

Adjoint pairs of image warping operators for motion modeling in 4D-CT

Jens Renders, Jan Sijbers and Jan De Beenhouwer

Abstract—In 4D-CT, a 4D image (3D+time) needs to be reconstructed from projection data of a moving object. A simple approach is to acquire a number of consecutive 3D-CT scans (called subscans) and to then reconstruct them separately. When the object is moving rapidly, or a high time resolution is desired, a short acquisition time is needed for the subscans, leading to a low number of projections and a strongly underdetermined system to solve. Recently, a number of methods have been proposed to alleviate this issue using image warping operators. By warping an image from a certain time frame to a neighboring time frame, the image can be compared to the projection data of that neighboring time frame, reducing underdetermination. The motion that is used for the warp can be either estimated in advance, or it can be reconstructed jointly with the images. To invert a model that incorporates warping operators, the adjoints of these warping operators are needed. The current methods approximate these adjoint operators by warping operators for an approximate inverse flow. We propose to use the exact adjoints instead, which leads to faster convergence, less computation and less memory requirements.

Index Terms—4D-CT, warping, motion.

I. INTRODUCTION

A 4D-CT scan is a series of n consecutive, regular CT scans called subscans. It is generally assumed that the scanned object is motionless during each subscan, such that the scan can be modeled by the equation $\mathbf{W}_i \mathbf{x}_i = \mathbf{p}_i$, where \mathbf{W}_i is the projection matrix, \mathbf{x}_i is the scanned object and \mathbf{p}_i is the projection data of the i -th subscan. The extent to which this assumption is true depends on the time that passed during the subscan, and the speed of the motion of the object. 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

$$\mathbf{W}_i \mathbf{x}_i = \mathbf{p}_i, \quad i = 1, \dots, n, \quad (1)$$

which, in this case, are highly underdetermined. Equivalently, the problem can be represented as one big underdetermined system of the form

$$\begin{bmatrix} \mathbf{W}_1 & 0 & 0 & 0 \\ 0 & \mathbf{W}_2 & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & \mathbf{W}_n \end{bmatrix} \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \vdots \\ \mathbf{x}_n \end{bmatrix} = \begin{bmatrix} \mathbf{p}_1 \\ \mathbf{p}_2 \\ \vdots \\ \mathbf{p}_n \end{bmatrix}. \quad (2)$$

One way to alleviate this underdetermination is to link the time frames together using image registration or optical flow

J. Renders, J. Sijbers and J. De Beenhouwer are with imec-Vision Lab, a research group at the University of Antwerp, Belgium (email: jens.renders@uantwerpen.be)

techniques [1]–[4] and image warping [5]. In [6], [7] problem (2) is regularized with terms that constrain the change between frames, i.e. with constraints on the optical flow. In [7], warping operators are involved in these regularization terms. In the MoVIT algorithm [8] and in [9], the images are warped along the flow between frames before they are projected using a standard projection geometry. In [10], [11] 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.

For methods that involve image warping operators in the forward model, or in the regularization terms, the adjoints of these warping operators are needed for reconstruction with gradient based methods. The current methods implement the adjoint image warping operators by warping along an approximated inverse of the flow. 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.

In [12] a pair of adjoint warping operators with a custom 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. However, these adjoint warp operators can not be directly substituted for the approximated adjoint warping operators in the previously mentioned methods, because the custom interpolation method of [12] requires more information than general optical flow methods provide. In this paper, we investigate the implementation and the effect of adjoint warping operators using the more generally applicable and commonly used multivariate spline interpolation.

II. METHODS

Although, our proposed method to calculate the adjoint warping operators is applicable to any method that uses image warping with multivariate spline interpolation, including [7]–[9], we will demonstrate their use on a basic method similar to [8]. In a first step, each time frame is reconstructed separately by solving (2). On these initial reconstructions, we use an optical flow algorithm to estimate the flow between the time frames. Finally, we use a least squares solver to solve the same dynamic tomographic model that MoVIT solves.

A. Warping operators

While the scanned object can change during a 4D-CT scan, we assume that the materials that make up the object do not

change, but only get repositioned¹. Under this assumption, we can deform an image $\mathbf{x}_i \in \mathbb{R}^N$ into the image $\mathbf{x}_j \in \mathbb{R}^N$ 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.

Moving 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 moving the voxels and resampling. A standard choice of resampling method is multivariate spline interpolation (usually linear or cubic splines), used in for example [5], [8], [9]. 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 a warping operator that transforms \mathbf{x}_i into \mathbf{x}_j .

