

3D Reconstruction of Stents and Guidewires in an Anthropomorphic Phantom From Three X-Ray Projections

Tim Vöth^{a,b}, Thomas König^b, Elias Eulig^a, Michael Knaup^a, Veit Wiesmann^b, Klaus Hörndler^b, and Marc Kachelrieß^a

^aGerman Cancer Research Center (DKFZ), Im Neuenheimer Feld 280, 69120 Heidelberg, Germany

^bZiehm Imaging GmbH, Lina-Ammon-Straße 10, 90471 Nürnberg, Germany

ABSTRACT

Today, the subcutaneous, minimally invasive procedures performed in interventional radiology are usually guided by 2D X-ray fluoroscopy. In 2D X-ray fluoroscopy a series of 2D X-ray images is displayed. For challenging procedures however, 3D X-ray fluoroscopy would be advantageous. In 3D X-ray fluoroscopy, a series of 3D images, which is reconstructed from a series of 2D X-ray images, is displayed. Because the number of images used for guiding an intervention is very high, little dose can be spent per 3D reconstruction of a 3D fluoroscopy. To save dose and to minimize motion artifacts, a reconstruction algorithm that requires very few X-ray projections is desirable. Earlier work showed that guidewires, stents and coils, which are commonly used in interventions, can be reconstructed using only four synthetic X-ray projections. The reconstruction from two or three X-ray projections was only studied briefly. In this work, we improve the method by using a more suitable neural network architecture and by using a multi-channel backprojection instead of a single-channel backprojection. We then apply the improved method to more realistic data measured in an anthropomorphic phantom. The results show that the method produces 3D reconstructions of stents and guidewires with submillimeter accuracy using only three measured X-ray projections.

Keywords: 3D fluoroscopy, sparse view, few view, X-ray, CT, deep learning, stent reconstruction, guidewire reconstruction, 4D, interventional guidance, interventional imaging

1. INTRODUCTION

Approaches for 3D reconstruction of interventional tools from few X-ray projections can be divided into two categories: algorithms specialized in the reconstruction of a single type of interventional tool and more general algorithms capable of reconstructing different types of tools. Belonging to the first category, many algorithms for the 3D reconstruction of curvilinear structures, like guidewires or catheters, have been proposed. A single guidewire^{1,2} or catheter² can be reconstructed from one X-ray projection, if a prior 3D dataset showing the patient's vasculature is available. Without a prior 3D dataset, a single guidewire^{3,4} or catheter^{3,5} can be reconstructed from two X-ray projections. An approach for computing a 3D representation of a stent from a single X-ray projection has been proposed.⁶ Since it requires a 3D model of the stent as prior knowledge, it is not a 3D reconstruction algorithm, but rather a registration algorithm. To the best of our knowledge, no algorithm specializing in the 3D reconstruction of stents from few X-ray projections has been proposed. In the second category of more general tool reconstruction algorithms, the reconstruction of guidewires and stents from about 16 X-ray projections has been demonstrated.⁷⁻¹⁰ Recently, the reconstruction of guidewires, stents and coils from only four X-ray projections has been demonstrated on synthetic data.^{11,12} In this work, we improve this algorithm and demonstrate on measured data of an anthropomorphic phantom, that guidewires and stents can be reconstructed from only three X-ray projections.

