application to 4D-CT Adjoint image warping in 4D-CT Jens Renders, Jan Sijbers and Jan De Beenhouwer imec-Vision Lab, University of Antwerp, Universiteitsplein 1, 2610 Antwerp, Belgium Version typeset August 25, 2021 email: jens.renders@uantwerpen.be Abstract Purpose: Adjoint image warping is an important tool to solve image reconstruction problems that warp the unknown image in the forward model. This includes 4D-CT models in which images are compared against recorded projection images of various

Adjoint image warping using multivariate splines with

Purpose: Adjoint image warping is an important tool to solve image reconstruction problems that warp the unknown image in the forward model. This includes 4D-CT models in which images are compared against recorded projection images of various time frames using image warping as a model of the motion. The inversion of these models require the adjoint of image warping, which up to now has been substituted by approximations. We introduce an efficient implementation of the exact adjoints of multivariate spline based image warping, and compare it against previously used

- Methods: Using symbolic computer algebra, we computed a list of 64 polynomials 18 that allow us to compute a matrix representation of trivariate cubic image warping. By 19 combining an on-the-fly computation of this matrix with a parallelized implementation 20 of columnwise matrix multiplication, we obtained an efficient, low memory implemen-21 tation of the adjoint action of 3D cubic image warping. We used this operator in the 22 solution of a previously proposed 4D-CT reconstruction model in which the image of a 23 single subscan was compared against projection data of multiple subscans by warping 24 and then projecting the image. We compared the properties of our exact adjoint with 25 those of approximate adjoints by warping along inverted motion. 26
- **Results:** Our method requires halve the memory to store motion between subscans, 27 compared to methods that need to compute and store an approximate inverse of the 28 motion. It also avoids the computation time to invert the motion and the tunable 29 parameter of the number of iterations used to preform this inversion. Yet, a similar 30 and often better reconstruction quality was obtained in comparison with these more 31 expensive methods, especially when the motion is large. When compared against a sim-32 pler method that is similar to ours in computational demands, our method achieves a 33 higher reconstruction quality in general. 34
- Conclusions: Our implementation of the exact adjoint of cubic image warping im proves efficiency and provides accurate reconstructions.
- 37

1

2

3

4

5

6

7 8

9

17

alternatives.

# 38 Contents

39	١.	Introduction	1
40	п.	Materials and Methods	3
41		II.A. Image warping operators	3
42		II.B. Tricubic interpolation	4
43		II.C. Adjoint image warping with tricubic interpolation	5
44		II.D. Dynamic tomographic model	6
45		II.E. Solver	7
46		II.F. Experiments	8
47		$II.F.1.$ Simulation experiment with known DVF $\hdots$	9
48		II.F.2. Simulation experiment with unknown DVF	9
49		II.F.3. Liquid foam dataset	11
50	ш.	Results	13
51		III.A. Simulation experiment with known DVF $\ldots$	13
52		III.B. Simulation experiment with unknown DVF	13
53		III.C. Liquid foam dataset	13
54	IV.	Discussion	22
55		IV.A.Simulation experiment with known $DVF$	22
56		IV.B. Simulation experiment with unknown DVF	23
57		IV.C. Liquid foam dataset	24
58	v.	Conclusions	25
59	VI.	Acknowledgements	25
60	VII	Conflict of Interest Statement	25
61		References	26

### 62 I. Introduction

A 4D-CT scan is a series of n consecutive, regular CT scans, called subscans. These subscans 63 are used to capture different 3D images of a deforming object. It is generally assumed that 64 the scanned object is motionless during each subscan, such that the scan can be modeled 65 by the equation  $W_i x_i = p_i$ . Here,  $x_i \in \mathbb{R}^N$  is a column vector representing an image 66 with N voxels,  $\boldsymbol{p}_i \in \mathbb{R}^K$  is a column vector representing the projection data of the *i*-th 67 subscan, with K the number of detector pixels times the number of projections, and  $W_i$  is a 68  $K \times N$  matrix representing the projection operator of the *i*-th subscan. If the object is not 69 motionless, the extent to which this static model is accurate depends on the time that passed 70 during the subscan. For this reason, subscans are usually fast scans with few projections. 71 Reconstructing a 4D image from a 4D-CT scan with n subscans then corresponds to solving 72 n linear systems 73

$$\boldsymbol{W}_{i}\boldsymbol{x}_{i}=\boldsymbol{p}_{i},\quad i=1,\cdots,n$$
 (1)

which are often highly underdetermined due to the low number of projections. Equivalently,
the problem can be represented as a single underdetermined system of the form

$$\begin{bmatrix} \boldsymbol{W}_{1} & 0 & 0 & 0 \\ 0 & \boldsymbol{W}_{2} & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & \boldsymbol{W}_{n} \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_{1} \\ \boldsymbol{x}_{2} \\ \vdots \\ \boldsymbol{x}_{n} \end{bmatrix} = \begin{bmatrix} \boldsymbol{p}_{1} \\ \boldsymbol{p}_{2} \\ \vdots \\ \boldsymbol{p}_{n} \end{bmatrix} \quad . \tag{2}$$

One way to alleviate this underdetermination is to link the time frames using image rr registration or optical flow techniques<sup>1,2,3,4</sup> and image warping<sup>5</sup>. Each specific image can then be reconstructed with a motion compensated reconstruction technique that combines the projection data of all subscans. Such techniques can be roughly classified as follows:

 The motion compensated reconstruction techniques employed in <sup>6,7,8,9</sup> first make a reconstruction of each subscan separately. The resulting reconstructions are then warped to a single point in time where they are averaged. This average reconstruction then depends on the projection data of all subscans. This type of method is very fast and it does not require the adjoint or the inverse of the warping operators.

2. In the MoVIT algorithm<sup>10</sup> and in<sup>11</sup>, the images are warped along the flow between
 frames before they are projected using a standard projection geometry. This entire

forward model is then iteratively inverted. Such an iterative procedure requires the adjoint of all operators present in the forward model. The adjoint of the projection operator is (non-filtered) backprojection and has been well studied. The adjoint of the image warping operator was substituted by an approximation of the inverse warp. It was shown in<sup>10</sup>, that this iterative approach outperforms the averaging of separate reconstructions in terms of reconstruction quality.

