In-Between Projection Interpolation in Cone-Beam CT Imaging using Convolutional Neural Networks

Samaa Dweek[a], Salam Dhou[*a], Tamer Shanableh[a]
[a]Department of Computer Science and Engineering, American University of Sharjah, Sharjah, United Arab Emirates

ABSTRACT

Respiratory-Correlated cone beam computed tomography (4D-CBCT) is an emerging image-guided radiation therapy (IGRT) technique that is used to account for the uncertainties caused by respiratory-induced motion in the radiotherapy treatment of tumors in thoracic and upper-abdomen regions. In 4D-CBCT, projections are sorted into bins based on their respiratory phase and a 3D image is reconstructed from each bin. However, the quality of the resulting 4D-CBCT images is limited by the streaking artifacts that result from having an insufficient number of projections in each bin. In this work, an interpolation method based on Convolutional Neural Networks (CNN) is proposed to generate new in-between projections to increase the overall number of projections used in 4D-CBCT reconstruction. Projections simulated using XCAT phantom were used to assess the proposed method. The interpolated projections using the proposed method were compared to the corresponding original projections by calculating the peak-signal-to-noise ratio (PSNR), root mean square error (RMSE), and structural similarity index measurement (SSIM). Moreover, the results of the proposed method were compared to the results of existing standard interpolation methods, namely, linear, spline, and registration-based methods. The interpolated projections using the proposed method had an average PSNR, RMSE, and SSIM of 35.939, 4.115, and 0.968, respectively. Moreover, the results achieved by the proposed method surpassed the results achieved by the existing interpolation methods tested on the same dataset. In summary, this work demonstrates the feasibility of using CNN-based methods in generating in-between projections and shows a potential advantage to 4D-CBCT reconstruction.

Keywords: Image-guided radiation therapy; 4D cone-beam CT (CBCT), respiratory motion, image interpolation; convolutional neural networks (CNNs), deep learning

1. INTRODUCTION

Respiratory motion introduces uncertainties in radiotherapy treatment of tumors in thoracic and upper-abdomen regions [1]. Image-guided radiotherapy (IGRT) uses imaging during radiation therapy to reduce these uncertainties and improve the accuracy of treatment delivery. Respiratory-Correlated (4D)-CBCT is an emerging IGRT technique used to account for respiratory motion [2]. In 4D-CBCT, projections are sorted into bins based on their respiratory phase and a 3D image is reconstructed from each bin [3–5]. However, 4D-CBCT has a limited applicability in the current radiation therapy practices, mainly because of the poor quality of the images that are reconstructed using an insufficient number of projections [6–8]. Several developments of 4D-CBCT reconstruction techniques have been proposed in the literature for image quality improvement [9]. Moreover, interpolating additional projections and using them in 4D-CBCT reconstruction has been suggested to improve the quality of the resulting images.

The topic of image interpolation has been studied in medical imaging literature [10–15]. These interpolation methods can be classified as traditional methods such as linear and spline interpolation [13,14], directional interpolation [10,11], motion-based and registration-based interpolation [12,15]. Deep learning and CNNs, in particular, are powerful tools that have demonstrated their superiority in many healthcare applications [16]. The objective of this work is to develop a CNN-based method for interpolating in-between projections to increase the total number of projections used to reconstruct 4D-CBCT images. The use of the CNNs in this work was driven by the need to improve the quality of 4D-CBCT images in order to improve the overall quality of radiotherapy treatment and motivated by the promising results of CNNs in the field of video frame interpolation. The rest of the paper is organized as follows. Section 2 discusses the materials and methods used. Section 3 presents and discusses the experimental results. Section 4 concludes the paper.

[*]sdhou@aus.edu

Samaa Dweek, Salam Dhou, and Tamer Shanableh "In-between projection interpolation in cone-beam CT imaging using convolutional neural networks", Proc. SPIE 12031, Medical Imaging 2022: Physics of Medical Imaging, 1203129 (4 April 2022); https://doi.org/10.1117/12.2611474

2. MATERIALS AND METHODS

In this work, a CNN-based method for generating in-between interpolated projections is proposed. The following sections 2.1 – 2.3 present the datasets used in this work, the proposed method, and the evaluation metrics.

