

Adjoint image warping using multivariate splines with application to four-dimensional computed tomography

Jens Renders^{a)}, Jan Sijbers and Jan De Beenhouwer
imec-Vision Lab, University of Antwerp, Universiteitsplein 1, Antwerp 2610, Belgium

(Received 5 October 2020; revised 28 January 2021; accepted for publication 8 February 2021; published 18 August 2021)

Purpose: Adjoint image warping is an important tool to solve image reconstruction problems that warp the unknown image in the forward model. This includes four-dimensional computed tomography (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 requires 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 alternatives.

Methods: Using symbolic computer algebra, we computed a list of 64 polynomials that allow us to compute a matrix representation of trivariate cubic image warping. By combining an on-the-fly computation of this matrix with a parallelized implementation of columnwise matrix multiplication, we obtained an efficient, low memory implementation of the adjoint action of 3D cubic image warping. We used this operator in the solution of a previously proposed 4D-CT reconstruction model in which the image of a single subscan was compared against projection data of multiple subscans by warping and then projecting the image. We compared the properties of our exact adjoint with those of approximate adjoints by warping along inverted motion.

Results: Our method requires halve the memory to store motion between subscans, compared to methods that need to compute and store an approximate inverse of the motion. It also avoids the computation time to invert the motion and the tunable parameter of the number of iterations used to perform this inversion. Yet, a similar and often better reconstruction quality was obtained in comparison with these more expensive methods, especially when the motion is large. When compared against a simpler method that is similar to ours in computational demands, our method achieves a higher reconstruction quality in general.

Conclusions: Our implementation of the exact adjoint of cubic image warping improves efficiency and provides accurate reconstructions. © 2021 The Authors. *Medical Physics* published by Wiley Periodicals LLC on behalf of American Association of Physicists in Medicine [https://doi.org/10.1002/mp.14765]

Key words: 4D-CT, image warping, optical flow, tomography

1. INTRODUCTION

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

$$W_i x_i = p_i, \quad i = 1, \dots, n, \quad (1)$$

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

$$\begin{bmatrix} W_1 & 0 & 0 & 0 \\ 0 & W_2 & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & W_n \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} p_1 \\ p_2 \\ \vdots \\ p_n \end{bmatrix}. \quad (2)$$

One way to alleviate this underdetermination is to link the time frames using image registration or optical flow techniques^{1–4} and image warping.⁵ 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:

1. The motion compensated reconstruction techniques employed in Refs. [6–9] 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 algorithm¹⁰ and in Ref. [11] 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 Ref. [10] that this iterative approach outperforms the averaging of separate reconstructions in terms of reconstruction quality.
3. In Refs. [12,13], Eq. (2) is regularized with terms that constrain the change of the object between frames, that is, with constraints on the optical flow. In Ref. [13], 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.
4. In Refs. [14,15], 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 Ref. [16].

The above mentioned methods implement the adjoint image warping operators by warping along an approximated inverse of the flow, or they are restricted to very small examples where they can work with matrix representations of the operators and their transpose. Computing the inverse of the flow requires computation time and memory. On top of that, since the flow is generally not exactly invertible and the adjoint of a warp is not exactly the warp along the inverse flow, it introduces inaccuracies. Few papers have investigated how these approximations compare with the exact adjoints. In Ref. [17], a pair of adjoint warping operators with a custom linear interpolation method is used for respiratory and cardiac motion correction in 4D PET. It is shown that using inverse warps as an approximation for the adjoint warp leads to image degradation compared to using the exact adjoint warps.

In this work, an efficient, GPU-based algorithm that exactly computes the adjoint action 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.

2. MATERIALS AND METHODS

2.A. Image warping operators

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

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

There are two different approaches to implement image warping operators, referred to as forward and backward warping⁵ (see Fig. 1). Assume we have two images, x_a and x_b and a DVF describing the flow from x_a to x_b . With forward image warping, the voxels of x_a are first repositioned along the DVF to obtain non-grid data representing the warped version of x_a , and this non-grid data is then resampled at grid points to get an image similar to x_b . Backward image warping is another approach, in which the DVF is followed in the opposite direction. For each voxel, we look at the position it is sent to by the DVF and interpolate the regular grid data of x_b at that point. The result is an image similar to x_a .

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 in Refs. [10,11]. Adjoint warping operators for the less complicated trilinear, bicubic, and bilinear interpolation methods were also implemented.

2.B. Tricubic interpolation

Let $f: \mathbb{Z}^3 \rightarrow \mathbb{R}$ be a function with values at only integer coordinates. Tricubic interpolation extends such a function to a piecewise polynomial function $\hat{f}: \mathbb{R}^3 \rightarrow \mathbb{R}$ which agrees with f on integer coordinates and is differentiable in every

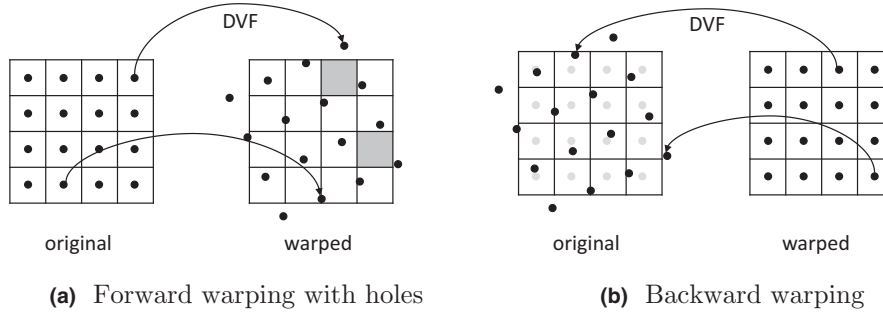


FIG. 1. Forward and backward image warping.

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. All other cubes can be interpolated by shifting. The 64 coefficients a_{lmn} can be obtained by demanding that \hat{f} agrees with f , and by putting constraints on the differentiability of \hat{f} . Once 64 independent linear constraints have been made, the coefficients are determined.¹⁸ Another common way of computing \hat{f} is via a composition of successive cubic one-dimensional (1D) interpolations, using, for example, Catmull–Rom splines (see, e.g., Refs. [19,20]). By computing the composition numerically, the coefficients a_{lmn} are never explicitly needed, but they can be determined by performing the composition symbolically. In both cases, the coefficients a_{lmn} are linear combinations of 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

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

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

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

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

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

with $\forall l, n, m : \beta_{ijk}^{lmn} \in \mathbb{R}$. 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 SageMath²¹ (the code is available here²²). 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 .