In<sup>12,13</sup>, Eq. (2) is regularized with terms that constrain the change of the object between
frames, i.e. with constraints on the optical flow. In<sup>13</sup>, warping operators are involved in
these regularization terms of the objective function, and the adjoints of these operators
are needed to use convex optimization techniques for minimizing it.

In<sup>14,15</sup>, the optical flow between frames is accounted for in the projection operators, by
using a curved ray projection geometry instead of explicitly involving image warping.
Solving the resulting system requires the adjoint of this specialized projection operator.
Since the projection operator can be seen as the composition of a regular projection
operator and image warps, adjoint image warping can also be employed here.

Next to motion compensated reconstruction, there are other inverse problems that involve
warping operators in their forward model, such as the direct inversion of a warping operator
as used in<sup>17</sup>.

The above mentioned methods implement the adjoint image warping operators by warp-105 ing along an approximated inverse of the flow, or they are restricted to very small examples 106 where they can work with matrix representations of the operators and their transpose. Com-107 puting the inverse of the flow requires computation time and memory. On top of that, since 108 the flow is generally not exactly invertible and the adjoint of a warp is not exactly the warp 109 along the inverse flow, it introduces inaccuracies. Few papers have investigated how these 110 approximations compare with the exact adjoints. In<sup>16</sup>, a pair of adjoint warping operators 111 with a custom linear interpolation method is used for respiratory and cardiac motion cor-112 rection in 4D PET. It is shown that using inverse warps as an approximation for the adjoint 113 warp leads to image degradation compared to using the exact adjoint warps. 114

<sup>115</sup> In this work, an efficient, GPU based algorithm that exactly computes the adjoint action <sup>116</sup> of generic multivariate spline based image warping is introduced. This algorithm avoids the memory overhead of storing an inverted flow. We specifically focus on 3D cubic warping, but
our methods are applicable in general to multivariate spline based warping of any degree.

### <sup>119</sup> II. Materials and Methods

### <sup>120</sup> II.A. Image warping operators

While the scanned object can change during a 4D-CT scan, a standard assumption of optical 121 flow based methods is that the attenuation values of the materials that make up the object 122 do not change, but only get repositioned. This assumption is only approximately valid, 123 which puts a limit on the number of time frames we will be able to combine. Under this 124 assumption, we can deform an image  $\boldsymbol{x}_i \in \mathbb{R}^N$  into the image  $\boldsymbol{x}_i \in \mathbb{R}^N$  of a different time 125 frame, by moving its voxel values without changing them. For each voxel, a vector in  $\mathbb{R}^3$ 126 describes its displacement. Together, these displacement vectors form a displacement vector 127 field or deformation vector field (DVF) representing the optical flow between the images. 128

Repositioning the voxel values according to the DVF results in non-grid data, because 129 the voxels are allowed to move to non-integer coordinates. To turn the result back into 130 an image in  $\mathbb{R}^N$ , resampling is required. General image warping is the combined action 131 of repositioning the voxels and resampling. A standard choice of resampling method is 132 multivariate spline interpolation (usually linear or cubic splines), used in for example  $^{5,10,11}$ . 133 With this choice, each voxel in the warped image is a linear combination of voxels in the 134 original image, so such warping operators are linear maps. We will write  $M_{ij}$  to denote an 135  $N \times N$  warping operator that transforms  $\boldsymbol{x}_i$  into  $\boldsymbol{x}_j$ . 136

There are two different approaches to implement image warping operators, referred to 137 as forward and backward warping<sup>5</sup> (see Fig. 1). Assume we have two images,  $\boldsymbol{x}_a$  and  $\boldsymbol{x}_b$  and 138 a DVF describing the flow from  $x_a$  to  $x_b$ . With forward image warping, the voxels of  $x_a$  are 139 first repositioned along the DVF to obtain non-grid data representing the warped version of 140  $x_a$ , and this non-grid data is then resampled at grid points to get an image similar to  $x_b$ . 141 Backward image warping is another approach, in which the DVF is followed in the opposite 142 direction. For each voxel, we look at the position it is sent to by the DVF and interpolate 143 the regular grid data of  $x_b$  at that point. The result is an image similar to  $x_a$ . 144



Figure 1: Forward and backward image warping.

In this work, we derive the adjoint operators of backward image warping operators that use multivariate spline interpolation to interpolate the regular grid data of the image that is warped. The methods are described for tricubic interpolation, which is 3D or trivariate spline interpolation using cubic splines. It is also the warping method that is used by<sup>10</sup> and<sup>11</sup>. Adjoint warping operators for the less complicated trilinear, bicubic and bilinear interpolation methods were also implemented.

#### <sup>151</sup> II.B. Tricubic interpolation

Let  $f : \mathbb{Z}^3 \to \mathbb{R}$  be a function with values at only integer coordinates. Tricubic interpolation extends such a function to a piecewise polynomial function  $\hat{f} : \mathbb{R}^3 \to \mathbb{R}$  which agrees with fon integer coordinates and is differentiable in every point. In each cube of the grid  $\mathbb{Z}^3$ , the interpolant  $\hat{f}$  is given by a multivariate polynomial of the form

$$\hat{f}(x,y,z) = \sum_{l=0}^{3} \sum_{m=0}^{3} \sum_{n=0}^{3} a_{lmn} x^{l} y^{m} z^{n} \quad .$$
(3)