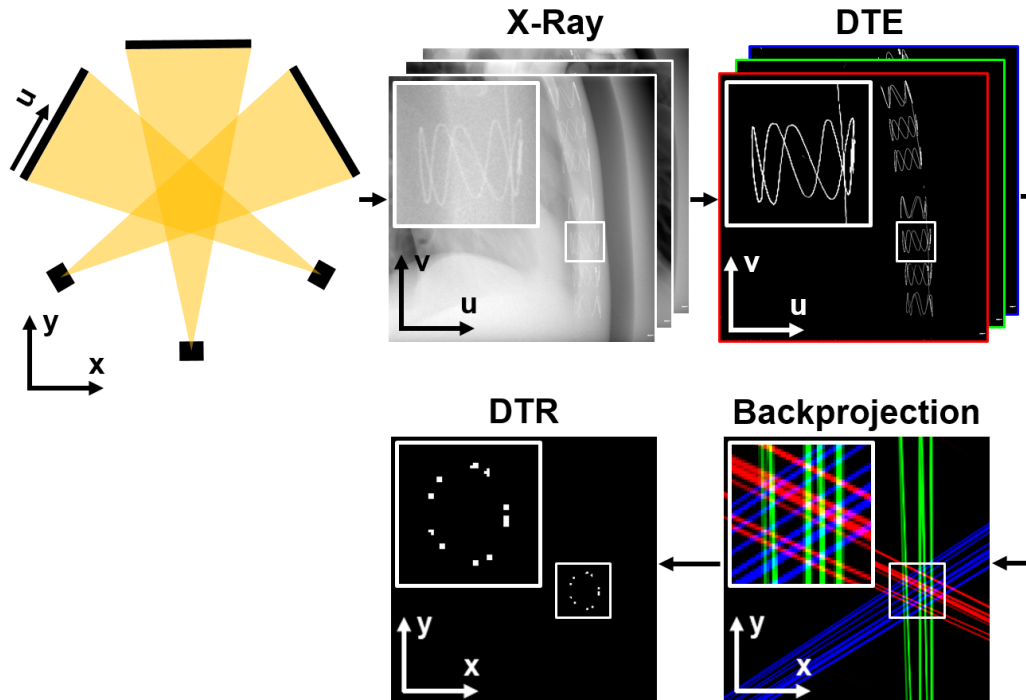


Figure 1. Sketch of the tool reconstruction pipeline. First, three X-ray projections are acquired with an angular increment of 60° . Then, DTE is applied to each X-ray projection. The DTE output images are backprojected into separate channels (illustrated by different colors) of a backprojection volume. The backprojection is then fed into DTR, which outputs a segmentation of the tools (only one transversal slice of the backprojection and the DTR output are shown).

2. METHODS

The tool reconstruction pipeline, which is similar to the one proposed by Eulig et al.,¹² is outlined in Figure 1. Three X-ray projections, offset by 60° , are acquired simultaneously. The deep tool extraction (DTE) algorithm extracts the tools from each projection. The output images of DTE are then backprojected into a volume. Finally, the deep tool reconstruction (DTR) algorithm transforms the backprojection into a 3D reconstruction of the tools. In this work, we improved DTR in two ways. First, we replaced the 2.5D convolutional neural network (CNN), which performed DTR in reference,¹² with a more suitable 3D CNN. Secondly, we backproject the three DTE output images into separate channels of the backprojection volume, which improves the reconstruction quality significantly.

2.1 Deep Tool Extraction

The task of DTE is to output the pixel-wise line integral of the X-ray attenuation coefficient of the guidewires and stents in the input projection. Training and validation data were generated online by combining forward projections of simulated guidewires and stents with clinical cone-beam computed tomography (CBCT) projections containing patients without interventional tools. In total, 12,000 guidewire projections, 12,000 stent projections and 2751 clinical projections from nine patients were used. The data were augmented by combining random tool patches and random patient patches, and by simulating varying levels of blur, noise and scatter. DTE was implemented in TensorFlow¹³ as a 2D CNN similar to the U-Net.¹⁴ In each resolution stage of the encoder and decoder, two (3×3 convolution + batch normalization¹⁵ + ReLU)-blocks are performed. The number of feature maps of the convolution layers in the n th resolution stage is 32×2^n where n ranges from 1 (highest resolution, 1024×1024) to 7 (lowest resolution, 16×16). Downsampling is performed by 2×2 max pooling, upsampling by nearest-neighbor interpolation followed by a 3×3 convolution layer. The mean absolute error was used as the loss function. Training took 200 epochs, where each epoch consisted of 16,000 training pairs and 4000 validation pairs. Each pair consisted of an input patch and an output patch of size 384×384 . The Adam optimizer¹⁶ (learning rate = 1×10^{-5} , $\beta_1 = 0.9$, $\beta_2 = 0.999$) and a mini-batch size of 24 were used.