2.C. Adjoint image warping with tricubic interpolation

A 3D image can be represented by a function $f: \mathbb{Z}^3 \rightarrow \mathbb{R}$ which is zero outside the image bounds, or it can be represented by a column vector $\mathbf{f} \in \mathbb{R}^N, N = \text{width} \times \text{height} \times \text{depth}$, which lists all voxel values in a predetermined order. The first representation allows to use tricubic interpolation on the image, while the second representation describes an image warping operator as a linear map $\mathbf{M}: \mathbb{R}^N \rightarrow \mathbb{R}^N$, which can be represented as an $N \times N$ matrix. The equation $\mathbf{g} = \mathbf{M} \mathbf{f}$ expresses the warping of image \mathbf{f} with resulting image \mathbf{g} . Each voxel value of \mathbf{g} is obtained by taking a linear combination of 64 voxel values of \mathbf{f} , with weights given by the coefficients b_{ijk} of Eq. (5). These weights make up the 64 non-zero entries of one row of matrix \mathbf{M} , and \mathbf{M} has one such row for each voxel of \mathbf{f} . Now that a rowwise representation of \mathbf{M} is known, a columnwise representation of \mathbf{M}^T can be obtained by transposing: $\mathbf{M}^T = (\mathbf{m}_1^T, \dots, \mathbf{m}_N^T)$, where \mathbf{m}_i is the i -th row of \mathbf{M} . An adjoint image warping operator can then be implemented as a columnwise matrix multiplication with \mathbf{M}^T :

$$\mathbf{M}^T \mathbf{g} = \sum_{i=1}^N \mathbf{g}_i \mathbf{m}_i^T \quad (7)$$

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

Algorithm 1: Adjoint tricubic image warping

input : Image \mathbf{g} , DVF

output: Adjoint warped image \mathbf{f}

```

1  $\mathbf{f} = \mathbf{0}$ 
2 foreach voxel position  $(p_1, p_2, p_3)$  do
3    $(q_1, q_2, q_3) = (p_1, p_2, p_3) + \text{DVF}(p_1, p_2, p_3)$ 
4    $(i', j', k') = ([q_1], [q_2], [q_3])$ 
5   for  $(i, j, k) \in \{-1, 0, 1, 2\}^3$  do
6      $\mathbf{f}(i', j', k') += b_{ijk}(i' - q_1, j' - q_2, k' - q_3) \mathbf{g}(p_1, p_2, p_3)$ 
7   end
8 end

```

When working with large 3D volumes, it is important to leverage the performance gains of parallelization. The outer loop of algorithm 1 can be computed in parallel, but special care needs to be taken because of the addition on line 6. The position to which we are adding is dictated by the DVF, and it is possible that multiple threads would write to the

same position, leading to a race condition. By implementing the addition as an atomic addition, race conditions can be avoided and additions are performed sequentially if needed. Because the amount of overlap caused by the DVF pointing multiple times to the same area is in most cases relatively small, the achieved parallelization is almost indistinguishable from a complete parallelization. A GPU-accelerated implementation is available here.²²

2.D. Dynamic tomographic model

We will now describe the dynamic tomographic model used in the MoVIT algorithm.¹⁰ 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, \mathbf{x}_i . In an ideal situation, any time frame \mathbf{x}_j can be produced from \mathbf{x}_i by using a suitable warping operator \mathbf{M}_{ij} , such that $\mathbf{x}_j = \mathbf{M}_{ij}\mathbf{x}_i$ holds for all j . This can be written as:

$$\begin{bmatrix} \mathbf{M}_{i1} \\ \mathbf{M}_{i2} \\ \vdots \\ \mathbf{M}_{in} \end{bmatrix} \mathbf{x}_i = \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \vdots \\ \mathbf{x}_n \end{bmatrix} \quad (8)$$

Substituting Eq. (8) into Eq. (2) yields the dynamic tomographic model:

$$\begin{bmatrix} \mathbf{W}_1 & 0 & 0 & 0 \\ 0 & \mathbf{W}_2 & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & \mathbf{W}_n \end{bmatrix} \begin{bmatrix} \mathbf{M}_{i1} \\ \mathbf{M}_{i2} \\ \vdots \\ \mathbf{M}_{in} \end{bmatrix} \mathbf{x}_i = \begin{bmatrix} \mathbf{p}_1 \\ \mathbf{p}_2 \\ \vdots \\ \mathbf{p}_n \end{bmatrix}, \quad (9)$$

or more concisely:

$$\mathbf{W}\mathbf{M}_i\mathbf{x}_i = \mathbf{p} \quad (10)$$

In practice, Eq. (8) cannot be achieved exactly, but by estimating the DVFs between initial reconstructions, a good approximation can be obtained. There are many image registration and optical flow algorithms available to estimate the DVFs, such as Refs. [1–4]. We used the TV-L1 optical flow algorithm of Ref. [3] implemented along the lines of Ref. [23]. System Eq. (10) is the one that is solved by MoVIT, and it can be interpreted as a factorization of the model presented in which the DVFs are used to directly modify the projection matrix \mathbf{W} instead of adding the extra factor \mathbf{M}_i . The new system has the same number of equations as Eq. (2), but the number of unknowns is reduced to only the number of voxels of the chosen time frame.

2.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 Refs. [11, 13] the Primal Dual Hybrid Gradient method of Ref. [24] 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.²⁵ We chose to use a basic (projected) gradient descent to minimize the objective function

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

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

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

$$= \mathbf{x}^k + \gamma_k \sum_{j=1}^n \mathbf{M}_{ij}^T \mathbf{W}_j^T (\mathbf{p}_j - \mathbf{W}_j \mathbf{M}_{ij} \mathbf{x}_i^k). \quad (13)$$

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

2.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).

1. 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.
2. The second experiment is a simulation experiment with unknown DVF. This experiment investigated the influence of noise in a realistic setting, including the estimation of the DVF.
3. The last experiment is an application to experimental data obtained at a synchrotron facility. This experiment evaluates the influence of the time step between combined subsamples, by selecting subsamples which are further and further apart.

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

1. The first method substitutes the adjoint warp by a regular image warp along an inverted DVF computed by the fixed point algorithm of Ref. [26]. This is the approach taken in the MoVIT algorithm and in Ref. [14]. 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).
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

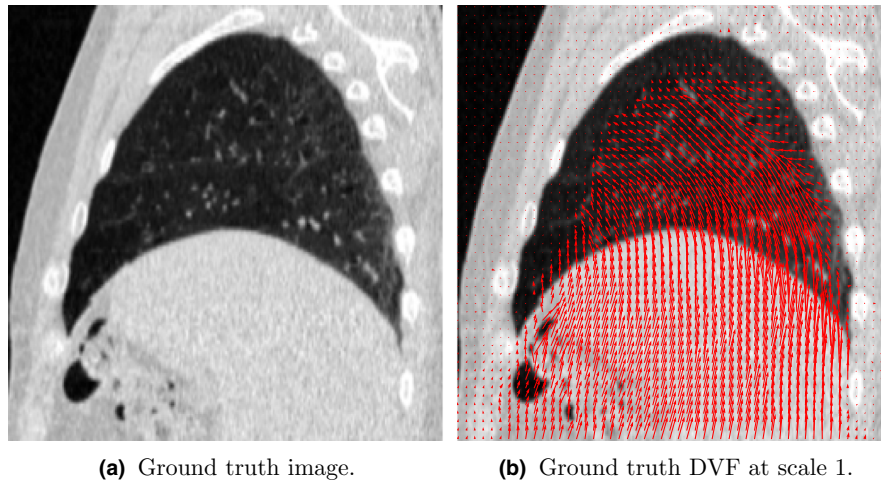


FIG. 2. The ground truth image and deformation vector field (DVF) for the simulation experiment with known DVF.

