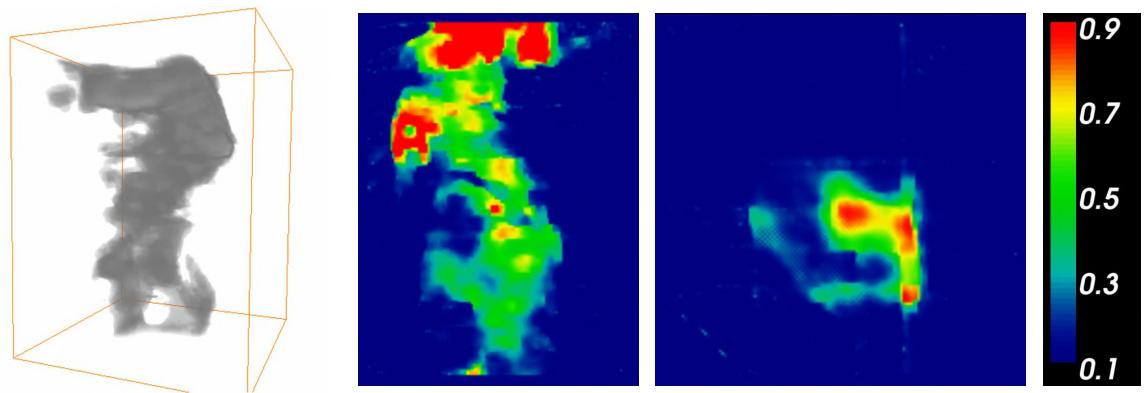


Figure 6.4: Reconstruction results for high density soot by proposed divergence-free based 4D tomography technique. Left: Reconstructed volume. Middle, front slice visualization. Right: Top view slice visualization.

6.3 4D Imaging of a Mixing Fluid Process

In this section, we focus on the 3D reconstruction of mixing fluid. As shown in Figure 6.5 (left), there are only two cameras, at 0 and 90 degrees respectively, in the set up of fluid capture, making it more challenge. Though it is a quite ill-posed problem, with our physically plausible reconstruction method as described in this Chapter. The results with significant improvement can already be obtained. Figure 6.5 (b) show the input projection images after post-processing, with 200 time steps in total. The reconstructed results is shown in Figure 6.5 (c).

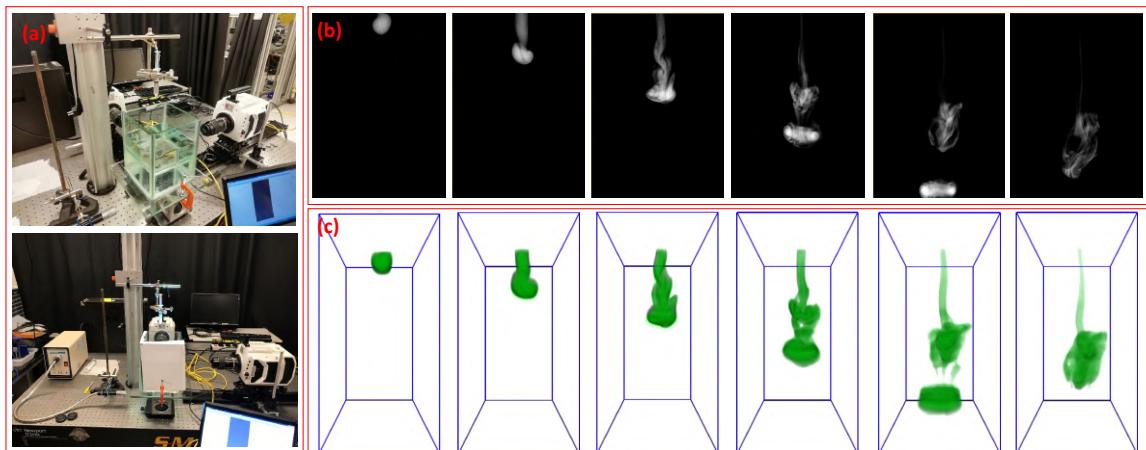


Figure 6.5: (a): the set up for the capture of high speed fluid imaging. (b): captured projection images from the camera at 90 degree.(c): Reconstruction result from proposed method. There are 200 time frames in total.

6.4 4D Imaging of a Fuel Injection Process

In this section, we show how our technique retrieves 4D volumetric fuel spray results with high quality. Fuel evaporation and fuel-air mixing in engine affect combustion efficiency and emission, which is definitely a promising research direction. As shown in Figure 6.6 and Figure 6.7 (a), with different conditions, the output projection images will be quite different. We assume that when a certain condition is fixed, every time we inject the fuel, the images we captured at each angle are similar to each other.

With this assumption, in each fixed condition, we do injections at five different angles (i.e., -60, -30, 0, 30 and 60 degrees, respectively). For validation, at each angle there are two injections will be obtained. The reconstructed results at one time step can be observed in Figure 6.8 and Figure 6.7 (b). Figure 6.6 shows five time steps for one condition from our reconstruction framework.

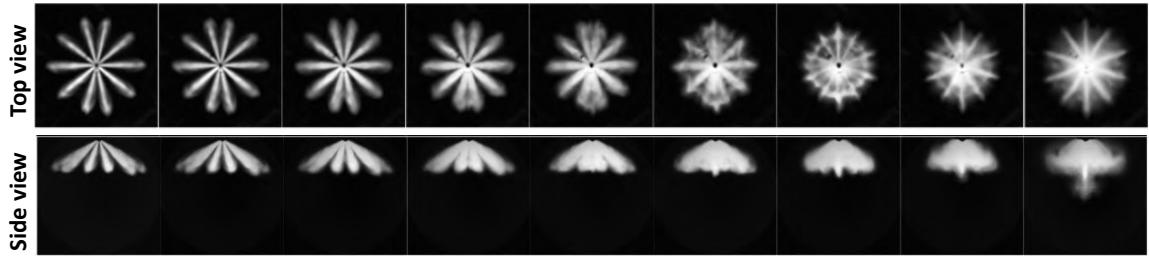


Figure 6.6: From left to right: captured spay data with different conditions. First row: top view. Second row: side view.