2.2 Deep Tool Reconstruction

The task of DTR is to transform the input backprojection volume into a 3D segmentation of the tools. Training and validation data were generated by simulating forward projections of simulated 3D models of guidewires and stents. The backprojection of these projections was used as input volume, a binary voxelization of the 3D models was used as target volume. To make DTR insensitive to errors of DTE, i.e. false-positives and false-negatives, such errors were simulated into the forward projections prior to backprojection. The 3D guidewire models were simulated as curved cylinders around center lines represented by splines with random-walk-generated control points. The 3D stent models were simulated by stacking cylindrical strut segments along their central axis and subsequently bending the stent along a spline. Segment diameter, segment height, number of strut oscillations per segment, strut thickness, strut pattern within the segments (e.g. sinusoidal, zigzag, ...), number of stent segments, axial offset between segments, and bending spline were varied randomly. We simulated aortic stents with diameters between 10 mm and 30 mm.

The simulated X-ray system has three imaging threads (60° inter-thread angle) with a CBCT-like projection geometry: each thread consists of a point-source and a flat detector (1024^2 pixels, 0.3 mm pixel pitch, source-detector distance $R_{FD} = 1100$ mm, source-isocenter distance $R_F = 600$ mm). For the simulation of the training and validation data and for the application to measured data, a grid of 256^3 voxels of size $(0.6 \text{ mm})^3$ was chosen. To save disk space, since training was performed on patches of size 128^3 anyway, only a 128^3 -patch of each simulated 256^3 -volume was stored (the full dataset would require 840 GB). The patches were augmented online by z -axis-flips and rotations around the z -axis by integer multiples of 90°. The dataset consisted of 40,000 scenes, each featuring one stent and one or two guidewires. The dataset was split 70/30 between training and validation.

DTR was implemented in TensorFlow as a 3D CNN similar to the 3D U-Net.¹⁷ In each resolution stage of the encoder and decoder, two ($3 \times 3 \times 3$ convolution + batch normalization + ReLU)-blocks are performed. The number of feature maps of the convolution layers in the n th resolution stage is 8×2^n where n ranges from 1 (highest resolution) to 5 (lowest resolution). Downsampling is performed by $2 \times 2 \times 2$ max pooling, upsampling by nearest-neighbor interpolation followed by a $3 \times 3 \times 3$ convolution layer. The soft Dice loss with Laplace smoothing was used for training. Training was performed for 150 epochs, where each epoch consisted of 16,000 training pairs and 4000 validation pairs. Each pair consisted of an input patch and an output patch of size 128^3 . The Adam optimizer (learning rate = 1×10^{-4} , $\beta_1 = 0.9$, $\beta_2 = 0.999$) and a mini-batch size of 8 were used.

2.3 Phantom Measurements

Results will be shown on measured data of an anthropomorphic X-ray phantom (PBU-50, Kyoto Kagaku Co. Ltd. Japan) with a soft-tissue-equivalent extension on the anterior side. The interventional tools were placed between phantom and extension. The X-ray projection geometry was the same as described in Section 2.2. The three required projections were selected retrospectively from a 3D scan with fine angular sampling. Results will be shown on three different scenes. The first scene contains two guidewires. The second scene contains one stent. The third scene contains one stent and one guidewire, which was placed inside the stent. All projections were acquired at 80 kV and 0.80 mA s per projection (scene 1) or 1.15 mA s per projection (scenes 2 and 3).