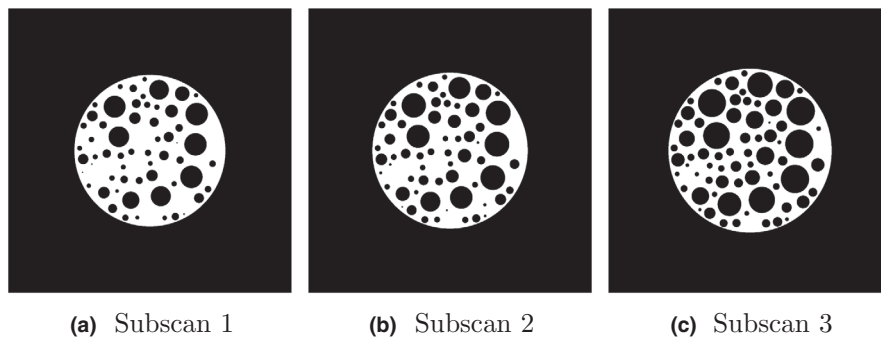


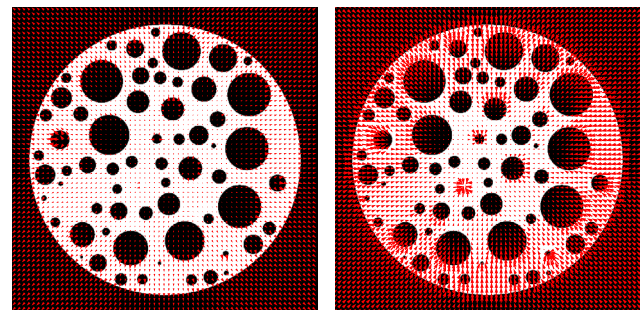
FIG. 3. The ground truth at the middle time point in each of the subscans.

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).²⁷ All forward and backward CT projections were performed with the GPU routines of the ASTRA toolbox, which use Joseph's method.^{28–30}

2.F.1. Simulation experiment with known DVF

To investigate the effect of the magnitude of the DVF on the adjoint warping techniques, a simulation experiment was performed using a 2D phantom of a lung at time 0, and a 2D DVF to simulate motion on it (Fig. 2). The moved phantom is then regarded as ground truth at time 1. This phantom is one of the reconstructions presented in Ref. [31], which we use as ground truth. The DVF was obtained by estimating the optical flow between the ground truth and the next time frame presented in Ref. [31], and is also used as ground truth. To simulate the motion, we used a bilinear warp, in contrast to the bicubic warp used in the reconstruction, to not use the same model in the synthesis as in the inversion. At



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

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

both time frames, a parallel beam CT scan was simulated with 128 projections at golden ratio angles. Gaussian noise with a standard deviation of 0.002 was added to the projection data, after rescaling it to the interval [0,1]. The image of time 0 was then reconstructed using the projection data of time 0 and time 1, by solving system

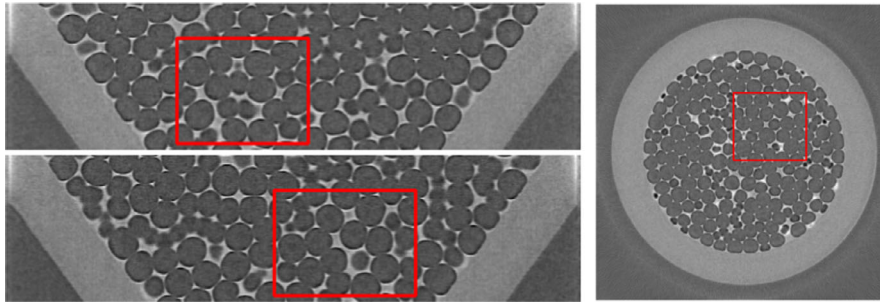


FIG. 5. Three orthogonal slices of the reference reconstruction of subscan 98. The region of interest with large complex motion is marked by the red rectangles (see Fig. 6).

Eq. (10). This experiment was repeated several times, each time with a different scalar multiplier applied to the DVF. We used 16 scalar multipliers from the range $[0.5, 3]$ and each experiment was performed with 10 noise realizations.

2.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.³² A 2D+time phantom (Fig. 3) with two materials was obtained by extracting a single slice of the 4D phantom, and pixelating it on a 1024×1024 grid. The width and height of the phantom is 1 cm. The first material is liquid with a constant attenuation coefficient of 0.8069 cm^{-1} . The second material is air with a constant attenuation coefficient of $0.9529 \times 10^{-3} \text{ cm}^{-1}$. A dynamic (2D+time) CT scan was simulated with three subscans recording a part of the foam formation process. Each subscan consists of 512 parallel beam projections with golden ratio angles. These projections were simulated with various noise levels on a detector with 1024 detector elements. The noise is simulated Poisson noise corresponding to 7 beam intensities in the range 10^3 to 10^5 (photon count). Each noise level was evaluated with 10 different realizations. To evaluate the quality of the reconstructions, they were compared against the ground truth in the middle of the subscan.

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

2.F.3. Liquid foam dataset

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

The experiment was focused on the reconstruction of subscan 98 since around this subscan, some challenging motion happens. The subscan was first reconstructed using its 300 projections and with a smoothed TV1 regularization term³⁴ added to the objective function of Eq. (11), to get a high quality reconstruction to be used as a reference. The reconstruction is shown in Fig. 5. Next, we evaluated our adjoint image warping method by reconstructing subscan 98, using only the 150 projections with even index, but combining it with 150 projections of a second subscan with odd index, using system Eq. (10). We repeated this experiment several times with different choices for the second subscan ranging from 89 to 107, to investigate how far away in time we can go, and still benefit from the added data. We paid special attention to a small region of interest shown in Fig. 6. Inside this region of interest, a lot of complex motion occurs, while the rest of the volume hardly moves.

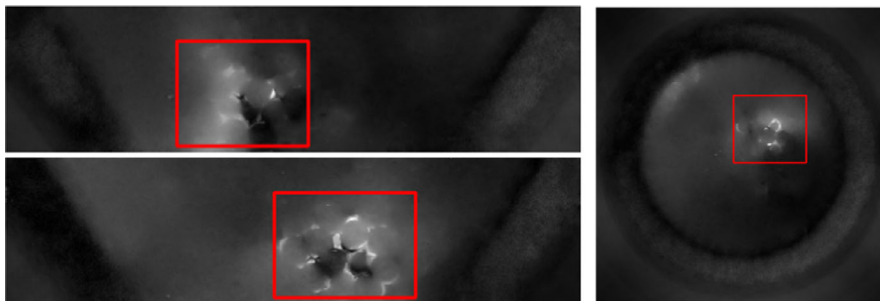


FIG. 6. A visualization of the norm of the optical flow between frame 95 and frame 98. Three orthogonal slices are shown, one for each dimension. The red rectangle marks a region of interest where the flow is large and complex, which complicates the inversion of the deformation vector field.