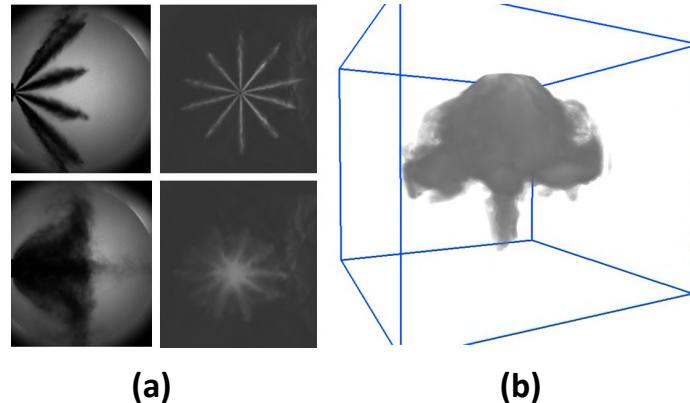


Figure 6.7: (a): two condition captured from the experiment. (b): the volumetric result from our reconstruction method.



Figure 6.8: visualization of reconstructed volumes at different time steps.

6.5 View Synthesis for Visible Light Tomography

Finally, we conducted a pilot study on the view synthesis (i.e., projections interpolation) for visible light tomography.

Observations. Assume parallel beam for simplicity, for visible light tomography, especially for the capture of the natural phenomenon by optical means, compared to traditional projection images, there are some fascinating features:

- The captured phenomenon is involved with (semi-) transparent material, no occlusion, with segment-free background, constant lighting condition with all captured videos at different sparse angles, which is similar to a tomographic setting.
- The predictable volumetric motion: The fluid is identical to be divergence-free and smooth.
- The predictable and physically plausible motion for 2D projection pair: For input videos at each time step, the total energy (not only the brightness) is constant (i.e., energy-conserving, see Equation 6.1) from different view cameras. Furthermore, with the assumption of the parallel beam for simplicity, the energy preserving flow is additionally penalized with larger weight on the direction that perpendicular to the rotation direction.

With these observations, we propose a novel view synthesis method to improve the reconstruction quality. From the experiments, the unprecedented results can be observed from our method. The structure of the pipeline is shown in Figure 6.9. For each time step: there are three images as input (only two input images at 0 and 45 degrees shown in Figure 6.9 for simplicity). A 3D version multi-scale Horn-Schunck like flow estimation (with more penalty on the direction that perpendicular to rotation) then applied on the input image pair. With this calculated flow motion

and input image pair, a view warping with morph mode technique then applied, generating a sequence of sharp images. The obtained 45 projections, together with the masks generated from the forward rendering of tomography, are treated as an input for the 4D tomography framework described in Chapter 5. With the joint optimization of volume reconstruction and motion estimation, high-quality density volumes then obtained.

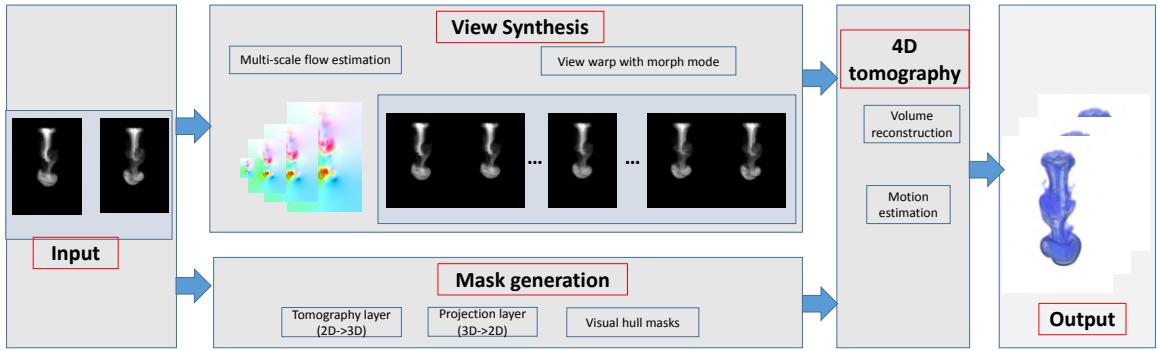


Figure 6.9: The pipeline

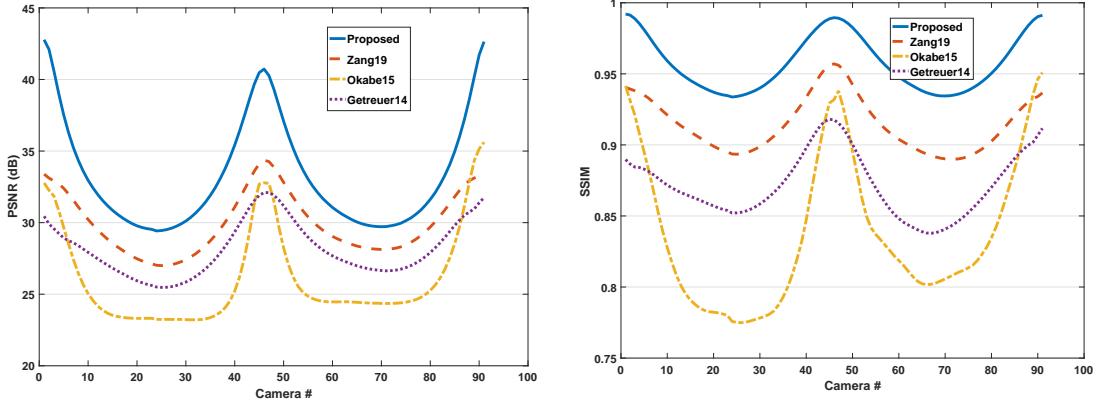


Figure 6.10: the numerical comparison for different methods for projection of reconstructed volume. The averaged result for total 46 time steps in each camera is presented.