2.4 Evaluation on Measured Data

To assess the quality of the output Y of our 3D tool reconstruction pipeline on measured data, a ground truth 3D reconstruction GT is needed. For each of the aforementioned scenes, this ground truth was generated by thresholding a 3D reconstruction that was computed from the projections of the respective 3D scan using the algorithm of Feldkamp, Davis and Kress.¹⁸ To quantify the deviation of Y from GT , one could use the popular Sørensen-Dice coefficient (DSC). However, since the guidewires and stent struts are only a few (1–3) voxels in diameter, the DSC is very sensitive to whether a voxel near the surface of such a structure is classified as foreground or background. Because the classification of the surface-voxels in the ground truth is uncertain (it is very sensitive to the threshold used to generate the ground truth), another metric is needed. We therefore decided to use the average Euclidean distance between the voxels of a skeleton of Y and the voxels of a skeleton of GT , $\bar{D}_{S(Y),S(GT)}$ and the average Euclidean distance between the voxels of a skeleton of GT and the voxels of a skeleton of Y , $\bar{D}_{S(GT),S(Y)}$. Skeletonization was used to make the metrics less sensitive to the diameter of the guidewires and stent struts. This is desirable, because the diameter in the ground truth is uncertain (as explained

above) and because the exact diameter would likely not matter for interventional guidance. Skeletonization was performed by the `skeletonize_3d` function from `scikit-image 0.17.2`,¹⁹ which implements the algorithm proposed by Lee et al.²⁰

2.5 Real-Time Capability

Inference of DTE on a 1024^2 projection takes about 40 ms, inference of DTR on a 3×256^3 volume takes about 260 ms on an NVIDIA RTX 3090 GPU. These times were measured in TensorFlow 2.5.0 using mixed precision, graph execution and with the input data already in GPU memory when starting the timer.

3. RESULTS

The training of DTR was performed twice to investigate whether passing the backprojections of the three DTE output images separately (3-channel backprojection) to DTR is advantageous compared to passing a single volume, into which all three DTE output images were backprojected (1-channel backprojection). For the first training, the 3-channel backprojection was used. For the second training, the 1-channel backprojection, which was generated by summing the channels of the 3-channel backprojection, was used. After 150 epochs, training 1 resulted in a DSC of 78.2% / 77.5% on the training / validation dataset, while training 2 resulted in 73.5% / 72.3%.

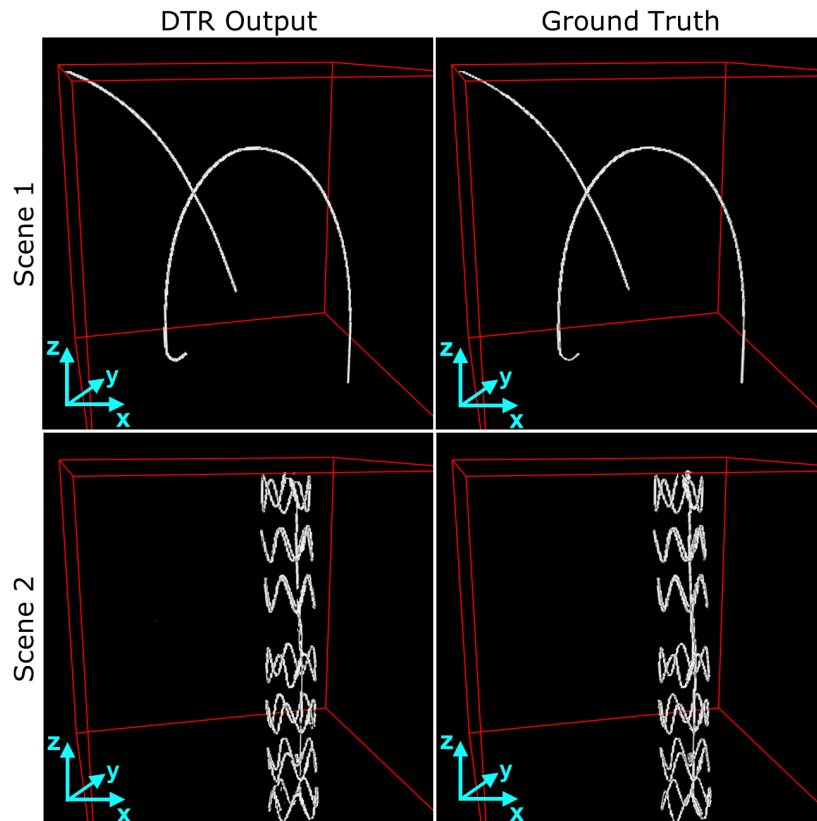


Figure 2. Volume-rendered DTR output and ground truth of the first and second scene.