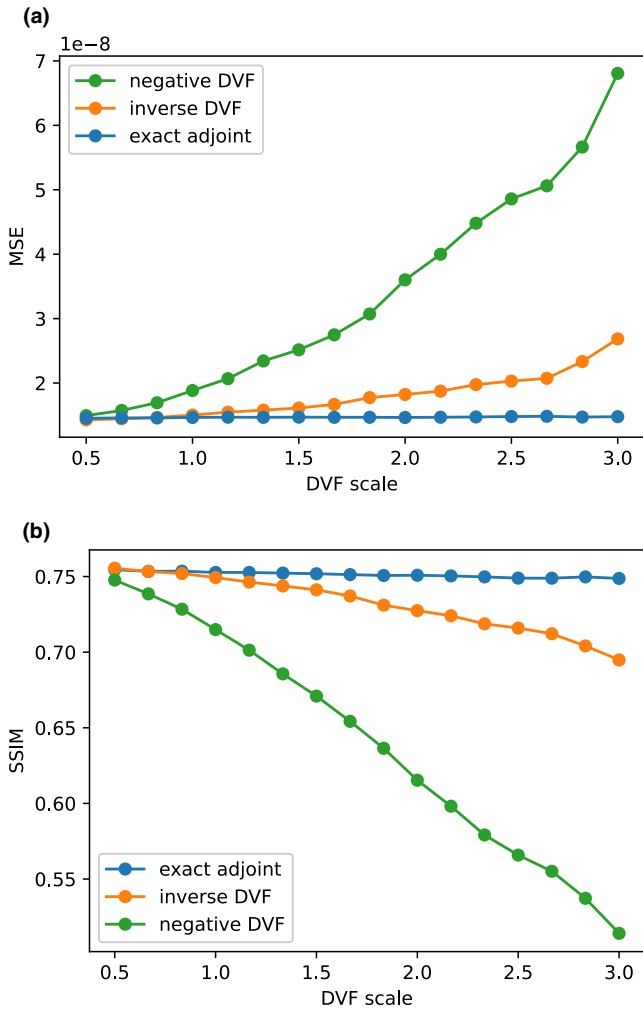


FIG. 7. Mean squared error (7a) and SSIM (7b) of the reconstructions of the lung phantom with differently scaled deformation vector fields (see Section 2.F.1)

3. RESULTS

3.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 Figs. 7(a) and 7(b), 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.

3.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 Figs. 10(a) and 10(b), respectively. The reconstructions at noise level 10^5 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 10^5 .

3.C. Liquid foam dataset

The MSE and SSIM, with respect to the reference reconstruction, of all reconstructions of subscan 98 are 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.

Figure 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.

4. DISCUSSION

When comparing the three methods, it is important to keep in mind that using the exact adjoint and using the negative DVF have almost equal computational demands, while using the inverse DVF requires an iterative inversion of the DVFs, and the storage of these inverted DVFs. For instance,

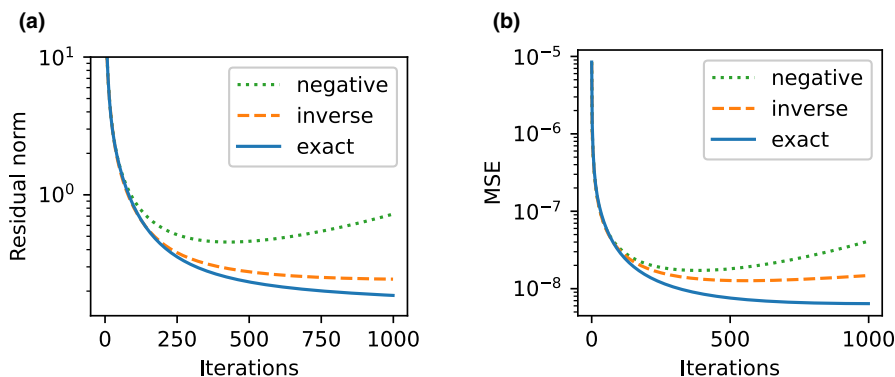


FIG. 8. Convergence plots comparing the different methods at scale 3.