Results and Analysis. We compare the proposed method (including divergence-free prior in the volumetric domain and view synthesis in projection domain) to three strong baselines. The first baseline is a variant of the Rudin-Osher-Fatemi total vari-

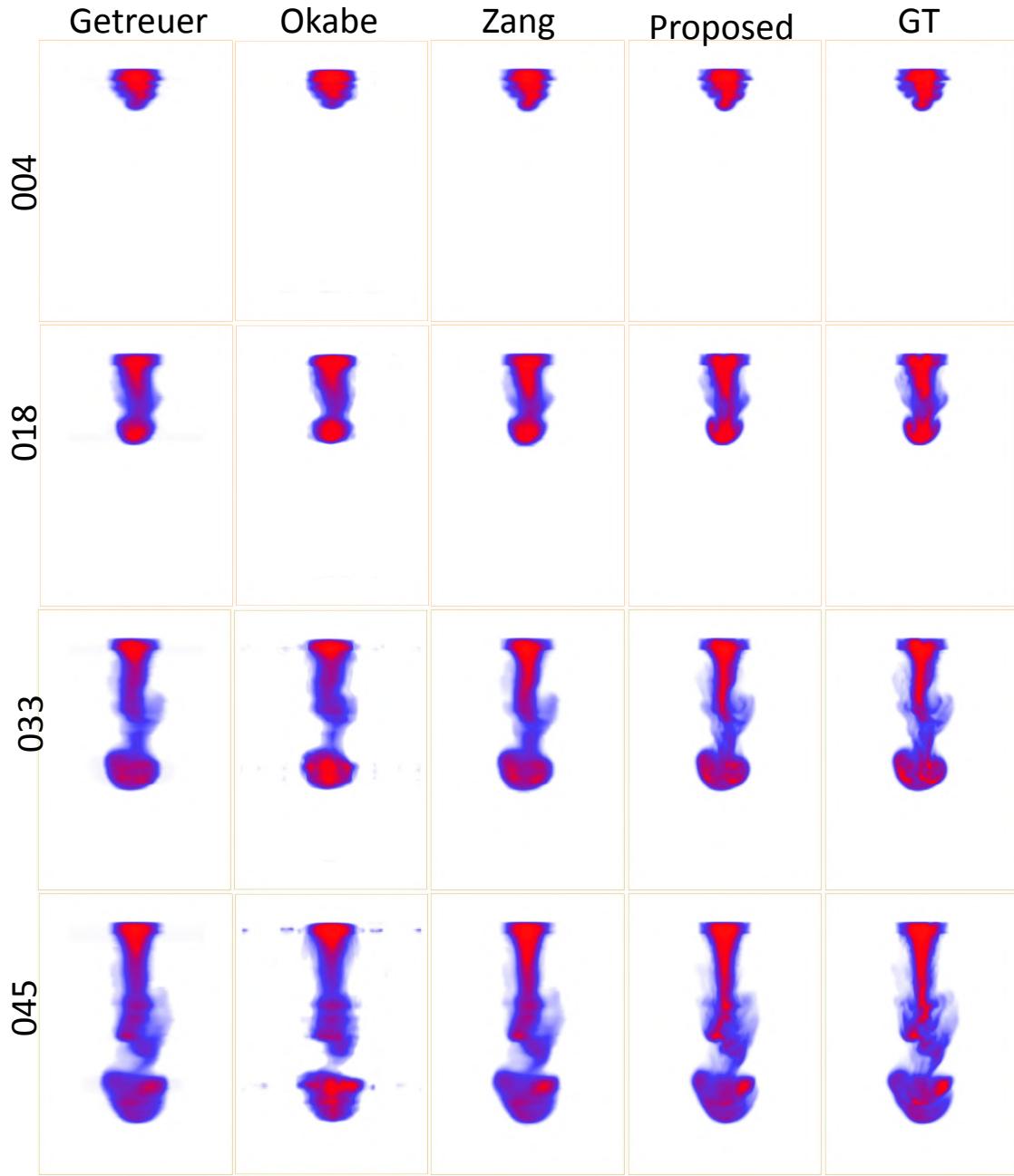


Figure 6.11: The comparison for different methods via slice visualization, display range [0,255]

ation denoising model [88], implemented by solving it in a primal-dual scheme [26], and using SART as the solver for data term (SART-ROF) to improve the reconstruction quality. The appearance transfer-based method for fluid modeling that proposed by [90] is compared as the third baseline. The last baseline reconstructs fast de-

forming object with internal structure by X-ray CT scanner via a warp and project strategy (presented in Chapter 5). The numerical result is presented in Figure 6.10, the unprecedented reconstruction quality can be observed from our proposed method for both SSIM and PSNR metrics. Additional visualization comparison is also conducted, as shown in Figure 6.11, which further verify the significant improvement from our method.

6.6 Tomographic Reconstruction for Nanoscale Chip Data

As a powerful non-destructive tool, the X-ray CT technique can also be used in other important engineering areas, for instance, design validation and quality control for integrated circuits. Recently Holler et al [91] successfully reconstructed 3D volumetric chip data with a resolution of 18.9 nm by a ptychographic X-ray laminography approach. Alternatively, with focused optics, we try to apply tomographic reconstruction techniques based on the traditional projections, with an even higher resolution.