We assume to be working on the cube that has (0,0,0) as its lowest coordinate value. 156 All other cubes can be interpolated by shifting. The 64 coefficients  $a_{lmn}$  can be obtained by 157 demanding that  $\hat{f}$  agrees with f, and by putting constraints on the differentiability of  $\hat{f}$ . Once 158 64 independent linear constraints have been made, the coefficients are determined<sup>18</sup>. Another 159 common way of computing  $\hat{f}$  is via a composition of successive cubic 1D interpolations, using 160 for example Catmull-Rom splines (see e.g.<sup>19,20</sup>). By computing the composition numerically, 161 the coefficients  $a_{lmn}$  are never explicitly needed, but they can be determined by performing 162 the composition symbolically. In both cases, the coefficients  $a_{lmn}$  are linear combinations of 163

the 64 values of f on the  $4 \times 4 \times 4$  cube surrounding the cube that is being interpolated. More explicitly, every coefficient  $a_{lmn}$  is of the form

$$a_{lmn} = \sum_{i=-1}^{2} \sum_{j=-1}^{2} \sum_{k=-1}^{2} \alpha_{lmn}^{ijk} f(i,j,k)$$
(4)

with  $\forall i, j, k : \alpha_{lmn}^{ijk} \in \mathbb{R}$ . Equivalently,  $\hat{f}$  is given by an expression of the form

$$\hat{f}(x,y,z) = \sum_{i=-1}^{2} \sum_{j=-1}^{2} \sum_{k=-1}^{2} b_{ijk}(x,y,z) f(i,j,k)$$
(5)

where each coefficient  $b_{ijk}(x, y, z)$  is a multivariate polynomial of the form

$$b_{ijk}(x,y,z) = \sum_{l=0}^{3} \sum_{m=0}^{3} \sum_{n=0}^{3} \beta_{ijk}^{lmn} x^{l} y^{m} z^{n}$$
(6)

with  $\forall l, n, m : \beta_{ijk}^{lnm} \in \mathbb{R}$ . The second representation of  $\hat{f}$  is obtained, simply by rearranging the terms in Eq. (3). This rearrangement is a straightforward but tedious computation to do by hand. Instead, it was performed using the open source computer algebra package Sage-Math<sup>21</sup> (the code is available here<sup>22</sup>). The benefit of Eq. (5) is that the 64 polynomials  $b_{ijk}$ express explicitly how the interpolated value  $\hat{f}(x, y, z)$  depends linearly on the surrounding values of f.

### <sup>174</sup> II.C. Adjoint image warping with tricubic interpolation

A 3D image can be represented by a function  $f: \mathbb{Z}^3 \to \mathbb{R}$  which is zero outside the image 175 bounds, or it can be represented by a column vector  $\boldsymbol{f} \in \mathbb{R}^N$ ,  $N = \text{width} \times \text{height} \times \text{depth}$ , 176 which lists all voxel values in a predetermined order. The first representation allows to 177 use tricubic interpolation on the image, while the second representation describes an image 178 warping operator as a linear map  $M : \mathbb{R}^N \to \mathbb{R}^N$ , which can be represented as an  $N \times N$ 179 matrix. The equation g = Mf expresses the warping of image f with resulting image g. 180 Each voxel value of g is obtained by taking a linear combination of 64 voxel values of f, 181 with weights given by the coefficients  $b_{ijk}$  of Eq. (5). These weights make up the 64 non-zero 182 entries of one row of matrix M, and M has one such row for each voxel of f. Now that a 183 rowwise representation of M is known, a columnwise representation of  $M^T$  can be obtained 184 by transposing:  $M^T = (m_1^T, \cdots, m_N^T)$ , where  $m_i$  is the *i*-th row of M. An adjoint image 185

warping operator can then be implemented as a columnwise matrix multiplication with  $M^T$ :

$$\boldsymbol{M}^{T}\boldsymbol{g} = \sum_{i=1}^{N} g_{i}\boldsymbol{m}_{i}^{T}$$
(7)

The matrix M or  $M^T$  does not need to be stored. Instead, the matrix multiplication can be performed by computing the rows of M or the columns of  $M^T$  on the fly, leading to the following algorithm:

	Algorithm 1: Adjoint tricubic image warping
_	$\mathbf{input}$ : Image $\boldsymbol{g}$ , DVF
	<b>output:</b> Adjoint warped image $f$
	1 $f=0$
	2 foreach voxel position $(p_1, p_2, p_3)$ do
191	<b>3</b> $(q_1, q_2, q_3) = (p_1, p_2, p_3) + \text{DVF}(p_1, p_2, p_3)$
	$4  (i',j',k') = (\lfloor q_1 \rfloor, \lfloor q_2 \rfloor, \lfloor q_3 \rfloor)$
	5 for $(i, j, k) \in \{-1, 0, 1, 2\}^3$ do
	6 $f(i'+i,j'+j,k'+k) = b_{ijk}(i'-q_1,j'-q_2,k'-q_3)g(p_1,p_2,p_3)$
	7 end
	8 end

When working with large 3D volumes, it is important to leverage the performance gains 192 of parallelization. The outer for loop of algorithm 1 can be computed in parallel, but special 193 care needs to be taken because of the addition on line 6. The position to which we are adding 194 is dictated by the DVF, and it is possible that multiple threads would write to the same 195 position, leading to a race condition. By implementing the addition as an atomic addition, 196 race conditions can be avoided and additions are performed sequentially if needed. Because 197 the amount of overlap caused by the DVF pointing multiple times to the same area is in 198 most cases relatively small, the achieved parallelization is almost indistinguishable from a 199 complete parallelization. A GPU accelerated implementation is available here  $^{22}$ . 200

#### <sup>201</sup> II.D. Dynamic tomographic model

We will now describe the dynamic tomographic model used in the MoVIT algorithm<sup>10</sup>. This model was used in our experiments to validate our algorithm of adjoint image warping. Suppose that we want to reconstruct a certain time frame of interest,  $x_i$ . In an ideal situation, any time frame  $x_j$  can be produced from  $x_i$  by using a suitable warping operator  $M_{ij}$ , such 206 that  $\boldsymbol{x}_j = \boldsymbol{M}_{ij} \boldsymbol{x}_i$  holds for all j. This can be written as:

$$\begin{bmatrix} \boldsymbol{M}_{i1} \\ \boldsymbol{M}_{i2} \\ \vdots \\ \boldsymbol{M}_{in} \end{bmatrix} \boldsymbol{x}_{i} = \begin{bmatrix} \boldsymbol{x}_{1} \\ \boldsymbol{x}_{2} \\ \vdots \\ \vdots \\ \boldsymbol{x}_{n} \end{bmatrix}$$
(8)