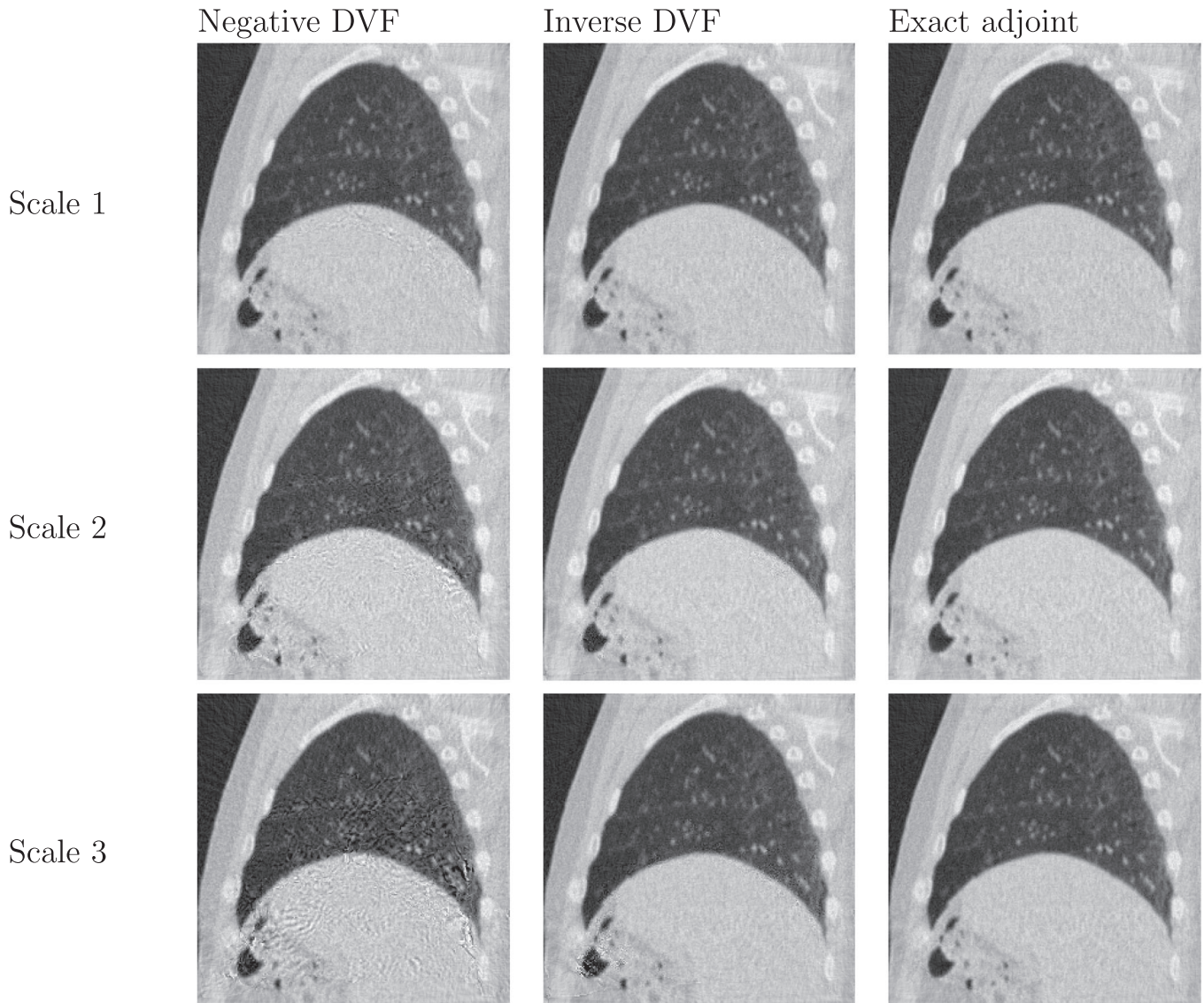


FIG. 9. Reconstructions of the lung phantom with different methods and differently scaled deformation vector fields.

if a $500 \times 500 \times 500$ volume is used, then a DVF stored in single precision floating point numbers requires 1.5 GB of memory. Having to store twice the amount of DVFs can easily lead to a lack of RAM or GPU memory. The inversion algorithm also requires the 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.

4.A. Simulation experiment with known DVF

The measurements in Fig. 7 show that using the exact adjoint is robust to the scale of the DVF, while the

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

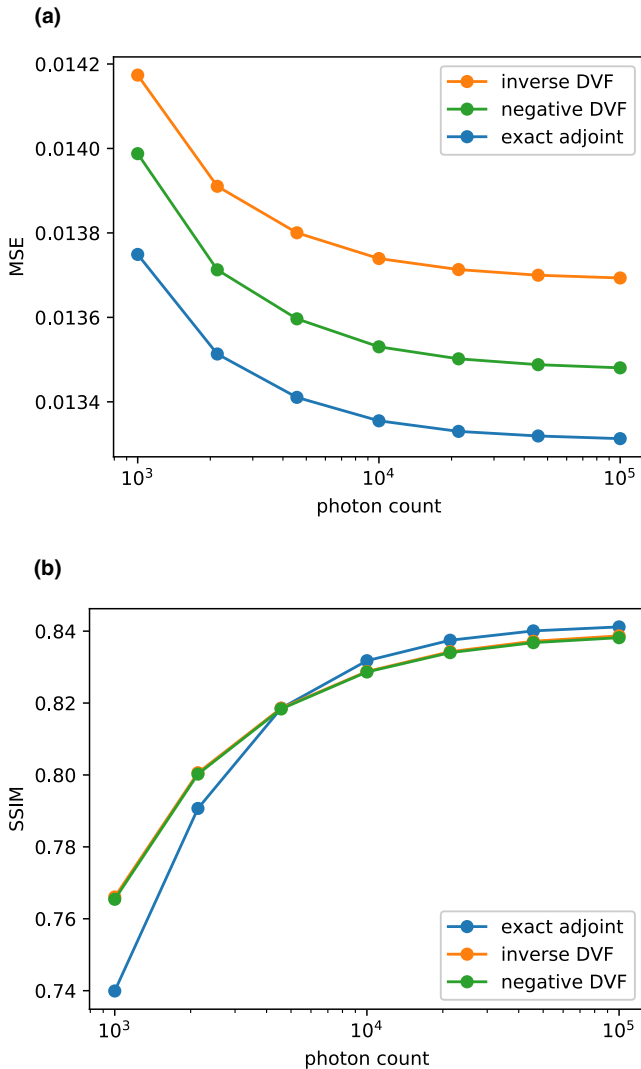


FIG. 10. Mean squared error (10a) and SSIM (10b) of the reconstructions of subscan 2 of the simulated scan, using the data of subscan 1 and 3, at different noise levels (see Section 2.F.2).

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.

4.B. Simulation experiment with unknown DVF

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

In Fig. 12, the effects of the different methods on the reconstruction can be seen visually. The reconstruction with exact adjoints is more sharp and granular, while the two approximating methods are smooth. The approximations do cause an artifact around a fast growing bubble. This is due to the fact that the DVF is not invertible in this area, so methods relying on an approximate inversion of the DVF

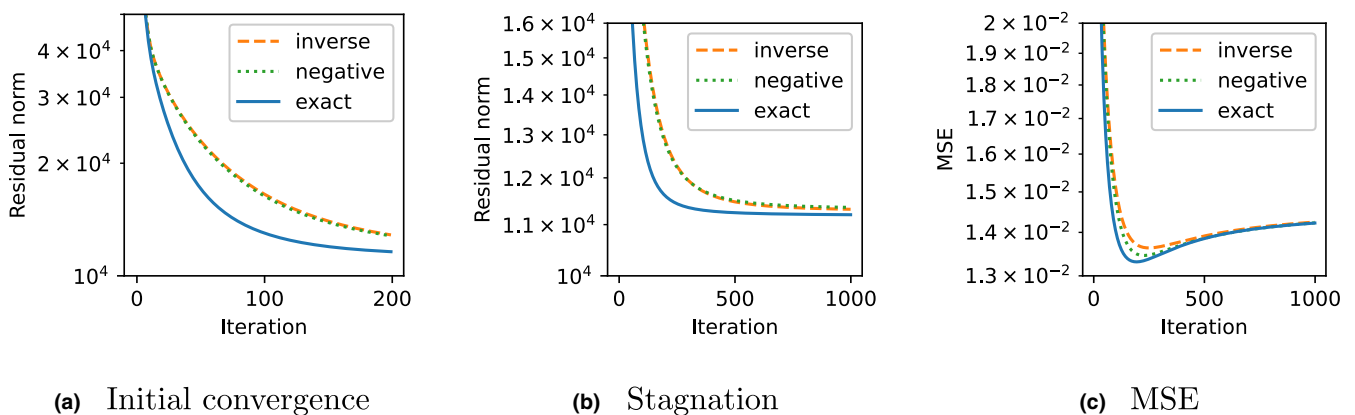


FIG. 11. Convergence plots comparing the different methods at photon count 10^5 .