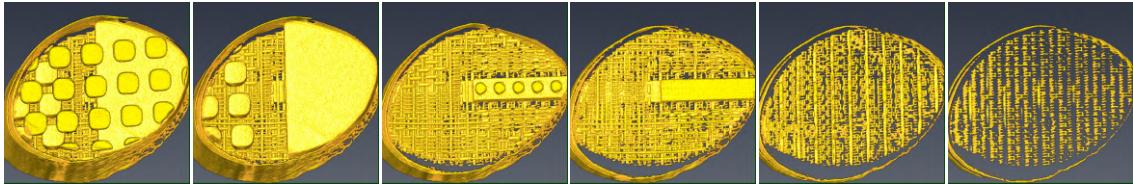


Figure 6.12: Volume visualization for chip data reconstructed by ST-Tomography-OF method, only the middle time frame is selected.

To capture and validate this type of nanoscale chip data is very challenging: First, some obvious deformation, which is mainly caused by sample degradation due to radiation, can be observed during the scanning process. Thus, a 2D alignment operation on projections as well as 3D alignments on volumes are required. Second, the imaging problem is right at the edge of the physically feasible with the current state of technology. Therefore, we typically do not have ground-truth data available, making

it hard to apply any objective metric to measure the reconstruction quality, except visual comparisons.

Figure 6.12 shows an initial result of the reconstructed volume by ST-tomography-OF method (i.e. ST-tomography method with additional 2D optical flow based alignment operation on projection). This dataset was acquired at the Advanced Photon Source synchrotron facility of the Argonne National Lab by Dr. De Andrade. The sliced volume comparison is presented in Figure 6.13. The resolution of the captured projection image is 6.6 nm, while the resolution of reconstructed volumes is 13 nm. This result shows promise for the use of our techniques to this challenging imaging problem. In the future we will explore this application in more detail, and develop tailored solutions to improve on the imaging resolution of nano-CT.

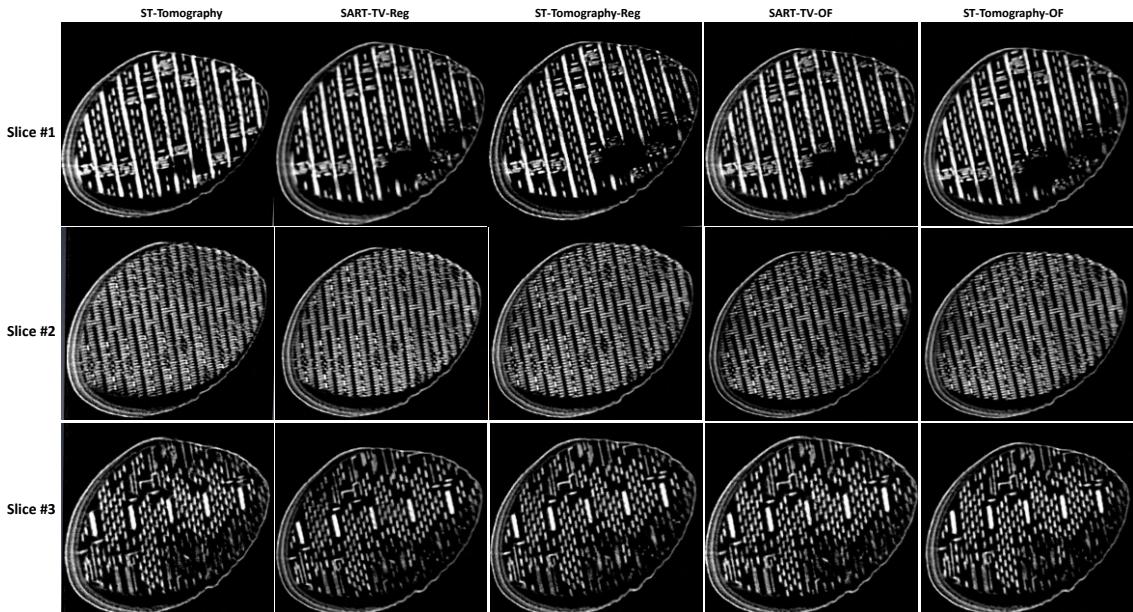


Figure 6.13: Slice visualization for chip data. From left to right: ST-tomography method (ST-Tomography), Total variation regularized SART with projection registration (SART-TV-Reg), ST-tomography with projection registration (ST-Tomography-Reg), Total variation regularized SART with optical flow based projection alignment (SART-TV-OF), and ST-tomography with optical flow based projection alignment (ST-Tomography-OF). From top to bottom: Different slice for the reconstructed volume by different methods.

Chapter 7

Conclusion

In this thesis, several new inverse problems related to computational tomography are introduced and addressed. Chapter 3 focuses on sparse view reconstruction and super-resolution for X-ray CT imaging. In this chapter we present a flexible and easy to use framework for 3D CT reconstruction. The framework is based on recently popular proximal algorithms. We derive the proximal operator for the data term using SART as a solver. In the experiments, we demonstrate several advantages of using SART over CG for this sub-problem. Another main contribution of Chapter 3 is that we extend a structure tensor-based regularizer (i.e., STP) to the 3D case, derive its proximal operator, and show its effectiveness in reconstructing specific structural features, such as thin sheets and filaments. We apply our algorithm in a super-resolution setting, where the input is a sequence of lower-resolution X-ray projection images and the required output is a higher-resolution volume reconstruction, which is beyond the Nyquist limit. We validate the efficacy of our algorithms and show superior reconstruction quality compared to existing popular methods. In the future, one of the interesting extensions is to reduce the time needed for X-ray CT acquisition, which will help reduce X-ray radiation, and potential effects of thermally expanding scanner platforms and the stress relaxation during the mechanical loading.

