

A five-dimensional cardiac CT model for generating virtual CT projections for user-defined bolus dynamics and ECG profiles

Eri Haneda*^a, Bernhard Claus^a, Jed Pack^a, Darin Okerlund^b,
Albert Hsiao^c, Elliot McVeigh^d, and Bruno De Man^a

^aGE Research, One Research Circle, Niskayuna, NY, USA 12309; ^bGE Healthcare, 3000 N. Grandview Blvd., Waukesha, WI, USA 53188; ^cDepartment of Radiology at University of California San Diego, 9500 Gilman Dr, La Jolla, CA, USA 92093; ^dDepartment of Bioengineering, Medicine, Radiology at University of California San Diego, 9500 Gilman Dr, La Jolla, CA, USA 92093

ABSTRACT

Cardiac CT exams are some of the most complex CT exams due to need to carefully time the scan to capture the heart during the quiescent cardiac phase and when the contrast bolus is at its peak concentration. We are interested in developing a robust and autonomous cardiac CT protocol, using deep learning approaches to extract contrast timing and cardiac phase timing directly from pulsed projections. In this paper, we present a new approach to generate large amounts of clinically realistic virtual data for training deep learning networks.

We propose a five-dimensional cardiac model generated from 4D cardiac coronary CT angiography (CTA) data for synthetic contrast bolus dynamics and patient ECG profiles. We apply deep learning to segment 7 heart compartments and simulate intravenous contrast propagation through each compartment to insert contrast bolus. Additional augmentation techniques by randomizing a bolus curve, patient ECG profile, acquisition timing, and patient motion are applied to increase the amount of data that can be generated. We demonstrate good performance of the deep learning segmentation network, examples of simulated bolus curves using a realistic protocol, and good correspondence between virtually generated projections and real projections from patient scans.

Keywords: Cardiac CTA, CT simulation, deep learning, contrast bolus, cardiac phase

1. INTRODUCTION

Cardiac CT exams such as Coronary CT Angiography (CTA) are some of the most complex CT exams due to need to carefully time the scan to capture the heart during the quiescent cardiac phase (when the heart is moving least) and when the contrast bolus in the heart chambers is at its peak concentration to achieve good contrast enhancement.

Timing the CT scan to coincide with the peak contrast concentration can be done using a ‘timing bolus’ or with ‘bolus tracking’. With a timing bolus, a small volume of contrast is injected to a patient in a pre-session and repeated single slice axial collimated low-dose scans are performed to establish the timing delay between the start of injection and peak enhancement. Then the diagnostic cardiac CTA exam is performed with the full contrast bolus and the CT scan start after this timing delay. With bolus tracking, there is no pre-session: the full contrast bolus volume is injected, and single slice axial collimated scans are performed until the CT number in a region-of-interest reaches a predefined threshold. During the following ‘diagnostic delay’ of several seconds, the scan table is repositioned, breath hold instructions are delivered, and the scanner is reconfigured, after which the diagnostic CTA scan is performed. Both approaches have pros and cons, and require highly trained operators to achieve consistent bolus enhancement.

The quiescent phase of the cardiac cycle is typically estimated based on recordings from an ECG monitor and evaluated by visually assessing the degree of motion artifact on reconstructed cardiac CTA images. Based on the ECG R-peaks, the time of the next end-diastolic phase (or end-systolic phase for higher heart rates) is estimated [1]. Application of the ECG may take several minutes per exam and in some patients may lead to reliability problems, such as due improper lead positioning.

Our overall project goal is to develop a smart cardiac CT scanner that autonomously determines the optimal scan time interval without ECG, traditional bolus tracking or timing bolus, but based on real-time deep learning analytics of sparsely

pulsed projections [2, 3]. However, one of the challenges to developing deep learning algorithms is collecting enough data to train without exposing volunteers to ionizing radiation. To address this challenge, we here present a new approach to generate virtual CT projection data at any view angle, bolus dynamics, and cardiac phase, based on a five-dimensional model of the cardiac CT volume, derived from retrospectively collected clinical images and using a series of data augmentation steps.