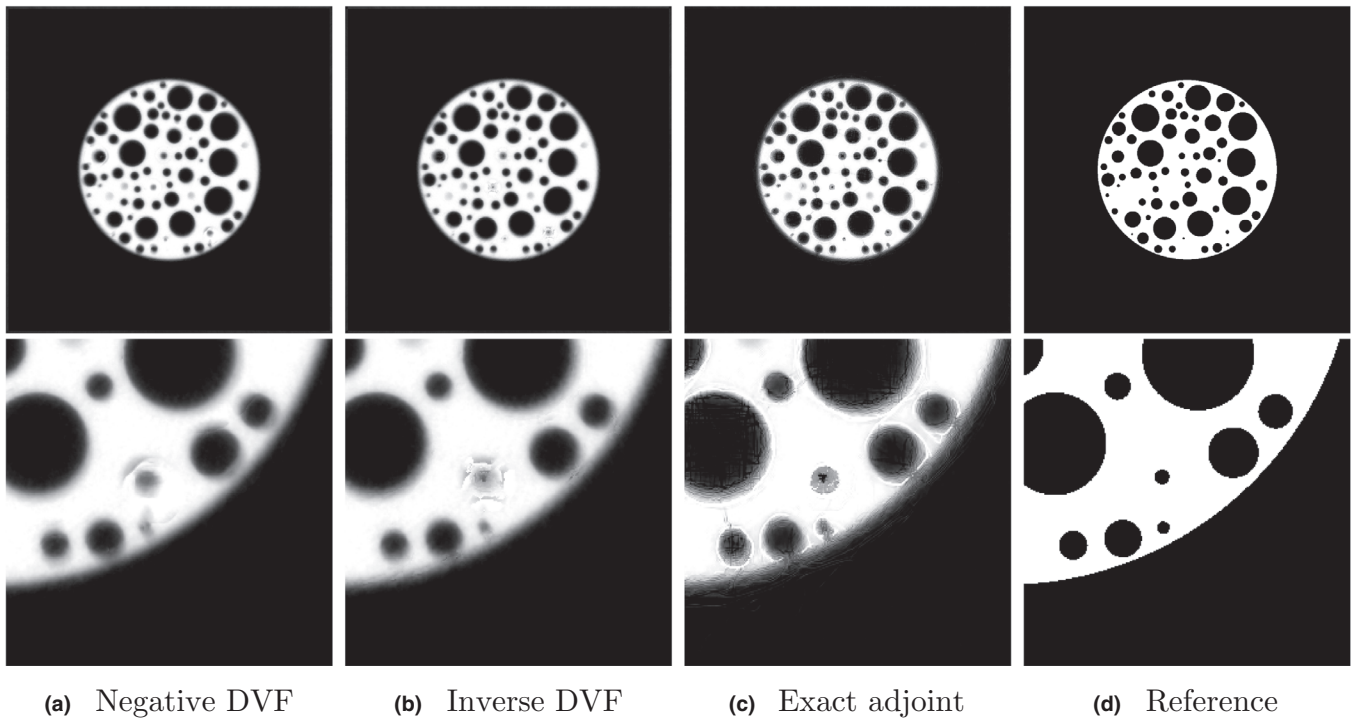


FIG. 12. Reconstructions of subscan 2 of the simulated scan, using the data of subscan 1 and 3, with a photon count of 10^5 , and different methods for the adjoint warps. The second row shows a zoomed in image of a bubble with artifacts.

fail in such locations. Our proposed algorithm for the exact adjoint does not require an approximate inverse of the DVF. Still, the reconstruction with the exact adjoints shows some artifacts which can not be seen in the reconstructions with the other methods. This might be caused by the inaccuracies present in the estimated DVF. Using the exact adjoint likely results in a reconstruction which is more faithful to the inaccurate DVF. Therefore, the decision of which method to use in practice must be made with the accuracy of the estimated DVF in mind. Possible improved methods could be made by combining the different methods based on the DVF accuracy.

4.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 small. Using the negative DVF yields the lowest reconstruction quality overall. In terms of SSIM, the method using the inverse DVF is almost indistinguishable from the method using the exact adjoints. For

select subscans (99, 105, 106, 107) there is a very small advantage to the exact adjoints. In terms of MSE, using the inverse DVF often gives a slightly higher quality than using the exact adjoint. It should be noted that these errors were measured over a large volume of which only a small part shows complex motion, as shown in Fig. 6. The lower MSE for the approximation using the inverse DVF might be explained as follows: in most of the volume, the DVF is very small, so an accurate inversion can be found. This explains why there is no substantial loss of quality using this method. On top of that, the approximation introduces a smoothing effect, which can remove detail, but it also removes noise and reduces some artifacts. In this experiment, no notable difference in convergence speed was observed between the different methods.

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 larger time steps.

Figure 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

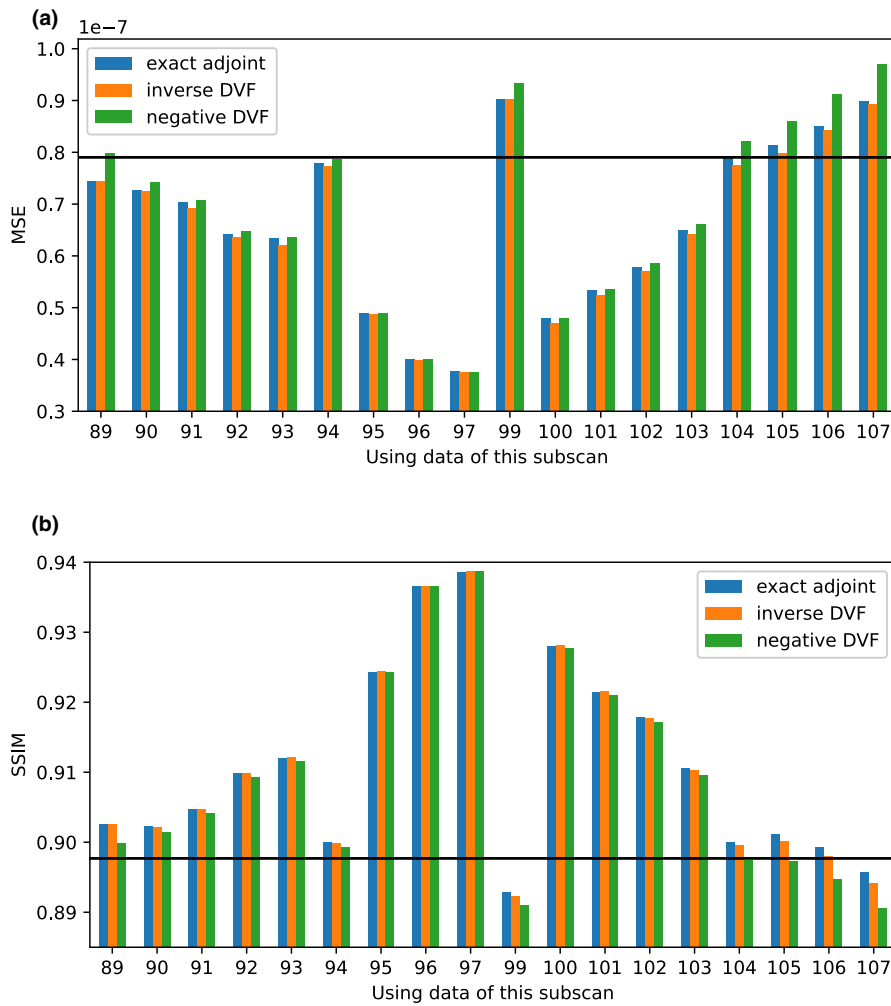


FIG. 13. Mean squared error (13a) and SSIM (13b) of the reconstruction of subscan 98, using 150 projections + 150 projections of one extra subscan. The black line shows the MSE and SSIM of a reconstruction with 150 projections, without extra projections from an other subscan. It can be used as a reference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