With the derivations of new proximal operators and additional slack variables allocated in the memory, the flexibility of our optimization framework allows us to add additional regularizers on the fly. Based on the 3D optimization framework, Chapter 4 then adapted this 3D framework to space time reconstruction, with more

efficient priors in both the spatial and the temporal domain. There are two main contributions in this chapter: First, a new sampling strategy for the scanning of moving objects. Our solution is built on using low discrepancy sequences to capture the projection images. The second key contribution is a joint image formation model and optimization method for retrieving the reconstructed volumetric results, as well as the deformation fields between them. With the assumption of relatively small motion at the scale of 10-60 projections (45-270 seconds in our experiments), an efficient and effective framework to handle deformations in X-ray computed tomography is then provided. The method can be used to analyze a wide range of dynamic phenomena, and we demonstrate it with extensive experiments on both real and simulated data.

Chapter 5 then relaxes the assumption that we made in Chapter 4, regarding ignoring the deformation in each time step (10-60 projections in the example). Instead, with a novel warp and project strategy, we show a significant improvement in the reconstruction quality compared to the state-of-the-art in tomographic reconstruction. In the future, combining the methods presented in this chapter with other tomography techniques, such as phase contrast imaging in order to enlarge the contrast for improving the reconstruction quality, will be an interesting topic.

With a physically plausible divergence-free prior for motion estimation, as well as a novel view synthesis technique, in Chapter 6, we present applications to dynamic fluid imaging (4D soot imaging of a combustion process, a mixing fluid process, a fuel injection process, and view synthesis for visible light tomography), which further demonstrates the flexibility of our optimization framework.

REFERENCES

- [1] A. C. Kak and M. Slaney, *Principles of computerized tomographic imaging*. SIAM, 2001.
- [2] G. T. Herman, *Fundamentals of computerized tomography: image reconstruction from projections*, 2009.
- [3] N. H. Clinthorne, T.-S. Pan, P.-C. Chiao, W. L. Rogers, and J. A. Stamos, “Preconditioning methods for improved convergence rates in iterative reconstructions,” *IEEE Trans. Med. Img.*, vol. 12, no. 1, pp. 78–83, 1993.
- [4] J. Xu and B. M. Tsui, “Quantifying the importance of the statistical assumption in statistical X-ray CT image reconstruction,” *IEEE Trans. Med. Img.*, vol. 33, no. 1, pp. 61–73, 2014.
- [5] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, “Distributed optimization and statistical learning via the alternating direction method of multipliers,” *Foundations and Trends in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2011.
- [6] C. Mory, B. Zhang, V. Auvray, M. Grass, D. Schäfer, F. Peyrin, S. Rit, P. Douek, and L. Boussel, “ECG-gated C-arm computed tomography using L1 regularization,” in *Proc. EUSIPCO*, 2012, pp. 2728–2732.
- [7] A. H. Andersen and A. C. Kak, “Simultaneous algebraic reconstruction technique (SART): a superior implementation of the ART algorithm,” *Ultrasonic Imaging*, vol. 6, no. 1, pp. 81–94, 1984.
- [8] O. Józsa, A. Börcs, and C. Benedek, “Towards 4D virtual city reconstruction from Lidar point cloud sequences.” ISPRS, 2013.
- [9] M. Dou, S. Khamis, Y. Degtyarev, P. Davidson, S. R. Fanello, A. Kowdle, S. O. Escolano, C. Rhemann, D. Kim, J. Taylor *et al.*, “Fusion4D: Real-time performance capture of challenging scenes,” *ACM Trans. Graph.*, vol. 35, no. 4, p. 114, 2016.
- [10] A. Reche-Martinez, I. Martin, and G. Drettakis, “Volumetric reconstruction and interactive rendering of trees from photographs,” *ACM Trans. Graph.*, vol. 23, no. 3, pp. 720–727, 2004.

- [11] S. Zhao, W. Jakob, S. Marschner, and K. Bala, “Building volumetric appearance models of fabric using micro CT imaging,” *ACM Trans. Graph.*, vol. 30, no. 4, p. 44, 2011.
- [12] W. H. Stuppy, J. A. Maisano, M. W. Colbert, P. J. Rudall, and T. B. Rowe, “Three-dimensional analysis of plant structure using high-resolution X-ray computed tomography,” *Trends in Plant Science*, vol. 8, no. 1, pp. 2–6, 2003.
- [13] T. Ijiri, S. Yoshizawa, H. Yokota, and T. Igarashi, “Flower modeling via X-ray computed tomography,” *ACM Trans. Graph.*, vol. 33, no. 4, p. 48, 2014.
- [14] J. Gregson, I. Ihrke, N. Thuerey, and W. Heidrich, “From capture to simulation: connecting forward and inverse problems in fluids,” *ACM Trans. Graph.*, vol. 33, no. 4, p. 139, 2014.
- [15] B. Trifonov, D. Bradley, and W. Heidrich, “Tomographic reconstruction of transparent objects,” in *Proc. EGSR*, 2006.
- [16] Q. Zheng, X. Fan, M. Gong, A. Sharf, O. Deussen, and H. Huang, “4D Reconstruction of Blooming Flowers,” *CGF*, vol. 36, no. 6, pp. 405–417, 2017.
- [17] I. Ihrke and M. Magnor, “Image-based tomographic reconstruction of flames,” in *Proc. SCA*, 2004, pp. 365–373.
- [18] G. E. Elsinga, F. Scarano, B. Wieneke, and B. W. van Oudheusden, “Tomographic particle image velocimetry,” *Exp. Fluids*, vol. 41, no. 6, pp. 933–947, 2006.
- [19] S. W. Hasinoff and K. N. Kutulakos, “Photo-consistent reconstruction of semi-transparent scenes by density-sheet decomposition,” *IEEE Trans. PAMI*, vol. 29, no. 5, pp. 870–885, 2007.
- [20] B. Atcheson, I. Ihrke, W. Heidrich, A. Tevs, D. Bradley, M. Magnor, and H.-P. Seidel, “Time-resolved 3D capture of non-stationary gas flows,” *ACM Trans. Graph.*, vol. 27, no. 5, p. 132, 2008.
- [21] J. Gregson, M. Krimerman, M. B. Hullin, and W. Heidrich, “Stochastic tomography and its applications in 3D imaging of mixing fluids.” *ACM Trans. Graph.*, vol. 31, no. 4, pp. 52–1, 2012.
- [22] C. Mory, V. Auvray, B. Zhang, M. Grass, D. Schäfer, S. J. Chen, J. D. Carroll, S. Rit, F. Peyrin, P. Douek *et al.*, “Cardiac C-arm computed tomography using a 3D + time ROI reconstruction method with spatial and temporal regularization,” *Med. Phys.*, vol. 41, no. 2, 2014.