B. Dynamic tomographic model

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 M_{ij} , such that $\mathbf{x}_j = M_{ij}\mathbf{x}_i$ holds for all j . This can be written as:

$$\begin{bmatrix} M_{i1} \\ M_{i2} \\ \vdots \\ M_{in} \end{bmatrix} \mathbf{x}_i = \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \vdots \\ \mathbf{x}_n \end{bmatrix} . \quad (3)$$

Substituting (3) into (2) yields the dynamic tomographic model:

$$\begin{bmatrix} \mathbf{W}_1 & 0 & 0 & 0 \\ 0 & \mathbf{W}_2 & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & \mathbf{W}_n \end{bmatrix} \begin{bmatrix} M_{i1} \\ M_{i2} \\ \vdots \\ M_{in} \end{bmatrix} \mathbf{x}_i = \begin{bmatrix} \mathbf{p}_1 \\ \mathbf{p}_2 \\ \vdots \\ \mathbf{p}_n \end{bmatrix} . \quad (4)$$

or more concisely:

$$\mathbf{W}\mathbf{M}\mathbf{x} = \mathbf{p} . \quad (5)$$

In practice, (3) can not be achieved exactly, but by estimating the flow between initial reconstructions, a good approximation can be obtained. System (5) is the one that is solved by MoVIT, and it can be interpreted as a factorization of the model presented in [10] and [11], where the DVFs are used to directly modify the projection matrix \mathbf{W} instead of adding the extra factor \mathbf{M} . The new system has the same number of equations as (2), but the unknowns are reduced to only the voxels of the chosen time frame.

C. Estimating the DVFs

There are many image registration and optical flow algorithms available to estimate the DVFs for the warping operators in (4) such as [1]–[4]. We use the TV-L1 optical flow algorithm provided by scikit-image, which is detailed in [13].

¹This assumption is only approximately valid, which puts a limit on the number of time frames we will be able to combine.

D. Least squares solvers

There are many choices of least squares solvers for the system (4). These different choices can enforce different types of constraints on the solution. In [7], [9] the Primal Dual Hybrid Gradient method of [14] is used to enforce sparsity of the gradient of the solution. MoVIT uses a modified version of SIRT with an intuitive interpretation. We choose to use a basic (projected) gradient descent to demonstrate the effect of using the exact adjoints. This is very close to what was done in MoVIT as SIRT is a slightly altered version of gradient descent [15]. We use the Barzilai–Borwein step size rule [16] and we enforce minimum and maximum constraints on the voxel values.

E. Adjoint pairs of warping operators using multivariate spline interpolation

There are two different approaches to implement image warping operators, referred to as forward and backward warping [5]. Assume we have two images, \mathbf{a} and \mathbf{b} and a DVF describing the flow from \mathbf{a} to \mathbf{b} . With forward image warping, the voxels of \mathbf{a} are first moved along the DVF to obtain non-grid data representing the warped version of \mathbf{a} , and this non-grid data is then resampled at grid points to get an image similar to \mathbf{b} . Backward image warping is an other approach, where 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 \mathbf{b} at that point. The result is an image similar to \mathbf{a} .

Both forward and backward warping are viable methods to implement the warping operators of (4). If we want to be able to transform \mathbf{x}_i into \mathbf{x}_j , we can either get a DVF from \mathbf{x}_i to \mathbf{x}_j and use forward warping, or we can get a DVF from \mathbf{x}_j to \mathbf{x}_i and use backward warping. We implement our warping operators using backward warping with rectangular multivariate spline interpolation. For convenience, we will work with 2D images and bilinear splines, but the methods are easily extendable to 3D images and higher order splines. In [8] and [9], cubic splines are used.

Our image \mathbf{x} can be thought of as a function

$$\mathbf{x} : [n] \times [m] \rightarrow \mathbb{R} , \quad (6)$$

where $n, m \in \mathbb{N}$ and $\forall k \in \mathbb{N} : [k] = \{1, \dots, k\}$. For each pair of integer coordinates, it yields a voxel value. Its bilinear interpolation at a non-integer position (a, b) is given by

$$x'(a, b) = c_1\mathbf{x}(\mathbf{q}_1) + c_2\mathbf{x}(\mathbf{q}_2) + c_3\mathbf{x}(\mathbf{q}_3) + c_4\mathbf{x}(\mathbf{q}_4) , \quad (7)$$

where $\mathbf{q}_1, \dots, \mathbf{q}_4$ are the integer valued points surrounding (a, b) , and c_1, \dots, c_4 are the bilinear spline coefficients.