2.1 Datasets

The proposed method were tested on a phantom dataset that simulates a free-breathing CBCT scan. In this work, the dataset was generated using 4D extended cardiac-torso (XCAT) phantom [17,18]. The dataset used for all experiments consists of projections forwarded from 3D images generated by XCAT phantom at six respiratory phases. The dataset consists of 360 projections taken at different angles over a 360-degree rotation. The projections are of size 512 × 512 pixels with pixel intensity values varying between 0 and 8.6122.

2.2 CNN-based Interpolation Method

The CNN-based interpolation method proposed in this work is inspired by the video frame interpolation work reported by Niklaus et al. [19]. Figure 1 shows the CNN architecture used in this work. As can be seen in the figure, the network takes two consecutive projections, $I1$ and $I2$, as input to the network, and for each output pixel it generates a pair of 2D kernel, $k1$ and $k2$, carrying extracted features information. To reduce the amount of memory consumption, the network is followed by four subnetworks that each estimate one of the 1D kernels carrying the extracted features information. Thus, each of $k1$ and $k2$ would be estimated as $<K1,v,K1,h>$ and $<K2,v,K2,h>$. Finally, the interpolated projection is generated by convolving the output kernels with their respective input projections and adding them as follows:

$$\hat{I} = \big((K1,v * K1,h) * I1\big) + \big((K2,v * K2,h) * I2\big). \qquad (1)$$

where $\hat{I}$ is the output projection, $I1$ and $I2$ are the input images to the network, $K1,v$ and $K2,v$ are the 1D vertical kernels, and $K1,h$ and $K2,h$ are the 1D horizontal kernel. In this work, a CNN that generates separable convolution kernels is utilized as it has a lower space and time complexity [19]. The CNN architecture is a U-net encoder-decoder network which consists of a contracting component, also referred to as the encoder, and an expanding component, also referred to as the decoder. The encoder extracts features by down-sampling via average pooling. The expanding component incorporates bilinear up-sampling to complete the dense prediction. Relu is the activation function used in all layers. As for the loss functions, we used and compared between sum of absolute difference, also known as L_1, and VGG-19 as per the recommendations and results generated in several related studies [19–21]. The kernel size used for the 1D kernels is 51 pixels. Although a bigger kernel size can accommodate for bigger movements, this kernel size was found to account for the breathing lung movement found in the datasets.

Holdout method is used to split the datasets into 80% for training and the remaining is preserved for testing the models. Triplets of projections $<I1$, ground truth projection, $I2>$ are extracted from the training dataset and used to train the model. Moreover, data augmentation approaches are adopted to maximize the gained knowledge similar to the work reported by Kartašev et al. [21]. Random vertical, horizontal, and +90- and -90-degree transformations are applied whenever a triplet is read. In addition to that, temporal order swaps of the first and third projections of triplets are applied to enhance and boost the dataset used in training the models.