- [23] G. Zang, M. Aly, R. Idoughi, P. Wonka, and W. Heidrich, “Super-resolution and sparse view ct reconstruction,” in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 137–153.
- [24] G. Zang, R. Idouchi, R. Tao, G. Lubineau, P. Wonka, and W. Heidrich, “Space-time tomography for continuously deforming objects,” *ACM Trans. Graph.*, vol. 37, no. 4, p. 100, 2018.
- [25] G. Zang, R. Idoughi, R. Tao, G. Lubineau, P. Wonka, and W. Heidrich, “Warp-and-project tomography for rapidly deforming objects,” *ACM Trans. Graph.*, vol. 38, no. 4, 2019.
- [26] A. Chambolle and T. Pock, “A first-order primal-dual algorithm for convex problems with applications to imaging,” *J. Math. Imaging and Vision*, vol. 40, no. 1, pp. 120–145, 2011.
- [27] N. Parikh, S. Boyd *et al.*, “Proximal algorithms,” *Foundations and Trends® in Optimization*, vol. 1, no. 3, pp. 127–239, 2014.
- [28] N. Parikh and S. Boyd, “Proximal algorithms,” *Foundations and Trends in Optimization*, vol. 1, no. 3, pp. 123–231, 2013.
- [29] E. Esser, X. Zhang, and T. F. Chan, “A general framework for a class of first order primal-dual algorithms for convex optimization in imaging science,” *SIAM Journal on Imaging Sciences*, vol. 3, no. 4, pp. 1015–1046, 2010.
- [30] M. Innmann, M. Zollhöfer, M. Nießner, C. Theobalt, and M. Stamminger, “VolumeDeform: Real-time volumetric non-rigid reconstruction,” in *Proc. ECCV*. Springer, 2016, pp. 362–379.
- [31] S. Wang, X. Zuo, C. Du, R. Wang, J. Zheng, and R. Yang, “Dynamic non-rigid objects reconstruction with a single RGB-D sensor,” vol. 18, no. 3, p. 886, 2018.
- [32] Y. Li, X. Fan, N. J. Mitra, D. Chamovitz, D. Cohen-Or, and B. Chen, “Analyzing growing plants from 4D point cloud data,” *ACM Trans. Graph.*, vol. 32, no. 6, p. 157, 2013.
- [33] H. Wang, M. Liao, Q. Zhang, R. Yang, and G. Turk, “Physically guided liquid surface modeling from videos,” *ACM Trans. Graph.*, vol. 28, no. 3, p. 90, 2009.
- [34] L. Feldkamp, L. Davis, and J. Kress, “Practical cone-beam algorithm,” *JOSA A*, vol. 1, no. 6, pp. 612–619, 1984.

- [35] W. Chlewicki, C. Badea, and N. Pallikarakis, “Cone based 3d reconstruction: a FDK-SART comparison for limited number of projections,” *Proceedings of MEDICON 2001*, 2001.
- [36] E. Y. Sidky and X. Pan, “Image reconstruction in circular cone-beam computed tomography by constrained, total-variation minimization,” *Physics in Medicine and Biology*, vol. 53, no. 17, p. 4777, 2008.
- [37] G.-H. Chen, P. Thériault-Lauzier, J. Tang, B. Nett, S. Leng, J. Zambelli, Z. Qi, N. Bevins, A. Raval, S. Reeder *et al.*, “Time-resolved interventional cardiac C-arm cone-beam CT: An application of the PICCS algorithm,” *IEEE Trans. Med. Img.*, vol. 31, no. 4, pp. 907–923, 2012.
- [38] J.-J. Sonke, L. Zijp, P. Remeijer, and M. van Herk, “Respiratory correlated cone beam CT,” *Med. Phys.*, vol. 32, no. 4, pp. 1176–1186, 2005.
- [39] M. L. Schmidt, P. R. Poulsen, J. Toftegaard, L. Hoffmann, D. Hansen, and T. S. Sørensen, “Clinical use of iterative 4d-cone beam computed tomography reconstructions to investigate respiratory tumor motion in lung cancer patients,” *Acta Oncologica*, vol. 53, no. 8, pp. 1107–1113, 2014.
- [40] R. Zeng, J. A. Fessler, and J. M. Balter, “Estimating 3-d respiratory motion from orbiting views by tomographic image registration,” *IEEE Trans. Med. Img.*, vol. 26, no. 2, pp. 153–163, 2007.
- [41] T. Li, A. Koong, and L. Xing, “Enhanced 4d cone-beam ct with inter-phase motion model,” *Medical physics*, vol. 34, no. 9, pp. 3688–3695, 2007.
- [42] O. Taubmann, G. Lauritsch, A. Maier, R. Fahrig, and J. Hornegger, “Estimate, compensate, iterate: joint motion estimation and compensation in 4-D cardiac C-arm computed tomography,” in *Proc. Int. Conf. on Medical Image Computing and Computer-Assisted Intervention*, 2015, pp. 579–586.
- [43] T. De Schryver, M. Dierick, M. Heyndrickx, J. Van Stappen, M. A. Boone, L. Van Hoorebeke, and M. N. Boone, “Motion compensated micro-CT reconstruction for in-situ analysis of dynamic processes,” *Scientific reports*, vol. 8, no. 1, p. 7655, 2018.
- [44] O. Weissenborn, S. Geller, M. Gude, F. Post, S. Praetorius, A. Voigt, and S. Aland, “Deformation analysis of polymer foams under compression load using in situ computed tomography and finite element simulation methods,” in *17th European conference on Composite Materials (ECCM-17)*, 2016.