Our DVF \mathbf{v} can be thought of as a function

$$\mathbf{v} : [n] \times [m] \rightarrow \mathbb{R}^2 \quad (8)$$

which yields a 2D displacement vector for each voxel. Our backward warping operator \mathbf{M} transforms \mathbf{x} into a new image $\mathbf{y} = \mathbf{M}\mathbf{x}$ of which the (i, j) -th voxel value is obtained by sampling \mathbf{x} at position $(i, j) + \mathbf{v}(i, j)$. That is:

$$(\mathbf{M}\mathbf{x})(i, j) = x'((i, j) + \mathbf{v}(i, j)) . \quad (9)$$

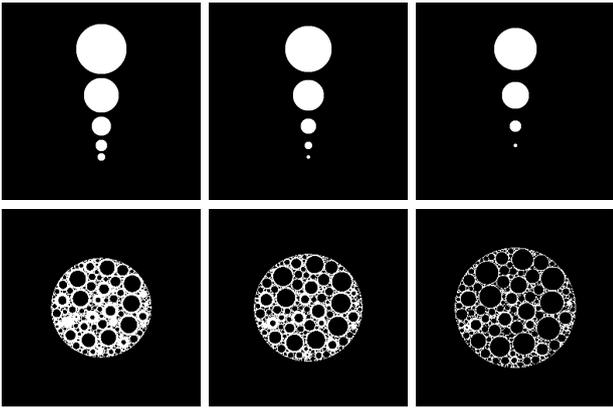


Fig. 1: Ground truths for our experiments. The top row is a 2D+time phantom consisting of 3 time frames with shrinking circles. The bottom row shows 3 time frames of a slice of a 4D simulated forming foam. The actual foam phantom has one time frame per projection: 128×3 .

Since, by (7), (9) is simply a linear combination of voxel values of \mathbf{x} , we can interpret the action of \mathbf{M} as a matrix vector product. To do this, we need to represent the image as a vector again: $\mathbf{x} \in \mathbb{R}^N$, where each voxel gets a single integer index. \mathbf{M} can then be represented by a matrix, with 4 non-zero coefficients on each row, namely the bilinear spline coefficients of (7) at the corresponding voxel indices. The adjoint operator \mathbf{M}^T is then simply given by the matrix with the rows of \mathbf{M} as its columns. The rows of \mathbf{M} or equivalently, the columns of \mathbf{M}^T can be computed on the fly, so there is no need to store these matrices. If we denote the i -th row of \mathbf{M} , i.e. the i -th column of \mathbf{M}^T by \mathbf{r}_i , then the action of \mathbf{M}^T on a vector $\mathbf{y} \in \mathbb{R}^N$ can be implemented as follows:

$$\mathbf{M}^T \mathbf{y} = \sum_{i=1}^N y_i \mathbf{r}_i \quad . \quad (10)$$

For comparison, we also implemented the approximation of the adjoint by using the inverse DVF. Like [8] and [10], we calculate the inverse DVF with the iterative method presented in [17] with 15 iterations. The action of \mathbf{M}^T is then approximated by using the same warping method as \mathbf{M} , but with this inverse DVF.

III. EXPERIMENTS

To compare the effects of the exact adjoints to the approximated adjoints, we performed two simulation experiments.

The first experiment used a simple 512×512 2D+time phantom of 5 circles of different sizes which shrink in 3 time frames. We performed 3 subscans, one for each time frame, during which the phantom is static. Each subscan consisted of 128 parallel beam projections over a range of π rad. The DVFs were calculated on the ground truth time frames.

The second experiment used a 256×256 slice of a 4D (3D+time) phantom of growing spheres in a cylinder, mimicking the formation of a foam [18]. This phantom was generated at as many time frames as projections, such that each projection sees a slightly different object. We performed

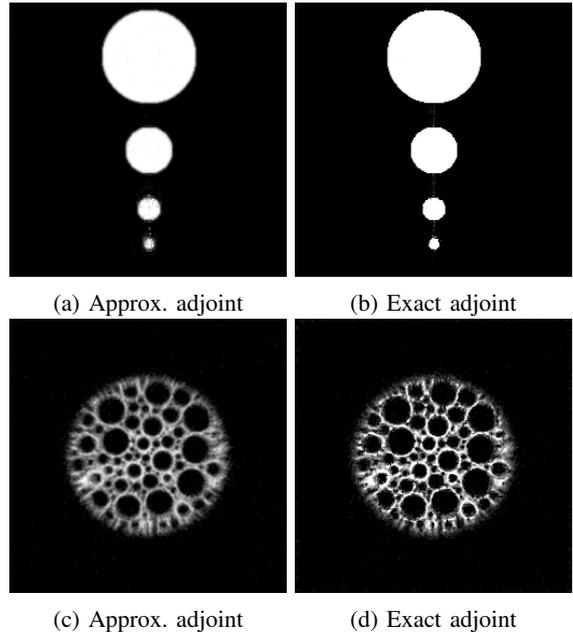


Fig. 2: The resulting reconstructions after 30 iterations. Figures (a) and (b) have been zoomed in on the smaller circles to highlight the details.