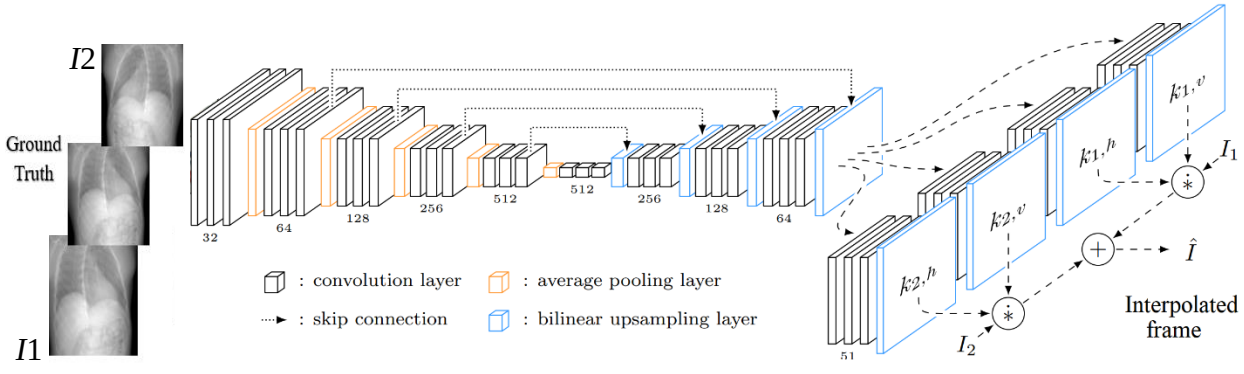


Figure 1. An overview of the CNN-based interpolation method. The network architecture is proposed by Niklaus et al. [19] and used in this work (reprinted with permission from IEEE). The input to the model in this work is triplets of x-ray projections $<I1$, ground truth projection, $I2>$ and output is the interpolated frame.

2.3 Evaluation Metrics

Several evaluation metrics are considered to examine the interpolated projections including peak signal-to-noise ratio (PSNR), root mean square error (RMSE), structural similarity index measurement (SSIM), and a difference image computed between the original and the interpolated projection. PSNR measures the quality difference between the original and the interpolated projection. The higher the value of the PSNR, the better the quality of the interpolated projection. PSNR is calculated as follows:

$$\boldsymbol{PSNR = 10.Log_{10}\left(\frac{MAX^2 i}{MSE}\right)}, \qquad (\mathbf{2})$$

where $\boldsymbol{MAX^2 i}$ is the maximum pixel value and the $\boldsymbol{MSE}$ is the cumulative mean square error between the interpolated projection *I* and the original ground truth projection *J* as shown in equation (3) below:

$$\boldsymbol{MSE = \frac{1}{mn}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}[I(i,j) - J(i,j)]^2}, \qquad (\mathbf{3})$$

where *i* and *j* are the pixels' intensity values of *I* and *J*, and *m* and *n* are the dimensions of the projections.

RMSE is calculated as the square root of MSE shown in equation (3). A low RMSE indicates a high-quality projection [22]. SSIM measures the level of similarity between the ground truth and the interpolated projections. It could be defined as the percentage of pixels in the interpolated projection that match the pixels in the ground truth one. Hence, the closer the SSIM value to 1.0, the better the interpolation results [23,24]. The SSIM of a projection *I* compared to the ground truth projection *J* is computed using the following formula:

$$\boldsymbol{SSIM(I,J) = \frac{(2\mu_I\mu_J + C_1)(2\sigma_{IJ} + C_2)}{(\mu_I^2 + \mu_J^2 + C_1)(\sigma_I^2 + \sigma_J^2 + C_2)}}, \qquad (\mathbf{4})$$

where *C*1 and *C*2 are constants, $\mu_I, \sigma_I, \sigma_{IJ}$ refer to the mean, standard deviation, and the cross-correlation, respectively.

The difference image contributes to the visual evaluation of the interpolated projections' quality. The difference image is a greyscale image that is calculated by simply subtracting the original projection from the interpolated one. The smaller the difference, the better the interpolation.

3. RESULTS

In this section, the experimental results of the proposed CNN-based interpolation method are presented. Moreover, the performance of the proposed method is compared to existing standard interpolation methods found in the literature.

3.1 Experiment Settings

The proposed CNN-based interpolation method was implemented in Python and using PyTorch library. The network parameters used for all training experiments, including AdaMax optimizer with $\beta1 = 0.9$, $\beta2 = 0.999$ and a learning rate of 0.001, were reported by Niklaus et al. [19]. MATLAB 2019, the MathWorks, Inc., Natick, Massachusetts, United States, was also used to complete some experiments and analyze the performance results.