- [45] J. Lachambre, J. Réthoré, A. Weck, and J.-Y. Buffiere, “Extraction of stress intensity factors for 3D small fatigue cracks using digital volume correlation and X-ray tomography,” *Int. J. Fatigue*, vol. 71, pp. 3–10, 2015.
- [46] T. F. Morgeneyer, L. Helfen, H. Mubarak, and F. Hild, “3D digital volume correlation of synchrotron radiation laminography images of ductile crack initiation: an initial feasibility study,” *Exp. Mech.*, vol. 53, no. 4, pp. 543–556, 2013.
- [47] S. Shah, J. Yang, J. P. Crawshaw, O. Gharbi, E. S. Boek *et al.*, “Predicting porosity and permeability of carbonate rocks from core-scale to pore-scale using medical ct, confocal laser scanning microscopy and micro ct,” in *SPE Annual Technical Conference and Exhibition*. Society of Petroleum Engineers, 2013.
- [48] S. Shah, F. Gray, J. Crawshaw, and E. Boek, “Micro-computed tomography pore-scale study of flow in porous media: Effect of voxel resolution,” *Advances in water resources*, vol. 95, pp. 276–287, 2016.
- [49] A. Shastry, P. Palacio-Mancheno, K. Braeckman, S. Vanheule, I. Josipovic, F. Van Assche, E. Robles, V. Cnudde, L. Van Hoorebeke, and M. Boone, “In-situ high resolution dynamic x-ray microtomographic imaging of olive oil removal in kitchen sponges by squeezing and rinsing,” *Materials*, vol. 11, no. 8, p. 1482, 2018.
- [50] F. Hild, H. Leclerc, and S. Roux, “Performing DVC at the Voxel Scale,” in *Advancement of Optical Methods in Experimental Mechanics, Volume 3*, 2014, pp. 209–215.
- [51] J. Neggers, J. Hoefnagels, M. Geers, F. Hild, and S. Roux, “Time-resolved integrated digital image correlation,” *International Journal for Numerical Methods in Engineering*, vol. 103, no. 3, pp. 157–182, 2015.
- [52] H. Leclerc, S. Roux, and F. Hild, “Projection savings in CT-based digital volume correlation,” *Exp. Mech.*, vol. 55, no. 1, pp. 275–287, 2015.
- [53] T. Taillandier-Thomas, S. Roux, and F. Hild, “Soft route to 4D tomography,” *Phys. Rev. Letters*, vol. 117, no. 2, p. 025501, 2016.
- [54] C. Jailin and S. Roux, “Dynamic tomographic reconstruction of deforming volumes,” *Materials*, vol. 11, no. 8, p. 1395, 2018.
- [55] D. Lanman, G. Wetzstein, M. Hirsch, W. Heidrich, and R. Raskar, “Polarization fields: dynamic light field display using multi-layer LCDs,” *ACM Trans. Graph.*, vol. 30, no. 6, p. 186, 2011.

- [56] G. Wetzstein, D. Lanman, W. Heidrich, and R. Raskar, “Layered 3D: tomographic image synthesis for attenuation-based light field and high dynamic range displays,” *ACM Trans. Graph.*, vol. 30, no. 4, p. 95, 2011.
- [57] R. Anirudh, H. Kim, J. J. Thiagarajan, K. A. Mohan, K. Champley, and T. Bremer, “Lose the views: Limited angle CT reconstruction via implicit sinogram completion,” in *Proc. CVPR*, vol. 2, 2018.
- [58] S. Kato, T. Narita, C. Tomiyama, T. Ijiri, and H. Tanaka, “4D computed tomography measurement for growing plant animation,” in *SIGGRAPH ASIA 2017 Posters*, 2017.
- [59] H. Schreier, J.-J. Orteu, and M. A. Sutton, *Image correlation for shape, motion and deformation measurements*. Springer US, 2009.
- [60] B. K. Bay, T. S. Smith, D. P. Fyhrie, and M. Saad, “Digital volume correlation: three-dimensional strain mapping using X-ray tomography,” *Exp. Mech.*, vol. 39, no. 3, pp. 217–226, 1999.
- [61] E. Verhulp, B. van Rietbergen, and R. Huiskes, “A three-dimensional digital image correlation technique for strain measurements in microstructures,” *J. Biomechanics*, vol. 37, no. 9, pp. 1313–1320, 2004.
- [62] L. Desbat, S. Roux, and P. Grangeat, “Compensation of some time dependent deformations in tomography,” *IEEE Trans. Med. Img.*, vol. 26, no. 2, pp. 261–269, 2007.
- [63] S. Roux, F. Hild, P. Viot, and D. Bernard, “Three-dimensional image correlation from X-ray computed tomography of solid foam,” *Composites Part A: Applied science and manufacturing*, vol. 39, no. 8, pp. 1253–1265, 2008.
- [64] J. Weickert, *Anisotropic diffusion in image processing*. Teubner Stuttgart, 1998, vol. 1.
- [65] S. Lefkimiatis, A. Roussos, P. Maragos, and M. Unser, “Structure tensor total variation,” *SIAM Journal on Imaging Sciences*, vol. 8, no. 2, pp. 1090–1122, 2015.
- [66] S. Lefkimiatis, A. Roussos, M. Unser, and P. Maragos, “Convex generalizations of total variation based on the structure tensor with applications to inverse problems,” in *Scale Space and Variational Methods in Computer Vision*, ser. Lecture Notes in Computer Science, A. Kuijper, K. Bredies, T. Pock, and H. Bischof, Eds. Springer Berlin Heidelberg, 2013, vol. 7893, pp. 48–60. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-38267-3_5