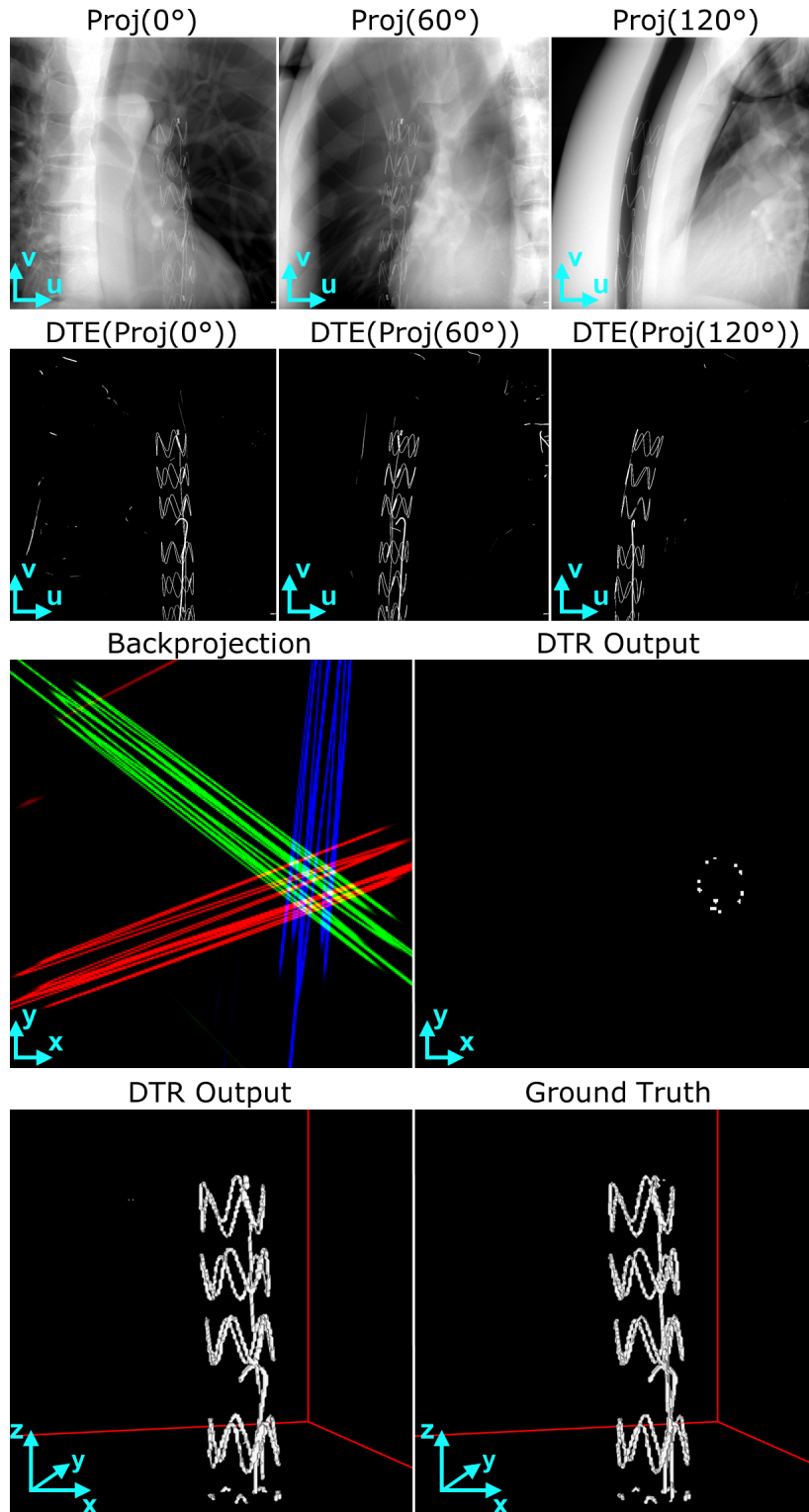


Figure 3. Inputs, intermediate results and outputs of the tool reconstruction pipeline on the third scene (see Section 2.3). First row: the input of the tool reconstruction pipeline are three X-ray projections separated by 60° . Second row: the outputs of DTE, when applied to the images of the first row. Third row: exemplary transversal slices of the backprojection of the DTE outputs and the output of DTR when applied to said backprojection. Fourth row: volume-rendered DTR output and ground truth.