3.2 Results of CNN-based Interpolation Method

Experiments were conducted using the phantom dataset described in Section 2.1. The trained model was tested using a testing set consisting of 60 projections, forming 20 triplets. In addition, we also experimented using different loss functions including L_1 and VGG. Table 1 summarizes the quantitative results of testing the trained models at different epochs up to 100 epochs. It was noticed that the model had already saturated after 100 epochs and started showing degradation in the quantitative results as well as the visual image quality.

Table 1. Quantitative results of the trained CNN model tested on 20 triplets – with L_1 loss function

Number of epochs	PSNR [dB]	RMSE	SSIM
1	20.8705	23.1879	0.6184
30	28.1205	15.7174	0.8364
60	34.2863	4.9597	0.9557
100	35.9386	4.1149	0.9577

Figures 2 shows a side by side comparison of a sample interpolated projection generated using the model trained for 100 epochs, the corresponding original projection, and a difference image between the two. As can be seen, the difference image is almost black which means the difference between the original projection and the interpolated one is minimal.

a) Original projection **b) interpolated projection** **c) difference image**

Figure 2. Visual results of the models trained using L_1 loss function for 100 epochs at angle 301 degrees.

Another experiment was conducted in which the VGG loss function was used instead of the standard baseline L_1 function. All other parameters and training setup remained unchanged. Unfortunately, the results achieved in this experiment were worse than those obtained using the L_1 loss function. The best results achieved using VGG loss function were at epoch 100 where PSNR, RMSE, and SSIM were 29.3033, 16.2119, and 0.8302, respectively. Further experiments were run to evaluate the performance of the model to ensure that it is well-fitted. These experiments were accomplished by analyzing the training versus validation loss. Thus, the dataset was divided it into training, validation, and testing datasets where 60% of the dataset was used as the training set, 20% as the validation set, and 20% as the testing set. The training dataset consists of 216 images, 72 triplets, while the validation and testing datasets both consist of 72 images, 24 triplets. L_1 loss function was used to train the model. Figure 3 shows the training verses validation loss of the model. As shown in the figure, the training loss stabilized and did not show any further changes at the last few epochs. Moreover, it can be noticed that the validation loss and training loss reached to a point in which they are almost equal. These observations indicate that the trained model is well-fitted and no over-fitting or under-fitting were observed.

Figure 3. Training and validation loss plot

3.3 Comparison with the Existing Standard Interpolation Methods

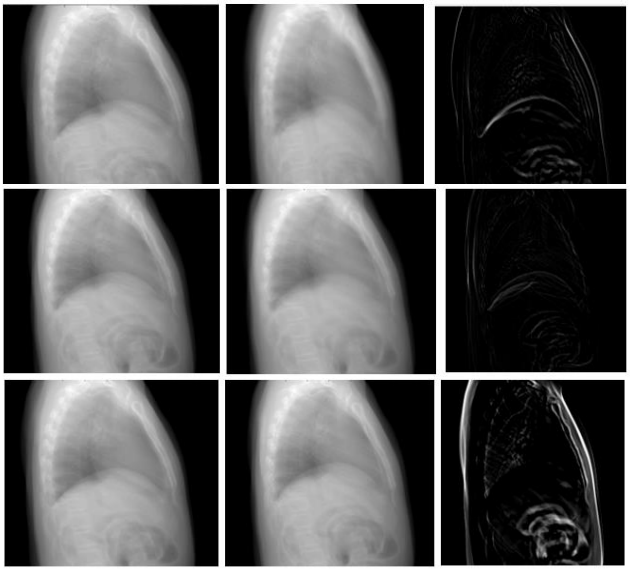
Three existing interpolation methods were implemented and compared with the proposed method, namely, linear [13], spline [14], and registration-based interpolation [15]. All experiments for the three methods were performed using the same dataset. For all the experiments, triplets of projections were selected such that the first and third projections are used to interpolate the middle projection. The second projection in the triplets was used as the ground truth.