2. METHODS

Figure 1 shows an overview of our virtual data generation scheme. We first derive five-dimensional cardiac CT models from multi-phase clinical cardiac (cine) CT scans by segmenting the heart compartments and identifying a blood (or contrast) flow propagation map in each compartment. To model contrast concentration at multiple bolus time points from datasets that were acquired at (approximately) a single bolus time point, we segment the cardiac compartments, we parametrize the voxels inside those compartments based on their location along the flow direction, and then increment the voxel values to model different bolus distributions based on location and time point. We then define multiple instantiations of CT exams based on specific timing of cardiac cycle, contrast bolus, and CT acquisition to generate virtual CT projection data. Each of the steps is described in detail in the next paragraphs.

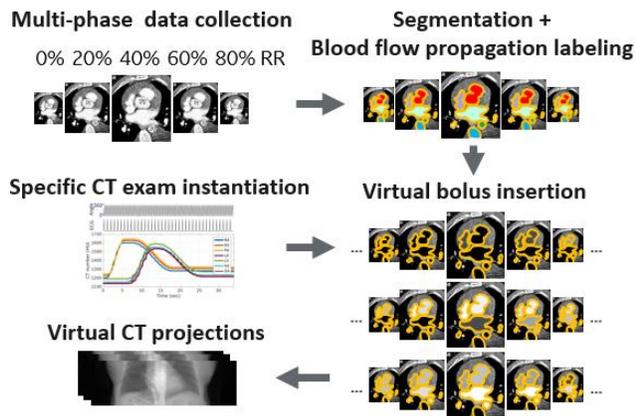


Fig. 1: Overview of our virtual data generation scheme

Clinical datasets: We retrospectively collected multiple phase (cine) cardiac CT data from 40 Transcatheter Aortic Valve Replacement (TAVR) patients. The data were acquired under IRB approval (IRB #191797X) at University of California San Diego using a GE Revolution CT scanner with a 16 cm z-coverage to image the whole heart in one rotation. ECG data was acquired simultaneously, and iodine contrast agent (bolus) was administered to each patient. Cine scan mode was used to image all cardiac phases, resulting in 1.1-1.4 sec scan times and covering at least one full R-R cycle. Phase-specific cardiac volumes were retrospectively reconstructed every 70ms. The image volume size was 512x512x256 voxels covering the full 50-cm-diameter field-of-view and 16 cm in the z direction. Approximately 15-20 volumes (phases) were reconstructed for each patient and the associated R-R% was extracted from the ECG. By interpolation, a cardiac CT image dataset can be extracted at any R-R%, which represents the fourth dimension in the five-dimensional model.

Segmentation: A subset of 149 datasets representing multiple phases were selected from 28 different patients. For each dataset, 7 compartments were manually segmented: right atrium (RA), right ventricle (RV), pulmonary artery (PA), left atrium (LA), left ventricle (LV), ascending aorta (AA), and descending aorta (DA), resulting in 149 3D masks. The PA region was further segmented into three regions: main pulmonary artery (MPA), left pulmonary artery (LPA), and right pulmonary artery (RPA). Any overlap between two adjacent compartments is removed by assigning the overlap voxels to one of the compartments. Any gaps between two adjacent compartments are avoided by inserting a 3D cylinder padding between them. Figure 2 shows an example of the resulting compartment segmentations with color coding.

Blood flow propagation labeling: We defined blood flow propagation labels by parametrizing the voxels inside the compartments based on their position along the blood flow (or bolus) propagation direction. The label assigned to each voxel are integer numbers that approximately represent how the blood propagates from RA to RV, and PA, then from LA to LV, to AA, and to DA. The propagation direction is determined by multiple reference points: the center-of-mass locations of each compartment and a set of touch points where

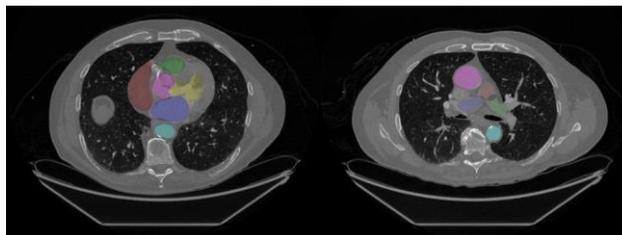


Fig. 2: An example of manual segmentation. The left image contains right atrium (red), right ventricle (green), ascending aorta (pink), left atrium (purple), left ventricle (yellow), and descending aorta (light blue). The right image contains ascending aorta (pink), main pulmonary artery (light red), left pulmonary artery (light green), and right pulmonary artery (light purple), and descending aorta (light blue).

each pair of compartments touch each other. From the manual segmentation masks, we first find touch points between RA-RV (RAV touch), RV-PA (ROV touch), LA-LV (LAV touch), and LV-AA (LOV touch), where O refers to an outgoing vessel from a ventricle.