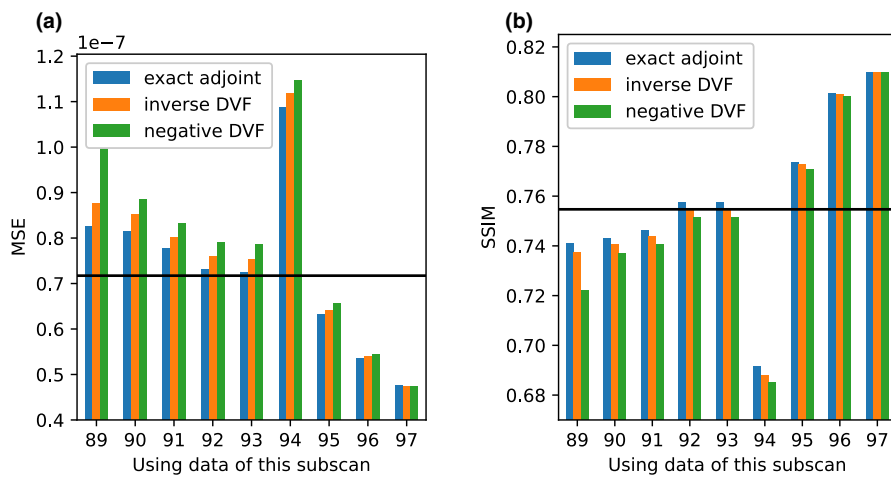


FIG. 14. Mean squared error (MSE) (14a) and Structural similarity index measure (SSIM) (14b) of the region of interest (Fig. 6) of subscan 98, using 150 projections + 150 projections of one extra subscan. The black line shows the MSE and SSIM of a reconstruction with 150 projections, without extra projections 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.

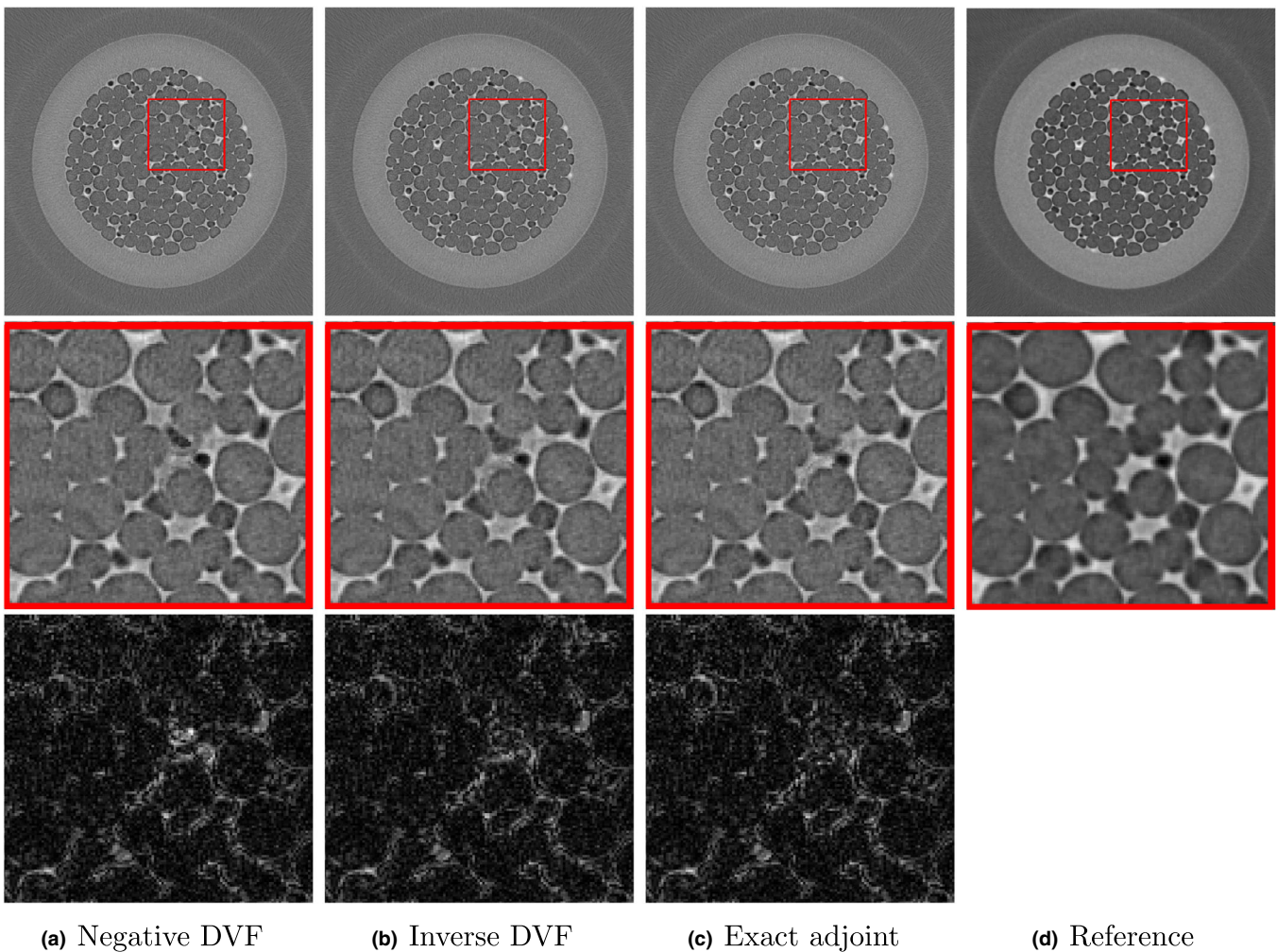


FIG. 15. A horizontal slice of the reconstruction of subscan 98 using data of subscan 95 with 3 different methods, with the region of interest (Fig. 6) magnified. The bottom row shows absolute difference images of the magnified region, with respect to the reference.

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.