Tables 2 summarizes the average PSNR results for the existing interpolation methods compared to the proposed method. The results in the table imply that none of the tested algorithms achieved good results. Figure 4 shows the visual results of applying linear, spline, and registration-based interpolation methods to interpolate projections and compare them to the original ones. The figures demonstrate visually the dissimilarity between the original projections and the interpolated ones represented by the white areas in the difference images.

The results of the linear and spline interpolation methods were expected since both methods interpolate the intensities of the intermediate projection with disregard to any motion. Surprisingly, the results of the registration-based method were not good either. This can be due to the assumption made in that study that all slices are parallel to the x-y plane [15]. However, this assumption cannot be true in this work since each of the projections are acquired at a different projection angles over a 360-degree rotation.

Table 2. PSNR results of the existing standard interpolation methods compared to the proposed method

Algorithm	**PSNR [dB]**
Linear [13]	25.398
Spline [14]	25.393
Registration-based [15]	20.087
Proposed CNN-based	35.939



a) Original projection **b) interpolated projection** **c) difference image**

Figure 4. Visual results of the standard interpolation methods: linear interpolation (top), spline interpolation (middle), and registration-based interpolation (bottom).

4. CONCLUSION

In this work, we proposed a CNN-based in-between projection interpolation approach to increase the number of projections that can be used to generate a high-quality reconstructed 4D-CBCT images. This approach is inspired by the promising results of CNNs in video frame interpolation. The resulting generated projections were assessed both quantitatively and qualitatively. The interpolated projections were evaluated by calculating several quantitative measurements including PSNR, RMSE, and SSIM. The achieved results were compared to those of standard interpolation methods available in the literature. The proposed method achieved better results than these methods which demonstrates the feasibility of using deep learning and CNNs in particular for generating in-between projections. The future work for this research includes using the interpolated projections to reconstruct the 4D-CBCT images and comparing these images to those reconstructed using the original projections only. In addition to that, applying the work to clinical datasets would be beneficial and can prove the clinical viability of the method.

REFERENCES

[1] Keall, P. J., Mageras, G. S., Balter, J. M., Emery, R. S., Forster, K. M., Jiang, S. B., Kapatoes, J. M., Low, D. A., Murphy, M. J., Murray, B. R., Ramsey, C. R., Van Herk, M. B., Vedam, S. S., Wong, J. W. and Yorke, E., "The management of respiratory motion in radiation oncology report of AAPM Task Group 76," Med Phys 33(10), 2006/11/09, 3874–3900 (2006).

[2] Sonke, J. J., Zijp, L., Remeijer, P. and van Herk, M., "Respiratory correlated cone beam CT," Med Phys 32(4), 2005/05/18, 1176–1186 (2005).

[3] Dhou, S., Motai, Y. and Hugo, G. D., "Local intensity feature tracking and motion modeling for respiratory signal extraction in cone beam CT projections," IEEE Trans. Biomed. Eng. 60(2), 332–342 (2013).

[4] Sabah, S. and Dhou, S., "Image-based extraction of breathing signal from cone-beam CT projections," Proc. SPIE - Int. Soc. Opt. Eng. 11315 (2020).

[5] Dhou, S., Docef, A. and Hugo, G., "Image-based respiratory signal extraction using dimensionality reduction for phase sorting in Cone-Beam CT Projections," ACM Int. Conf. Proceeding Ser., 79–84 (2017).

[6] Dhou, S., Alkhodari, M., Ionascu, D., Williams, C. and Lewis, J. H., "Fluoroscopic 3D Image Generation from Patient-Specific PCA Motion Models Derived from 4D-CBCT Patient Datasets: A Feasibility Study," J. Imaging 8(2) (2022).

[7] Guo, M., Chee, G., O'Connell, D., Dhou, S., Fu, J., Singhrao, K., Ionascu, D., Ruan, D., Lee, P., Low, D. A., Zhao, J. and Lewis, J. H., "Reconstruction of a high-quality volumetric image and a respiratory motion model from patient CBCT projections," Med. Phys. 46(8), 3627–3639 (2019).