- [67] S. Lefkimiatis, J. P. Ward, and M. Unser, “Hessian schatten-norm regularization for linear inverse problems,” *Image Processing, IEEE Transactions on*, vol. 22, no. 5, pp. 1873–1888, 2013.
- [68] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.
- [69] S. Ramani and J. A. Fessler, “A splitting-based iterative algorithm for accelerated statistical x-ray ct reconstruction,” *Medical Imaging, IEEE Transactions on*, vol. 31, no. 3, pp. 677–688, 2012.
- [70] A. Bjorck, *Numerical methods for least squares problems*. Siam, 1996, vol. 51.
- [71] G. H. Golub and C. F. Van Loan, *Matrix computations*. JHU Press, 2012, vol. 3.
- [72] S. Boyd, “Ee364b: Convex optimization ii,” *Course Notes*, <http://www.stanford.edu/class/ee364b>, 2012.
- [73] K. Mueller, R. Yagel, and J. J. Wheller, “Fast implementations of algebraic methods for three-dimensional reconstruction from cone-beam data,” *Medical Imaging, IEEE Transactions on*, vol. 18, no. 6, pp. 538–548, 1999.
- [74] E. Y. Sidky, J. H. Jørgensen, and X. Pan, “Convex optimization problem prototyping for image reconstruction in computed tomography with the chambolle–pock algorithm,” *Physics in medicine and biology*, vol. 57, no. 10, p. 3065, 2012.
- [75] S. Rit, M. V. Oliva, S. Brousmiche, R. Labarbe, D. Sarrut, and G. C. Sharp, “The reconstruction toolkit (rtk), an open-source cone-beam ct reconstruction toolkit based on the insight toolkit (itk),” in *Journal of Physics: Conference Series*, vol. 489, no. 1. IOP Publishing, 2014, p. 012079.
- [76] W. van Aarle, W. J. Palenstijn, J. Cant, E. Janssens, F. Bleichrodt, A. Dabravolski, J. De Beenhouwer, K. J. Batenburg, and J. Sijbers, “Fast and flexible x-ray tomography using the astra toolbox,” *Optics express*, vol. 24, no. 22, pp. 25 129–25 147, 2016.
- [77] M. Aly, G. Zang, W. Heidrich, and P. Wonka, “Trex: A tomography reconstruction proximal framework for robust sparse view x-ray applications,” *arXiv preprint arXiv:1606.03601*, 2016.
- [78] X. Pan, E. Y. Sidky, and M. Vannier, “Why do commercial CT scanners still employ traditional, filtered back-projection for image reconstruction?” *Inverse Problems*, vol. 25, no. 12, p. 123009, 2009.

- [79] D. Forsyth and J. Ponce, *Computer Vision: A modern approach*. Prentice Hall, 2002.
- [80] K. A. Mohan, S. Venkatakrishnan, J. W. Gibbs, E. B. Gulsoy, X. Xiao, M. De Graef, P. W. Voorhees, and C. A. Bouman, “TIMBIR: A method for time-space reconstruction from interlaced views,” vol. 1, no. 2, pp. 96–111, 2015.
- [81] J. Van der Corput, “Verteilungsfunktionen i, ii,” *Nederl. Akad. Wetensch. Proc.* 38, 1935.
- [82] M. Burger, H. Dirks, and C.-B. Schonlieb, “A variational model for joint motion estimation and image reconstruction,” *SIAM Journal on Imaging Sciences*, vol. 11, no. 1, pp. 94–128, 2018.
- [83] J. Xiong, R. Idoughi, A. A. Aguirre-Pablo, A. B. Aljedaani, X. Dun, Q. Fu, S. T. Thoroddsen, and W. Heidrich, “Rainbow particle imaging velocimetry for dense 3D fluid velocity imaging,” *ACM Trans. Graph.*, vol. 36, no. 4, p. 36, 2017.
- [84] B. K. Horn and B. G. Schunck, “Determining optical flow,” *Artificial Intelligence*, vol. 17, no. 1-3, pp. 185–203, 1981.
- [85] E. Meinhardt-Llopis, J. S. Pérez, and D. Konermann, “Horn-schunck optical flow with a multi-scale strategy. image processing on line, 2013: 151–172, 2013,” 2013.
- [86] P. J. Huber, “Robust statistics,” in *Int. Encyclopedia of Statistical Science*, 2011, pp. 1248–1251.
- [87] B. Kim, Y. Liu, I. Llamas, and J. R. Rossignac, “Flowfixer: Using bfcc for fluid simulation,” Georgia Institute of Technology, Tech. Rep., 2005.
- [88] P. Getreuer, “Rudin-oshер-fatemi total variation denoising using split bregman,” *Image Processing On Line*, vol. 2, pp. 74–95, 2012.
- [89] N. Thuerey and T. Pfaff, “MantaFlow,” 2018, <http://mantaflow.com>.
- [90] M. Okabe, Y. Dobashi, K. Anjyo, and R. Onai, “Fluid volume modeling from sparse multi-view images by appearance transfer,” *ACM Trans. Graph.*, vol. 34, no. 4, p. 93, 2015.
- [91] M. Holler, M. Odstrcil, M. Guizar-Sicairos, M. Lebugle, E. Müller, S. Finizio, G. Tinti, C. David, J. Zusman, W. Unglaub *et al.*, “Three-dimensional imaging of integrated circuits with macro-to nanoscale zoom,” *Nature Electronics*, vol. 2, no. 10, pp. 464–470, 2019.