We developed two blood propagation models depending on the compartment. For RA, LA, PA, AA, and DA, a ‘linear propagation model’ is used. For **RA**, we assign linearly increasing values along the line which passes from the center-of-mass of the atrium (RA_{ctr}) to the RAV touchpoint. Then, each voxel is labeled based on its projected location on the line, normalized from 100 to 199, such that the blood flow mask value increases linearly in the direction from 100 at the entrance to 199 at the exit of the atrium. Similarly, the **LA** voxels are labeled from 400 to 499. For **PA**, we find the end of the LPA branch and the end of RPA branch by identifying maximum distance from ROV touch. Then, voxels are labeled linearly from 300 at ROV touch to 399 at the end of the LPA and the RPA. For **AA**, the direction for linear propagation is defined by the line from LOV touch through the center of the AA (AA_{ctr}) and voxels are labeled from 600 to 699. For **DA**, assuming that DA is a straight vessel running in z direction, the voxels are labeled from 700 at the maximum z to the pixel to 799 at the minimum z position. For **RV** and **LV**, an ‘angular propagation model’ was developed. First a rotation center point was defined outside of the ventricle, triangulated from the center-of-mass of the ventricle (V_{ctr}) and the two adjacent touch points. Then an angle is computed for each voxel relative to that line, encoding a circular path. Voxels are labeled from 200 to 299 in the RV and from 500 to 599 in the LV.

Blood flow propagation labels were computed for all 149 segmented volumes and will be used for deep learning network training and validation, as described in the next section.

Deep learning segmentation and blood flow propagation labeling: We trained a 3D Unet [4] (Figure 3) to perform both segmentation and voxel labeling to model the blood propagation direction. The input to Unet are patient images, truncated to a 30cm field-of-view and then downsampled to (152x152x64). The network has two output of the same size. The first output is the probability prediction for a voxel to be inside the bloodpool (i.e. one of the 7 compartments). The second output volume is the voxel label prediction. Finally, both output volumes are combined by setting zero to all voxel coordinates that have less than 50% probability, and the resulting volume is upsampled to the original voxel size.

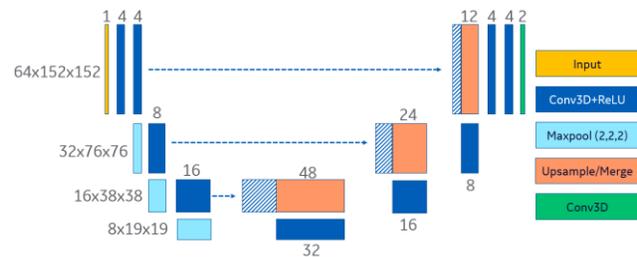


Fig. 3: Neural network architecture to generate a blood flow mask

A two-part loss function was used with one part being a mean squared error on the voxel labels. Outside the ground truth mask, this error was set to zero. The second part of the loss function was a piecewise linear function on the absolute error of the mask output. This function has three sections, with the steep portion residing between 0.4 and 0.6.

Bolus dynamics definition: The obtained volumes are preprocessed to remove the estimated real average bolus level from all compartments and then adding in the simulated bolus level. A simulated bolus time sequence $B(t)$ is defined by the parameters in Figure 4. The bolus curve remains 0 HU for a given *Bolus Delay*, rises from 0 HU to *Bolus Height* during the *Bolus Rise Time* (with a cosine shape), remains at *Bolus Height* for the *Bolus Peak Width*, decays from *Bolus Height* to *Converged Bolus Level* during the *Bolus Decay Time* (with a cosine shape), and remains at *Converged Bolus Level* for the remainder of the time.