<sup>207</sup> Substituting Eq. (8) into Eq. (2) yields the dynamic tomographic model:

$$\begin{bmatrix} \mathbf{W}_{1} & 0 & 0 & 0 \\ 0 & \mathbf{W}_{2} & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & \mathbf{W}_{n} \end{bmatrix} \begin{bmatrix} \mathbf{M}_{i1} \\ \mathbf{M}_{i2} \\ \vdots \\ \mathbf{M}_{in} \end{bmatrix} \mathbf{x}_{i} = \begin{bmatrix} \mathbf{p}_{1} \\ \mathbf{p}_{2} \\ \vdots \\ \mathbf{p}_{n} \end{bmatrix} , \qquad (9)$$

<sup>208</sup> or more concisely:

$$\boldsymbol{W}\boldsymbol{M}_{i}\boldsymbol{x}_{i}=\boldsymbol{p}\quad.$$

In practice, Eq. (8) can not be achieved exactly, but by estimating the DVFs between 209 initial reconstructions, a good approximation can be obtained. There are many image reg-210 istration and optical flow algorithms available to estimate the DVFs, such as  $^{1,2,3,4}$ . We used 211 the TV-L1 optical flow algorithm of<sup>3</sup> implemented along the lines of<sup>23</sup>. System Eq. (10) is 212 the one that is solved by MoVIT, and it can be interpreted as a factorization of the model 213 presented in<sup>14</sup> and<sup>15</sup>, in which the DVFs are used to directly modify the projection matrix 214 W instead of adding the extra factor  $M_i$ . The new system has the same number of equa-215 tions as Eq. (2), but the number of unknowns is reduced to only the number of voxels of the 216 chosen time frame. 217

#### <sup>218</sup> II.E. Solver

There are many choices of Solvers for the system Eq. (10). These different choices can enforce different types of constraints on the solution. In<sup>11,13</sup>, the Primal Dual Hybrid Gradient method of<sup>24</sup> is used to enforce sparsity of the gradient of the solution. MoVIT uses a modified version of SIRT with an intuitive interpretation. This can be seen as a gradient descent based algorithm, as SIRT is a slightly altered version of gradient descent<sup>25</sup>. We chose to use a basic (projected) gradient descent to minimize the objective function

$$F_i(\boldsymbol{x}_i) = \|\boldsymbol{W}\boldsymbol{M}_i\boldsymbol{x}_i - \boldsymbol{p}\|_2^2 \quad . \tag{11}$$

The gradient descent update step for Eq. (11) is

$$\boldsymbol{x}^{k+1} = \boldsymbol{x}^k + \gamma_k \boldsymbol{M}_i^T \boldsymbol{W}^T (\boldsymbol{p} - \boldsymbol{W} \boldsymbol{M}_i \boldsymbol{x}^k)$$
(12)

$$= \boldsymbol{x}^{k} + \gamma_{k} \sum_{j=1}^{n} \boldsymbol{M}_{ij}^{T} \boldsymbol{W}_{j}^{T} (\boldsymbol{p}_{j} - \boldsymbol{W}_{j} \boldsymbol{M}_{ij} \boldsymbol{x}_{i}^{k}) \quad .$$
(13)

Eq. (13) shows explicitly how the adjoint image warping operators  $\boldsymbol{M}_{ij}^{T}$  are used.

#### <sup>227</sup> II.F. Experiments

To evaluate the validity of the proposed algorithm for computing adjoint image warps, three experiments were performed involving the solution of system Eq. (10).

The first experiment is a simulation experiment that investigates the influence of the magnitude of the DVF, in the case that the DVF is known.

232 2. The second experiment is a simulation experiment with unknown DVF. This experi 233 ment investigated the influence of noise in a realistic setting, including the estimation
 234 of the DVF.

3. The last experiment is an application the experimental data obtained at a synchrotron
 facility. This experiments evaluates the the influence of the time step between subscans,
 by selecting subscans which are further and further apart.

In these experiments, our method was compared against two alternative methods to compute
(approximate) adjoint image warps.

The first method substitutes the adjoint warp by a regular image warp along an inverted
 DVF computed by the fixed point algorithm of<sup>26</sup>. This is the approach taken in the
 MoVIT algorithm and in<sup>14</sup>. For this method, a convergence criterion for the inversion
 algorithm needs to be chosen, and twice the amount of DVFs need to be stored (for
 each DVF also its inverse).

245 2. A simpler and computationally less expensive approach is to substitute the adjoint
warp by a regular warp along the negative DVF. A simple sign change is inexpensive
to compute and no extra storage is required because we can compute it on the fly. This

method is valid when the DVF is interpreted as the derivative of the voxel position with respect to time. Changing the sign of this derivative is equivalent to an inversion of the motion with respect to time. In practice, this is only an approximation because the DVFs are not exactly derivatives. Instead, they contain the change of a voxel position with respect to a discrete time step. This method is thus reliant on small time steps.

The quality of the reconstructions was evaluated using the Mean Squared Error (MSE) and the Structural Similarity Index Measure (SSIM)<sup>27</sup>. All forward and backward CT projections were performed with the GPU routines of the ASTRA toolbox, which use Joseph's method<sup>28,29,30</sup>.

#### <sup>258</sup> II.F.1. Simulation experiment with known DVF

To investigate the effect of the magnitude of the DVF on the adjoint warping techniques, 259 a simulation experiment was performed using a 2D phantom of a lung at time 0, and a 2D 260 DVF to simulate motion on it (Fig. 2). The moved phantom is then regarded as ground 261 truth at time 1. This phantom is one of the reconstructions presented  $in^{31}$ , which we use 262 as ground truth. The DVF was obtained by estimating the optical flow between the ground 263 truth and the next time frame presented in<sup>31</sup>, and is also used as ground truth. To simulate 264 the motion we used a bilinear warp, in contrast to the bicubic warp used in reconstruction, 265 to not use the same model in the synthesization as in the inversion. At both time frames, a 266 parallel beam CT scan was simulated with 128 projections at golden ratio angles. Gaussian 267 noise with a standard deviation of 0.002 was added to the projection data, after rescaling it 268 to the interval [0, 1]. The image of time 0 was then reconstructed using the projection data of 269 time 0 and time 1, by solving system Eq. (10). This experiment was repeated several times, 270 each time with a different scalar multiplier applied to the DVF. We used 16 scalar multipliers 271 from the range [0.5, 3] and each experiment was performed with 10 noise realizations. 272