5. CONCLUSIONS

In this work, a GPU accelerated algorithm that computes the exact adjoint action of multivariate spline based image warping was introduced. It was shown how this adjoint operator can be applied to reconstruction problems in 4D-CT that rely on optical flow techniques. Our method was experimentally compared against two alternative methods that approximate the adjoint image warp by a regular warp along an approximate inverse of the optical flow. The experiments showed that our method can improve reconstruction quality when the flow is difficult to invert. Moreover, an accurate inversion of a flow field requires an iterative algorithm with a stopping criterion, as well as the storage of the resulting inverse flow. Because our method does not require the inversion of the

flow, we avoid this tunable parameter and the memory costs of storing the inverted flow.

ACKNOWLEDGMENTS

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

CONFLICT OF INTEREST

The authors have no relevant conflicts of interest to disclose.

^{a)} Author to whom correspondence should be addressed. Electronic mail: jens.renders@uantwerpen.be

REFERENCES

1. Horn BKP, Schunck BG. Determining optical flow. *Artif Intell.* 1981;17:185–203.

2. Lucas BD, Kanade T. An Iterative Image Registration Technique with an Application to Stereo Vision, in Proceedings of the 7th International Joint Conference on Artificial Intelligence -Volume 2, IJCAI'1, San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.; 1981:674–679.
3. Zach C, Pock T, Bischof H. A duality based approach for realtime TV-L1 optical flow, In Joint Pattern Recognition Symposium. Springer; 2007:214–223.
4. Rueckert D, Sonoda LI, Hayes C, Hill DLG, Leach MO, Hawkes DJ. Nonrigid registration using free-form deformations: application to breast MR images. *IEEE Trans Med Imaging*. 1999;18:712–721.
5. Wolberg G. *Digital Image Warping*. CA: IEEE Computer Society Press Los Alamitos; 1990:10662.
6. Brehm M, Paysan P, Oelhafen M, M. Kachelriess, . Artifact-resistant motion estimation with a patient-specific artifact model for motion-compensated cone-beam CT. *Med Phys*. 2013;40:101913.
7. Mory C, Janssens G, Rit S. Motion-aware temporal regularization for improved 4D cone-beam computed tomography. *Phys Med Biol*. 2016;61:6856.
8. Wang J, Gu X. Simultaneous motion estimation and image reconstruction (SMEIR) for 4D cone-beam CT. *Med Phys*. 2013;40:101912.
9. Tang Q, Cammin J, Srivastava S, Taguchi K. A fully four-dimensional, iterative motion estimation and compensation method for cardiac CT. *Med Phys*. 2012;39:4291–4305.
10. Van Nieuwenhove V, De Beenhouwer J, Vlassenbroeck J, Brennan M, Sijbers J. MoVIT: a tomographic reconstruction framework for 4D-CT. *Opt Express*. 2017;25:19236–19250.
11. Zang G, Idoughi R, Tao R, Lubineau G, Wonka P, Heidrich W. Warp-and-project tomography for rapidly deforming objects. *ACM Trans Graphics (TOG)*. 2019;38:86.
12. Burger M, Dirks H, Frerking L, Hauptmann A, Helin T, Siltanen S. A variational reconstruction method for undersampled dynamic X-ray tomography based on physical motion models. *Inverse Prob*. 2017;33:124008.
13. Zang G, Idouchi R, Tao R, Lubineau G, Wonka P, Heidrich W. Space-time tomography for continuously deforming objects. *ACM Trans Graphics (TOG)*. 2018;37:100.
14. Odstrcil M, Holler M, Raabe J, et al. Ab initio nonrigid x-ray nanotomography. *Nat Commun*. 2019;10:2600.
15. Ruymbeek K, Vanroose W. Algorithm for the reconstruction of dynamic objects in CT-scanning using optical flow. *J Comput Appl Math*. 2020;367:112459.
16. Mory C, Rit S. Improving iterative 4D CBCT through the use of motion information, Proc. of Fully 3D 2015 (Newport, Rhode Island, USA.); 2015.
17. Feng T, Wang J, Fung G, Tsui B. Non-rigid dual respiratory and cardiac motion correction methods after, during, and before image reconstruction for 4D cardiac PET. *Phys Med Biol*. 2015;61:151.
18. Lekien F, Marsden J. Tricubic interpolation in three dimensions. *Int J Numer Meth Eng*. 2005;63:455–471.
19. Paeth AW. *Graphics Gems V*. USA: Academic Press Inc; 1995.
20. Van der Walt, S, Schönberger JL, Nunez-Iglesias J, et al. Scikit-image: image processing in Python. *PeerJ*. 2014;2:e453.
21. The Sage Developers, SageMath, the Sage Mathematics Software System (Version 9.1), 2020, <https://www.sagemath.org>.
22. Renders J. Adjoint Image Warping. 2020; <https://doi.org/10.5281/zenodo.4288715>.
23. Sánchez PJ, Meinhardt-Llopis E, Facciolo G. TV-L1 optical flow estimation. *Image Process Line*. 2013;3:137–150.
24. Chambolle A, Pock T. A first-order primal-dual algorithm for convex problems with applications to imaging. *J Math Imaging Vision*. 2011;40:120–145.
25. Hansen PC. Discrete inverse problems: insight and algorithms, volume 7, chapter 6, Siam, 2010:111–112.
26. Chen M, Lu W, Chen Q, Ruchala KJ, Olivera GH. A simple fixed-point approach to invert a deformation field. *Med Phys*. 2008;35:81–88.
27. Wang Z, Bovik AC, Sheikh HR, Simoncelli EP. Image quality assessment: from error visibility to structural similarity. *IEEE Trans Image Process*. 2004;13:600–612.
28. van Aarle W, Palenstijn WJ, Cant J, et al. Fast and flexible x-ray tomography using the ASTRA toolbox. *Opt Express*. 2016;24:25129–25147.
29. van Aarle W, Palenstijn WJ, De Beenhouwer J, et al. The ASTRA toolbox: a platform for advanced algorithm development in electron tomography. *Ultramicroscopy*. 2015;157:35–47.
30. Palenstijn WJ, Batenburg KJ, Sijbers J. Performance improvements for iterative electron tomography reconstruction using graphics processing units (GPUs). *J Struct Biol*. 2011;176:250–253.
31. Werner R, Hofmann C, Mücke E, Gauer T. Reduction of breathing irregularity-related motion artifacts in low-pitch spiral 4D CT by optimized projection binning. *Radiat Oncol*. 2017;12:100.
32. Pelt D, Batenburg KJ, Sethian J. Improving tomographic reconstruction from limited data using mixed-scale dense convolutional neural networks. *J Imaging*. 2018;4:128.
33. De Carlo F, Gürsoy D, Ching DJ, et al. TomoBank: a tomographic data repository for computational x-ray science. *Measur Sci Technol*. 2018;29:034004.
34. Haase V, Stierstorfer K, Hahn K, Schöndube H, Maier A, Noo F. Exploring the space between smoothed and non-smooth total variation for 3D iterative CT reconstruction. In: Schmidt TG, Chen G-H, Bosmans H, eds. *Medical Imaging 2019: Physics of Medical Imaging*. Vol. 10948, International Society for Optics and Photonics, SPIE, 2019:511–517.