a simulated scan with 3 subscans. Each subscan consisted of 128 parallel beam projections over a range of π rad. In this experiment, we first made initial reconstructions by solving (2). We then calculated the DVFs on these initial reconstructions.

In both experiments, we reconstructed the middle time frame using both of its neighbors by solving (4) with 3 time frames. The algorithm was ran for 30 iterations. Both phantoms are $\{0, 1\}$ -valued, so we enforced a minimum constraint of 0 and a maximum constraint of 1 on the reconstruction.

All projection operators, both for the simulations and for the reconstruction, were implemented using the ASTRA toolbox [19]–[21]

IV. RESULTS AND DISCUSSION

The implementation using the approximate adjoints required the storage of two DVFs per neighboring time frame, and one image. A DVF consists of one displacement vector per voxel, so it takes as much storage as two images for a 2D problem, or three images for a 3D problem. In our experiments, a total storage of 9 images was required. The implementation using the exact adjoint required only the storage of one DVF per neighboring time frame, cutting the memory requirements almost in half, to a total of 5 images.

The effect of the choice of adjoints can be visually observed in Fig. 2. After 30 iterations, the reconstruction with the exact adjoints is visibly sharper than the reconstruction with the approximate adjoints. Some bubbles which can barely be discerned in Fig. 2(c) are clearly visible in Fig. 2(d), while Fig. 2(d), used the same amount of iterations and less computation time. In Fig. 2(b) the sharper edges are clearly visible.

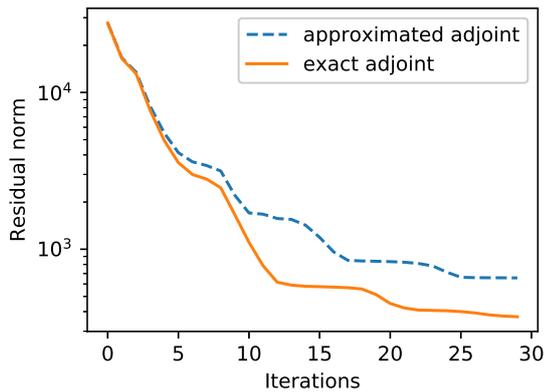


Fig. 3: The difference in convergence when using the exact adjoints versus the approximated adjoints on the circle phantom

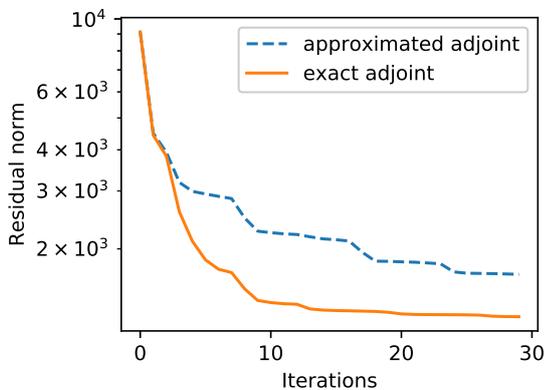


Fig. 4: The difference in convergence when using the exact adjoints versus the approximated adjoints on the foam phantom

The faster convergence can also be verified in the residual plots in Fig. 3 and Fig. 4. For each iteration, we plotted the 2-norm of the residual, i.e. $\|Ax_i - p\|_2$, where $A = WM$. We see that the method with the exact adjoints gets the residual down much earlier, and keeps it lower at all iterations.

V. CONCLUSION

Although the adjoint of an image warping operator can be approximated by the same kind of warping operator with an approximate inverse of its DVF, we have shown that it is beneficial to use the exact adjoint instead. Not only does it provide a faster rate of convergence, it also avoids several problems that the approximate adjoints impose. Calculating an inverse of a DVF is computationally intensive and takes as much space as three images. When scaling to large 3D problems, this becomes an issue. The inverting of a DVF also requires the choice of an extra parameter, namely the number of iterations for the inversion algorithm. These problems are not present with the proposed approach of computing the adjoints.