#### <sup>273</sup> II.F.2. Simulation experiment with unknown DVF

A numerical study was performed on a cylindrical phantom containing growing spheres, mimicking the formation process of foam<sup>32</sup>. A 2D+time phantom (Fig. 3) with two materials



(a) Ground truth image. (b) Ground truth DVF at scale 1.



was obtained by extracting a single slice of the 4D phantom, and pixelating it on a  $1024 \times 1024$ 276 grid. The width and height of the phantom is 1 cm. The first material is liquid with a 277 constant attenuation coefficient of  $0.8069 \,\mathrm{cm}^{-1}$ . The second material is air with a constant 278 attenuation coefficient of  $0.9529 \times 10^{-3} \,\mathrm{cm}^{-1}$ . A dynamic (2D+time) CT scan was simulated 279 with three subscans recording a part of the foam formation process. Each subscan consists 280 of 512 parallel beam projections with golden ratio angles. These projections where simulated 281 with various noise levels using the on a detector with 1024 detector elements. The noise is 282 simulated Poisson noise corresponding to 7 beam intensities in the range  $10^3$  to  $10^5$  (photon 283 count). Each noise level was evaluated with 10 different realizations. To evaluate the quality 284 of the reconstructions, they were compared against the ground truth in the middle of the 285 subscan. 286

For each of the generated datasets, we reconstructed subscan 2, combining its data with the data of subscan 1 and 3 using system Eq. (10). The estimated DVFs used in this system are shown in Fig. 4.



(a) Subscan 1

Figure 3: The ground truth at the middle time point in each of the subscans.



(a) DVF between sub- (b) DVF between subscan 1 and subscan 2scan 2 and subscan 3

Figure 4: The ground truth at the middle time point of subscan 2, with the DVFs mapping it to the other subscans overlayed on top of it. For a clearer visualization, we only display each 10-th vector in both the horizontal and vertical direction.

#### II.F.3. Liquid foam dataset 290

To evaluate our method on real experimental data, an experiment was performed with a 291 4D-CT scan of a liquid foam that is pushed through a funnel, available from Tomobank<sup>33</sup>. 292 This 4D-CT scan was recorded at the TOMCAT beamline of the Swiss Light Source. The 293 data consists of 180 subscans. Each of the subscans has 300 parallel beam projections taken 294 uniformly over a 0 to 180 degree range with a rotational speed of 840 deg/s. The projections 295 were recorded at an energy level of 16 keV, with an exposure time of 0.7 ms. The detector 296 consists of  $1800 \times 2016$  detector pixels of size 3  $\mu$ m. For our experiments, the projections were 297 vertically cropped and downsampled by a factor of 4, resulting in  $128 \times 504$  pixel projections 298 with a virtual pixel size of  $12 \,\mu m$ . 299

311

The experiment was focussed on the reconstruction of subscan 98 since around this sub-300 scan, some challenging motion happens. The subscan was first reconstructed using its 300 301 projections and with a smoothed TV1 regularization term<sup>34</sup> added to the objective function 302 of Eq. (11), to get a high quality reconstruction to be used as a reference. The reconstruction 303 is shown in Fig. 5. Next, we evaluated our adjoint image warping method by reconstruct-304 ing subscan 98, using only the 150 projections with even index, but combining it with 150 305 projections of a second subscan with odd index, using system Eq. (10). We repeated this 306 experiment several times with different choices for the second subscan ranging from 89 to 307 107, to investigate how far away in time we can go, and still benefit from the added data. 308 We paid special attention to a small region of interest shown in Fig. 6. Inside this region of 309 interest, a lot of complex motion occurs, while the rest of the volume hardly moves. 310



Figure 5: Three orthogonal slices of the reference reconstruction of subscan 98. The region of interest with large complex motion is marked by the red rectangles. (see Fig. 6)



Figure 6: A visualization of the norm of the optical flow between frame 95 and frame 98. Three orthogonal slices are shown, one for each dimension. The red rectangle marks a region of interest where the flow is large and complex, which complicates the inversion of the DVF.

### 312 III. Results

#### <sup>313</sup> III.A. Simulation experiment with known DVF

The MSE and SSIM of the reconstructions from this simulation experiment, averaged over 10 noise realizations, are displayed in Fig. 7a and 7b, respectively. The horizontal axis shows the scalar multiplier applied to the DVF. In Fig. 9 the reconstructions are shown for three of those scalar multipliers (1, 2 and 3). In Fig. 8, the convergence rates of the different methods are visualized, at DVF scale 3.

#### <sup>319</sup> III.B. Simulation experiment with unknown DVF

The MSE and SSIM of the reconstructions from this second simulation experiment, averaged over 10 noise realizations, are displayed in Fig. 10a and 10b, respectively. The reconstructions at noise level 10<sup>5</sup> are shown in Fig. 12, next to the ground truth. In that same figure, a zoomed in version of the reconstructions is shown that highlights a bubble with artifacts. In Fig. 11, the convergence rates of the different methods are visualized, at photon count 10<sup>5</sup>.

#### 325 III.C. Liquid foam dataset

The MSE and SSIM, with respect to the reference reconstruction, of all reconstructions of subscan 98 is shown in Fig. 13. The black line indicates the MSE and SSIM of a reconstruction using only 150 projections of subscan 98, in contrast to the other reconstructions that also include 150 projections of a second subscan. Obtaining an MSE lower than the black line, or an SSIM higher than the black line, indicates that the information of the extra subscan was exploited to improve the reconstruction quality.

Fig. 14 shows the MSE and SSIM of our reconstructions in the region of interest shown in Fig. 6, using the data of subscans 89 to 97. This region is of interest because during the recording of subscan 89 to 97, the motion is concentrated in this region. A slice of the reconstructions of subscan 98 using the data of subscan 95 is shown in Fig. 15. The region of interest is marked and magnified in this figure.



Figure 7: MSE (7a) and SSIM (7b) of the reconstructions of the lung phantom with differently scaled DVFs (see II.F.1.)



Figure 8: Convergence plots comparing the different methods at scale 3.





 $Figure \ 9:$  Reconstructions of the lung phantom with different methods and differently scaled DVFs

Scale 1



(b)

Figure 10: MSE (10a) and SSIM (10b) of the reconstructions of subscan 2 of the simulated scan, using the data of subscan 1 and 3, at different noise levels. (see II.F.2.)