	Scene	$\bar{D}_{S(Y),S(GT)}$	$\bar{D}_{S(GT),S(Y)}$
3-channel backprojection	1	0.27 mm	0.26 mm
	2	0.36 mm	0.28 mm
	3	0.31 mm	0.33 mm
1-channel backprojection	1	0.32 mm	0.31 mm
	2	0.50 mm	0.35 mm
	3	6.20 mm	0.44 mm

Table 1. Quantitative evaluation of the deviation of the tool reconstruction from the ground truth using the metrics defined in Section 2.4.

The tool reconstruction pipeline was applied to the measured projections of the three scenes described in Section 2.3. The volume-rendered tool reconstructions look very similar to the ground truths, as can be seen in Figure 2 (scenes 1 & 2) and Figure 3 (scene 3). The red box is a cube of size $256 \times 0.6 \text{ mm} = 153.6 \text{ mm}$, indicating the extent of the reconstructed volume. For scene 3, inputs and intermediate results are shown in Figure 3. The deviation of the tool reconstruction from the ground truth was quantified using the metrics defined in 2.4. The results are shown in Table 1. For all scenes, the tool reconstruction using the 3-channel backprojection outperformed the one using the 1-channel backprojection. The value of $\bar{D}_{S(Y),S(GT)} = 6.20 \text{ mm}$ for the reconstruction using the 1-channel backprojection on scene 3 is due to outliers in the reconstruction.

4. NEW WORK TO BE PRESENTED

In this work, we improved an earlier method for computing 3D reconstructions of interventional tools from very few X-ray projections and demonstrate its performance on more realistic data, which were measured in an anthropomorphic phantom.

5. CONCLUSIONS

We demonstrated that our improved pipeline can produce 3D reconstructions of stents and guidewires with sub-millimeter accuracy from only three X-ray projections measured in an anthropomorphic phantom. Furthermore, we demonstrated that passing the backprojections of the three DTE output images as three separate input channels to DTR, significantly improves the reconstruction quality. The low number of X-ray projections required per 3D reconstruction and the straightforward adaptability to different types of tools makes this algorithm a promising candidate for implementing 3D fluoroscopic interventional guidance.

REFERENCES

- [1] T. van Walsum, S. A. M. Baert, and W. J. Niessen, "Guide Wire Reconstruction and Visualization in 3DRA Using Monoplane Fluoroscopic Imaging," *IEEE Transactions on Medical Imaging* **24**(5), pp. 612–623, 2005.
- [2] M. Brückner, F. Deinzer, and J. Denzler, "Temporal Estimation of the 3D Guide-Wire Position Using 2D X-Ray Images," *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2009*, pp. 386–393, 2009.
- [3] M. Wagner, S. Schafer, C. Strother, and C. Mistretta, "4D Interventional Device Reconstruction from Biplane Fluoroscopy," *Medical Physics* **43**(3), pp. 1324–1334, 2016.
- [4] S. Baert, E. van de Kraats, T. van Walsum, M. Viergever, and W. Niessen, "Three-Dimensional Guide-Wire Reconstruction from Biplane Image Sequences for Integrated Display in 3d Vasculature," *IEEE Transactions on Medical Imaging* **22**(10), pp. 1252–1258, 2003.
- [5] M. Hoffmann, A. Brost, C. Jakob, M. Koch, F. Bourier, K. Kurzidim, J. Hornegger, and N. Strobel, "Reconstruction Method for Curvilinear Structures from Two Views," *SPIE Medical Imaging*, p. 86712F, 2013.