ACKNOWLEDGMENT

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

REFERENCES

- [1] B. K. P. Horn and B. G. Schunck, "Determining optical flow," *Artificial Intelligence*, vol. 17, no. 1-3, pp. 185–203, 1981.
- [2] 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 Artificial Intelligence - Volume 2*, ser. IJCAI'81. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1981, p. 674–679.
- [3] C. Zach, T. Pock, and H. Bischof, "A duality based approach for realtime TV-L1 optical flow," in *Joint Pattern Recognition Symposium*. Springer, 2007, pp. 214–223.
- [4] D. Rueckert, L. I. Sonoda, C. Hayes, D. L. G. Hill, M. O. Leach, and D. J. Hawkes, "Nonrigid registration using free-form deformations: application to breast MR images," *IEEE Transactions on Medical Imaging*, vol. 18, no. 8, pp. 712–721, 1999.
- [5] G. Wolberg, *Digital Image Warping*. IEEE Computer Society Press Los Alamitos, CA, 1990, vol. 10662.
- [6] M. Burger, H. Dirks, L. Frerking, A. Hauptmann, T. Helin, and S. Siltanen, "A variational reconstruction method for undersampled dynamic X-ray tomography based on physical motion models," *Inverse Problems*, vol. 33, no. 12, p. 124008, 2017.
- [7] 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)*, vol. 37, no. 4, p. 100, 2018.
- [8] V. Van Nieuwenhove, J. De Beenhouwer, J. Vlassenbroeck, M. Brennan, and J. Sijbers, "MoVIT: a tomographic reconstruction framework for 4D-CT," *Optics Express*, vol. 25, no. 16, pp. 19 236–19 250, 2017.
- [9] G. Zang, R. Idoughi, R. Tao, G. Lubineau, P. Wonka, and W. Heidrich, "Warp-and-project tomography for rapidly deforming objects," *ACM Transactions on Graphics (TOG)*, vol. 38, no. 4, p. 86, 2019.
- [10] 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*, vol. 10, no. 1, p. 2600, 2019.
- [11] K. Ruymbeek and W. Vanroose, "Algorithm for the reconstruction of dynamic objects in CT-scanning using optical flow," *Journal of Computational and Applied Mathematics*, vol. 367, p. 112459, 2020.
- [12] T. Feng, J. Wang, G. Fung, and B. Tsui, "Non-rigid dual respiratory and cardiac motion correction methods after, during, and before image reconstruction for 4d cardiac pet," *Physics in Medicine & Biology*, vol. 61, no. 1, p. 151, 2015.
- [13] J. S. Pérez, E. Meinhardt-Llopis, and G. Facciolo, "TV-L1 optical flow estimation," *Image Processing On Line*, vol. 2013, pp. 137–150, 2013.
- [14] A. Chambolle and T. Pock, "A first-order primal-dual algorithm for convex problems with applications to imaging," *Journal of Mathematical Imaging and Vision*, vol. 40, no. 1, pp. 120–145, 2011.
- [15] P. C. Hansen, *Discrete inverse problems: insight and algorithms*. Siam, 2010, vol. 7, ch. 6, pp. 111–112.
- [16] J. Barzilai and J. M. Borwein, "Two-point step size gradient methods," *IMA Journal of Numerical Analysis*, vol. 8, no. 1, pp. 141–148, 1988.
- [17] M. Chen, W. Lu, Q. Chen, K. J. Ruchala, and G. H. Olivera, "A simple fixed-point approach to invert a deformation field," *Medical Physics*, vol. 35, no. 1, pp. 81–88, 2008.
- [18] D. Pelt, K. J. Batenburg, and J. Sethian, "Improving tomographic reconstruction from limited data using mixed-scale dense convolutional neural networks," *Journal of Imaging*, vol. 4, no. 11, p. 128, 2018.
- [19] W. van Aarle, W. J. Palenstijn, J. Cant, E. Janssens, F. Bleichrodt, A. Dabrovolski, J. De Beenhouwer, K. J. Batenburg, and J. Sijbers, "Fast and flexible X-ray tomography using the ASTRA toolbox," *Optics Express*, vol. 24, no. 22, pp. 25 129–25 147, 2016.
- [20] W. van Aarle, W. J. Palenstijn, J. De Beenhouwer, T. Altantzis, S. Bals, K. J. Batenburg, and J. Sijbers, "The ASTRA toolbox: A platform for advanced algorithm development in electron tomography," *Ultra-microscopy*, vol. 157, pp. 35–47, 2015.
- [21] 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*, vol. 176, no. 2, pp. 250–253, 2011.