Figure 11: Convergence plots comparing the different methods at photon count  $10^5$ .



(a) Negative DVF (b) Inverse DVF (c) Exact adjoint (d) Reference

Figure 12: Reconstructions of subscan 2 of the simulated scan, using the data of subscan 1 and 3, with a photon count of  $10^5$ , and different methods for the adjoint warps. The second row shows a zoomed in image of a bubble with artifacts.



(a)





Figure 13: MSE (13a) and SSIM (13b) of the reconstruction of subscan 98, using 150 projections + 150 projections of one extra subscan. The black line shows the MSE and SSIM of a reconstruction with 150 projections, without extra projections from an other subscan. It can be used as a reference to evaluate whether the information of the extra subscan was able to provide an improved reconstruction quality.



Figure 14: MSE (14a) and SSIM (14b) of the region of interest (Fig. 6) of subscan 98, using 150 projections + 150 projections of one extra subscan. The black line shows the MSE and SSIM of a reconstruction with 150 projections, without extra projections from an other subscan. It can be used as a reference to evaluate whether the information of the extra subscan was able to provide an improved reconstruction quality.



Figure 15: A horizontal slice of the reconstruction of subscan 98 using data of subscan 95 with 3 different methods, with the region of interest (Fig. 6) magnified. The bottom row shows absolute difference images of the magnified region, with respect to the reference.

### <sup>337</sup> IV. Discussion

<sup>338</sup> When comparing the three methods, it is important to keep in mind that using the exact <sup>339</sup> adjoint and using the negative DVF have almost equal computational demands, while using <sup>340</sup> the inverse DVF requires an iterative inversion of the DVFs, and the storage of these inverted <sup>341</sup> DVFs. For instance, if a  $500 \times 500 \times 500$  volume is used, then a DVF stored in single precision <sup>342</sup> floating point numbers requires 1.5 GB of memory. Having to store twice the amount of DVFs <sup>343</sup> can easily lead to a lack of RAM of GPU memory. The inversion algorithm also requires the <sup>344</sup> choice of a convergence criterion, which further complicates the 4D-CT algorithm.

Our experiments show how our method compares to two alternatives when used in a basic 4D-CT reconstruction algorithm. However, our method is applicable to various other motion compensated reconstruction techniques that make use of adjoint image warping, including the methods of classes 2, 3 and 4 discussed in the introduction, which leaves much room for further exploration.

#### <sup>350</sup> IV.A. Simulation experiment with known DVF

The measurments in Fig. 7 show that using the exact adjoint robust to the scale of the DVF, 351 while the approximation methods lead to a degrades quality if the DVF gets too large. The 352 quality decays the fastest in the case of the negative DVF, which is to be expected as this 353 method is derived under the assumption that all changes are infinitesimal. The fixpoint 354 inverted DVF provides a much better approximation and only shows a small degradation in 355 quality for large DVFs. When the DVF is scaled to a small magnitude, the three methods 356 come very close together, and the fixpoint inverted DVF sometimes gives better results that 357 the exact adjoint. This can possibly be explained by the fact that an accurate inversion can 358 be found in the case of a very small DVF, combined with the fact that the incorrect error 359 redistribution in the update of the image has a smoothing effect, which can remove noise 360 and other artifacts. The convergence plot in Fig. 8 shows that, in the case of a large DVF, 361 the method using the negative DVF is not capable of minimizing the residue after a certain 362 point. The other methods keep lowering the residue, and the lowest residue is obtained using 363 the exact DVF. 364

The reconstructions in Fig. 9 show what kind of artifacts are caused by the different

approximations. At scale 2, the negative DVF method shows distortions in wide regions around interfaces between different materials. The fixpoint inverted DVF method causes tiny speckles in similar locations. These distortions remain present at smaller scales, although to a lesser extent. The exact adjoint avoids both of these artifacts, at a cheaper computational cost than the fixpoint inverted DVF method. However, in practice, there will be other artifacts present due to the fact that the DVF is not exactly known. This is further discussed in the following experiments.

#### <sup>373</sup> IV.B. Simulation experiment with unknown DVF

Fig. 10 shows that, in terms of MSE, using the exact adjoint provides the best results, followed 374 by using the negative DVF as an approximation, and then the inverse DVF. This holds for 375 all considered noise levels. In terms of SSIM, the method with exact adjoints provides the 376 best results for noise levels with photon count  $10^4$  and above. Below this photon count, the 377 quality drops faster than the other methods, indicating that the method is more sensitive 378 to noise. Similar to the previous experiment, this might be explained by the fact that the 379 approximating methods cause a smoothing effect which suppresses noise. If that is the case, a 380 possible improvement to our method would be to include some kind of explicit regularization 381 in the method. It was observed that the method using the exact adjoint is capable of lowering 382 the residue faster than the other methods (Fig. 11), however, this seems to depend on the 383 situation as it was not observed in the previous experiment. Fig. 11c shows the evolution of 384 the MSE with respect to the iteration number. It shows that overfitting to the noise and 385 the DVF estimation errors occurs around iteration 200. At this iteration, the method using 386 the exact adjoint provides the lowest MSE and residual error of the three methods. 387

In Fig. 12, the effects of the different methods on the reconstruction can be seen visually. 388 The reconstruction with exact adjoints is more sharp and granular, while the two approxi-389 mating methods are smooth. The approximations do cause an artifact around a fast growing 390 bubble. This is due to the fact that the DVF is not invertible in this area, so methods relying 391 on an approximate inversion of the DVF fail in such locations. Our proposed algorithm for 392 exact adjoint does not require an approximate inverse of the DVF. Still, the reconstruction 393 with the exact adjoints shows some artifacts which can not be seen in the reconstructions 394 with the other methods. This might be caused by the inaccuracies present in the estimated 395

<sup>396</sup> DVF. Using the exact adjoint likely results in a reconstruction which is more faithful to the <sup>397</sup> inaccurate DVF. Therefore, the decision of which method to use in practice must be made <sup>398</sup> with the accuracy of the estimated DVF in mind. Possible improved methods could be made <sup>399</sup> by combining the different methods based on the DVF accuracy.