[8] Dhou, S., Hurwitz, M., Mishra, P., Cai, W., Rottmann, J., Li, R., Williams, C., Wagar, M., Berbeco, R., Ionascu, D. and Lewis, J. H., "3D fluoroscopic image estimation using patient-specific 4DCBCT-based motion models," Phys. Med. Biol. 60(9), 3807–3824 (2015).

[9] Zhang, Y., Huang, X. and Wang, J., "Advanced 4-dimensional cone-beam computed tomography reconstruction by combining motion estimation, motion-compensated reconstruction, biomechanical modeling and deep learning," Vis. Comput. Ind. Biomed. Art 2(1) (2019).

[10] Zhang, H. and Sonke, J. J., "Directional interpolation for motion weighted 4D cone-beam CT reconstruction," Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics) 7510 LNCS (2012).

[11] Bertram, M., Wiegert, J., Sch??fer, D., Aach, T. and Rose, G., "Directional view interpolation for compensation of sparse angular sampling in cone-beam CT," IEEE Trans. Med. Imaging 28(7), 1011–1022 (2009).

[12] Dhou, S., Hugo, G. D. and Docef, A., "Motion-based projection generation for 4D-CT reconstruction," 2014 IEEE Int. Conf. Image Process. ICIP 2014 (2014).

[13] Lehmann, T. M., Gönner, C. and Spitzer, K., "Survey: Interpolation methods in medical image processing," IEEE Trans. Med. Imaging 18(11) (1999).

[14] Wang, J. W. and Chiu, C. Te., "Edge-based motion and intensity prediction for video super-resolution," 2014 IEEE Glob. Conf. Signal Inf. Process. Glob. 2014 (2014).

[15] Horváth, A., Pezold, S., Weigel, M., Parmar, K. and Cattin, P., "High order slice interpolation for medical images," Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics) 10557 LNCS (2017).

[16] Liu, Z., Yeh, R. A., Tang, X., Liu, Y. and Agarwala, A., "Video Frame Synthesis Using Deep Voxel Flow," Proc. IEEE Int. Conf. Comput. Vis. 2017-October (2017).

[17] Myronakis, M. E., Cai, W., Dhou, S., Cifter, F., Hurwitz, M., Segars, P. W., Berbeco, R. I. and Lewis, J. H., "A graphical user interface for XCAT phantom configuration, generation and processing," Biomed. Phys. Eng. Express (2017).

[18] Segars, W. P., Sturgeon, G., Mendonca, S., Grimes, J. and Tsui, B. M., "4D XCAT phantom for multimodality imaging research," Med Phys 37(9), 2010/10/23, 4902–4915 (2010).

[19] Niklaus, S., Mai, L. and Liu, F., "Video Frame Interpolation via Adaptive Separable Convolution," Proc. IEEE Int. Conf. Comput. Vis. 2017-October (2017).

[20] Niklaus, S., Mai, L. and Liu, F., "Video frame interpolation via adaptive convolution," Proc. - 30th IEEE Conf. Comput. Vis. Pattern Recognition, CVPR 2017 2017-January (2017).

[21] Kartašev, M., Rapisarda, C. and Fay, D., "Implementing Adaptive Separable Convolution for Video Frame Interpolation" (2018).

[22] Razzak, M. I., Naz, S. and Zaib, A., "Deep learning for medical image processing: Overview, challenges and the

future," [Lecture Notes in Computational Vision and Biomechanics] (2018).
[23] Horé, A. and Ziou, D., "Image quality metrics: PSNR vs. SSIM," Proc. - Int. Conf. Pattern Recognit. (2010).
[24] Asamoah, D., Ofori, E., Opoku, S. and Danso, J., "Measuring the Performance of Image Contrast Enhancement Technique," Int. J. Comput. Appl. 181(22) (2018).