- [6] S. Demirci, A. Bigdelou, L. Wang, C. Wachinger, M. Baust, R. Tibrewal, R. Ghotbi, H.-H. Eckstein, and N. Navab, “3D Stent Recovery from One X-Ray Projection,” in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2011*, G. Fichtinger, A. Martel, and T. Peters, eds., **6891**, pp. 178–185, Springer Berlin Heidelberg, Berlin, Heidelberg, 2011. Series Title: Lecture Notes in Computer Science.
- [7] J. Kuntz, B. Flach, R. Kueres, W. Semmler, M. Kachelrieß, and S. Bartling, “Constrained Reconstructions for 4D Intervention Guidance,” *Physics in Medicine and Biology* **58**(10), pp. 3283–3300, 2013.
- [8] J. Kuntz, R. Gupta, S. O. Schönberg, W. Semmler, M. Kachelrieß, and S. Bartling, “Real-Time X-Ray-Based 4D Image Guidance of Minimally Invasive Interventions,” *European Radiology* **23**(6), pp. 1669–1677, 2013.
- [9] B. Flach, M. Brehm, S. Sawall, and M. Kachelrieß, “Deformable 3D–2D Registration for CT and Its Application to Low Dose Tomographic Fluoroscopy,” *Physics in Medicine and Biology* **59**(24), pp. 7865–7887, 2014.
- [10] B. Flach, J. Kuntz, M. Brehm, R. Kueres, S. Bartling, and M. Kachelrieß, “Low Dose Tomographic Fluoroscopy: 4D Intervention Guidance with Running Prior,” *Medical Physics* **40**(10), p. 101909, 2013.
- [11] E. Eulig, J. Maier, N. R. Bennett, M. Knaup, K. Hörndler, A. S. Wang, and M. Kachelrieß, “Deep Learning-Aided CBCT Image Reconstruction of Interventional Material from Four X-Ray Projections,” in *SPIE Medical Imaging Conference Record*, **113121L:1-7**, Mar. 2020.
- [12] E. Eulig, J. Maier, M. Knaup, N. R. Bennett, K. Hörndler, A. S. Wang, and M. Kachelrieß, “Deep Learning-Based Reconstruction of Interventional Tools from Four X-Ray Projections for Tomographic Interventional Guidance,” *Medical Physics* **48**, pp. 5837–5850, Oct. 2021.
- [13] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Y. Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng, “Tensorflow: Large-Scale Machine Learning on Heterogeneous Systems,” 2015.
- [14] O. Ronneberger, P. Fischer, and T. Brox, “U-Net: Convolutional Networks for Biomedical Image Segmentation,” *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, pp. 234–241, 2015.
- [15] S. Ioffe and C. Szegedy, “Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift,” *Proceedings of the 32nd International Conference on Machine Learning* **37**, pp. 448–456, 2015.
- [16] D. P. Kingma and J. L. Ba, “Adam: A Method for Stochastic Optimization,” *3rd International Conference for Learning Representations*, 2015.
- [17] Ö. Çiçek, A. Abdulkadir, S. S. Lienkamp, T. Brox, and O. Ronneberger, “3D U-Net: Learning Dense Volumetric Segmentation from Sparse Annotation,” *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2016*, pp. 424–432, 2016.
- [18] L. A. Feldkamp, L. C. Davis, and J. W. Kress, “Practical Cone-Beam Algorithm,” *J. Opt. Soc. Am. A* **1**(6), pp. 612–619, 1984. Publisher: OSA.
- [19] S. van der Walt, J. L. Schönberger, J. Nunez-Iglesias, F. Boulogne, J. D. Warner, N. Yager, E. Guillard, and T. Yu, “Scikit-Image: Image Processing in Python,” *PeerJ* **2**, p. e453, June 2014.
- [20] T. C. Lee, R. L. Kashyap, and C. N. Chu, “Building Skeleton Models Via 3-D Medial Surface Axis Thinning Algorithms,” *CVGIP: Graphical Models and Image Processing* **56**(6), pp. 462–478, 1994.