### 400 IV.C. Liquid foam dataset

The results in Fig. 13 show that we can successfully exploit the data of many other subscans to improve the reconstruction quality of subscan 98. It can be observed that the improvement in reconstruction quality gets smaller when using subscans that are further away in time. This holds for all three methods. There are some outliers that do not follow this pattern: subscan 94 and subscan 99 give a substantially lower reconstruction quality. This can be attributed to the fact that there is a substantial amount of motion during these subscans, so there is a substantial amount of inconsistency in their data.

As shown in Fig. 13, the difference between the MSE and SSIM of the three methods is 408 small. Using the negative DVF yields the lowest reconstruction quality overall. In terms of 409 SSIM, the method using the inverse DVF is almost indistinguishable from the method using 410 the exact adjoints. For select subscans (99, 105, 106, 107) there is a very small advantage 411 to the exact adjoints. In terms of MSE, using the inverse DVF often gives a slightly higher 412 quality than using the exact adjoint. It should be noted that these errors were measured 413 over a large volume of which only a small part shows complex motion, as shown in Fig. 6. 414 The lower MSE for the approximation using the inverse DVF might be explained as follows: 415 in most of the volume, the DVF is very small, so an accurate inversion can be found. This 416 explains why there is no substantial loss of quality using this method. On top of that, the 417 approximation introduces a smoothing effect, which can remove detail, but it also removes 418 noise and reduces some artifacts. In this experiment, no notable difference in convergence 419 speed was observed between the different methods. 420

The error measurements in Fig. 14 show how the methods compare in the region with the largest motion. This paints a different picture than the error measurements of the full volume. A first observation is that only the two closest subscans can be used to give an improved reconstruction quality in this region. A second observation is that the difference between the three methods becomes more apparent. In this region, using the exact adjoints yields the highest MSE and SSIM overall, and the difference is more pronounced for largertime steps.

Fig. 15 shows a slice of the reconstructions of subscan 98 with subscan 95 as the extra subscan. When zooming in on the region of interest, some double edge artifacts become apparent. These artifacts are present in all three methods, but they are most prominent when using the negative DVF approximation. They are less prominent in the reconstructions of the exact adjoint method and the method with the inverse DVF approximation. of these two methods, the exact adjoint method is computationally the cheapest.

### 434 V. Conclusions

In this work, a GPU accelerated algorithm that computes the exact adjoint action of multi-435 variate spline based image warping was introduced. It was shown how this adjoint operator 436 can be applied to reconstruction problems in 4D-CT that rely on optical flow techniques. 437 Our method was experimentally compared against two alternative methods that approxi-438 mate the adjoint image warp by a regular warp along an approximate inverse of the optical 439 flow. The experiments showed that our method can improve reconstruction quality when 440 the flow is difficult to invert. Moreover, an accurate inversion of a flow field requires an 441 iterative algorithm with a stopping criterion, as well as the storage of the resulting inverse 442 flow. Because our method does not require the inversion of the flow, we avoid this tunable 443 parameter and the memory costs of storing the inverted flow. 444

### 445 VI. Acknowledgements

This research is funded by the Fund for Scientific Research-Flanders (FWO) grant nr.
S007219N and grant nr. 1SA2920N.

### 448 VII. Conflict of Interest Statement

<sup>449</sup> The authors have no relevant conflicts of interest to disclose.

## $^{451}_{450}$ References

- <sup>452</sup> <sup>1</sup> B. K. P. Horn and B. G. Schunck, Determining optical flow, Artificial Intelligence 17,
  <sup>453</sup> 185–203 (1981).
- <sup>454</sup> <sup>2</sup> B. D. Lucas and T. Kanade, An Iterative Image Registration Technique with an Application to Stereo Vision, in *Proceedings of the 7th International Joint Conference on* <sup>456</sup> Artificial Intelligence Volume 2, IJCAI'81, page 674–679, San Francisco, CA, USA,
   <sup>457</sup> 1981, Morgan Kaufmann Publishers Inc.
- <sup>3</sup> C. Zach, T. Pock, and H. Bischof, A duality based approach for realtime TV-L1 optical
   flow, in *Joint Pattern Recognition Symposium*, pages 214–223, Springer, 2007.
- <sup>460</sup> <sup>4</sup> D. Rueckert, L. I. Sonoda, C. Hayes, D. L. G. Hill, M. O. Leach, and D. J. Hawkes,
   <sup>461</sup> Nonrigid registration using free-form deformations: application to breast MR images,
   <sup>462</sup> IEEE Transactions on Medical Imaging 18, 712–721 (1999).
- <sup>5</sup> G. Wolberg, *Digital Image Warping*, volume 10662, IEEE Computer Society Press Los
   Alamitos, CA, 1990.
- <sup>465</sup> <sup>6</sup> M. Brehm, P. Paysan, M. Oelhafen, and M. Kachelrieß, Artifact-resistant motion estimation with a patient-specific artifact model for motion-compensated cone-beam CT,
  <sup>467</sup> Medical physics 40, 101913 (2013).
- <sup>468</sup> <sup>7</sup> C. Mory, G. Janssens, and S. Rit, Motion-aware temporal regularization for improved
   <sup>469</sup> 4D cone-beam computed tomography, Physics in Medicine & Biology **61**, 6856 (2016).
- <sup>470</sup> <sup>8</sup> J. Wang and X. Gu, Simultaneous motion estimation and image reconstruction (SMEIR)
  <sup>471</sup> for 4D cone-beam CT, Medical physics 40, 101912 (2013).
- <sup>472</sup> <sup>9</sup> Q. Tang, J. Cammin, S. Srivastava, and K. Taguchi, A fully four-dimensional, iterative
  <sup>473</sup> motion estimation and compensation method for cardiac CT, Medical physics **39**, 4291–
  <sup>474</sup> 4305 (2012).
- <sup>475</sup> <sup>10</sup> V. Van Nieuwenhove, J. De Beenhouwer, J. Vlassenbroeck, M. Brennan, and J. Sijbers,
  <sup>476</sup> MoVIT: a tomographic reconstruction framework for 4D-CT, Optics Express 25, 19236–
  <sup>477</sup> 19250 (2017).

- page 27
- <sup>478</sup> <sup>11</sup> G. Zang, R. Idoughi, R. Tao, G. Lubineau, P. Wonka, and W. Heidrich, Warp-and<sup>479</sup> Project Tomography for Rapidly Deforming Objects, ACM Transactions on Graphics
  <sup>480</sup> (TOG) 38, 86 (2019).
- <sup>481</sup> <sup>12</sup> M. Burger, H. Dirks, L. Frerking, A. Hauptmann, T. Helin, and S. Siltanen, A variational
   <sup>482</sup> reconstruction method for undersampled dynamic X-ray tomography based on physical
   <sup>483</sup> motion models, Inverse Problems **33**, 124008 (2017).
- <sup>484</sup> <sup>13</sup> G. Zang, R. Idouchi, R. Tao, G. Lubineau, P. Wonka, and W. Heidrich, Space-time
  tomography for continuously deforming objects, ACM Transactions on Graphics (TOG)
  <sup>486</sup> **37**, 100 (2018).
- <sup>14</sup> M. Odstrcil, M. Holler, J. Raabe, A. Sepe, X. Sheng, S. Vignolini, C. G. Schroer, and
   M. Guizar-Sicairos, Ab initio nonrigid X-ray nanotomography, Nature Communications
   <sup>489</sup> 10, 2600 (2019).
- <sup>490</sup><sup>15</sup> K. Ruymbeek and W. Vanroose, Algorithm for the reconstruction of dynamic objects
   <sup>491</sup> in CT-scanning using optical flow, Journal of Computational and Applied Mathematics
   <sup>492</sup> 367, 112459 (2020).
- <sup>493</sup><sup>16</sup> T. Feng, J. Wang, G. Fung, and B. Tsui, Non-rigid dual respiratory and cardiac motion
  <sup>494</sup> correction methods after, during, and before image reconstruction for 4D cardiac PET,
  <sup>495</sup> Physics in Medicine & Biology **61**, 151 (2015).
- <sup>496</sup> <sup>17</sup> C. Mory and S. Rit, Improving iterative 4D CBCT through the use of motion informa<sup>497</sup> tion, Proc. of Fully 3D 2015 (Newport, Rhode Island, USA,) (2015).
- <sup>498</sup> <sup>18</sup> F. Lekien and J. Marsden, Tricubic interpolation in three dimensions, International
   <sup>499</sup> Journal for Numerical Methods in Engineering **63**, 455–471 (2005).
- <sup>500</sup> <sup>19</sup> A. W. Paeth, *Graphics Gems V*, Academic Press, Inc., USA, 1995.
- S. Van der Walt, J. L. Schönberger, J. Nunez-Iglesias, F. Boulogne, J. D. Warner,
  N. Yager, E. Gouillart, and T. Yu, scikit-image: image processing in Python, PeerJ 2,
  e453 (2014).
- <sup>21</sup> The Sage Developers, SageMath, the Sage Mathematics Software System (Version 9.1),
   <sup>505</sup> 2020, https://www.sagemath.org.

- <sup>22</sup> J. Renders, *Adjoint Image Warping*, 2020, https://doi.org/10.5281/zenodo.4288715.
- J. S. Pérez, E. Meinhardt-Llopis, and G. Facciolo, TV-L1 optical flow estimation, Image
   Processing On Line 2013, 137–150 (2013).
- A. Chambolle and T. Pock, A first-order primal-dual algorithm for convex problems
   with applications to imaging, Journal of Mathematical Imaging and Vision 40, 120–145
   (2011).
- <sup>25</sup> P. C. Hansen, *Discrete inverse problems: insight and algorithms*, volume 7, chapter 6,
  <sup>513</sup> pages 111–112, Siam, 2010.
- <sup>514</sup> <sup>26</sup> M. Chen, W. Lu, Q. Chen, K. J. Ruchala, and G. H. Olivera, A simple fixed-point <sup>515</sup> approach to invert a deformation field, Medical Physics **35**, 81–88 (2008).
- <sup>27</sup> Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, Image quality assessment:
  from error visibility to structural similarity, IEEE Transactions on Image Processing 13,
  <sup>518</sup> 600–612 (2004).
- <sup>28</sup> 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
  <sup>521</sup> using the ASTRA toolbox, Optics Express 24, 25129–25147 (2016).
- <sup>522</sup><sup>29</sup> W. van Aarle, W. J. Palenstijn, J. De Beenhouwer, T. Altantzis, S. Bals, K. J. Batenburg, <sup>523</sup> and J. Sijbers, The ASTRA Toolbox: A platform for advanced algorithm development <sup>524</sup> in electron tomography, Ultramicroscopy **157**, 35–47 (2015).
- <sup>30</sup> W. J. Palenstijn, K. J. Batenburg, and J. Sijbers, Performance improvements for iterative electron tomography reconstruction using graphics processing units (GPUs), Journal of Structural Biology **176**, 250–253 (2011).
- <sup>31</sup> R. Werner, C. Hofmann, E. Mücke, and T. Gauer, Reduction of breathing irregularity related motion artifacts in low-pitch spiral 4D CT by optimized projection binning,
   Radiation Oncology 12, 100 (2017).
- <sup>32</sup> D. Pelt, K. J. Batenburg, and J. Sethian, Improving tomographic reconstruction from
   limited data using mixed-scale dense convolutional neural networks, Journal of Imaging
   4, 128 (2018).

- <sup>33</sup> F. De Carlo, D. Gürsoy, D. J. Ching, K. J. Batenburg, W. Ludwig, L. Mancini,
  F. Marone, R. Mokso, D. M. Pelt, J. Sijbers, and M. Rivers, TomoBank: a tomographic
  data repository for computational X-ray science, Measurement Science and Technology
  29, 034004 (2018).
- <sup>34</sup> V. Haase, K. Stierstorfer, K. Hahn, H. Schöndube, A. Maier, and F. Noo, Exploring the
  space between smoothed and non-smooth total variation for 3D iterative CT reconstruction, in *Medical Imaging 2019: Physics of Medical Imaging*, edited by T. G. Schmidt,
  G.-H. Chen, and H. Bosmans, volume 10948, pages 511 517, International Society for
  Optics and Photonics, SPIE, 2019.