The new bolus level is assigned to voxels based on the current simulated scan time and the delay associated with each voxel location. The time delays at the compartment boundaries are set to 0RR, 0.5RR, 1.0 RR, and 1.5RR for the right side of the heart, where RR refers to the R-to-R interval. The pulmonary circulation delay is then added before re-entering the left heart side, after which the following set of additional delays are applied 0RR, 0.5RR, 1.0 RR, 1.5RR and 1.8RR for the compartment boundaries on the left side of the heart. To implement dispersion of bolus in the left cardiac region relative to the right cardiac region, a trapezoid filter is applied. That means the bolus levels for RA, RV, and PA follow the original

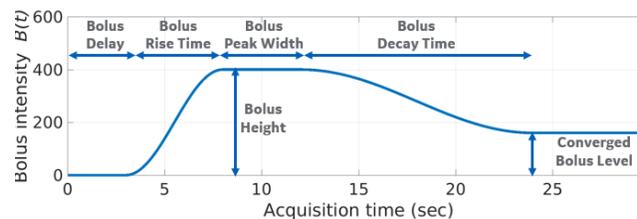


Fig. 4: Bolus curve definition by parameters

bolus curve, and the bolus levels for LA, LV, AA, and DA follow the dispersed bolus curve. We also scaled all delays by a random number between [0.3 0.55]. For this particular study we also averaged the bolus curve over each period between two time points at neighboring compartment boundaries and assigned the average value uniformly to each compartment. In future work, this model will be refined to better model realistic behavior in discussion with experts in cardiac blood flow.

Protocol instantiation: We simulate series of acquisitions by randomly selecting a patient, acquisition time points, cardiac conditions, bolus curve parameters, and patient rigid motion parameters. In this study, only anterior-posterior (AP) and posterior-anterior (PA) view acquisitions are simulated with 125 rotations. The rotation time is set to 0.28 sec, therefore data is acquired every 0.14 sec and the total scan time is 35 sec. For each instantiation, we randomly select a bolus curve and a cardiac condition. Table 1 and 2 summarize the value range of bolus curve parameters and cardiac motion parameters. For each acquisition time point t_{aq} , we find two closest R-peaks time (t_{R1}, t_{R2}) from the patient’s ECG to compute $RR\% = (t_{aq} - t_{R1}) / (t_{R2} - t_{R1})$. We then interpolate a new dataset from the original patient dataset based on the respective RR%. In this study, we used nearest neighbor interpolation.

Extrapolation, Augmentation, and Reprojection:

The voxels outside cone beam are extrapolated from the neighboring slice if available. Extra z slices are also padded on the top and bottom by repeating the boundary slice to prevent boundary artifacts during forward-projection. Then, a random rigid motion augmentation was performed per patient by rotation, scaling and shifting to add more anatomy variation. Finally, we forward-projected the volumes using a distance-driven projector [5] and modeling a GE Revolution CT cone beam geometry.

Table 1: Bolus curve parameters

Parameter	Value range
Bolus Delay	[0.5, 3.5] sec
Bolus Rise Time	[2, 12.5] sec
Bolus Peak Width	[1.0, 7.0] sec
Bolus Decay Time	[7.0, 15.0] sec
Converged Bolus Level	[15, 65] % of peak height
Bolus Height	[120 180] %
Trapezoid filter full-width	[60, 95] % of pulmonary circ delay

Table 2: Cardiac motion parameters

Parameter	Value range
Heart rate	[40, 75] bpm
RR% offset at scan start	[0, 1]
Pulmonary circulation delay	[4.0, 9.0] sec

3. RESULTS

Figure 5 shows a specific CT exam instance as a function of time. The top row shows the CT gantry rotation angle. The second row shows the patient ECG signal. The 7 colored curves show CT number averaged over each of the 7 cardiac compartments. For each curve, we can clearly observe a rising edge, a plateau, and a more gradual decay. The large delay between compartments in the right left sides of the heart is due to the pulmonary circulation. For the same reason there is some additional dispersion in the curves for the left side of the heart. The bottom row shows the time of the injection, the start of the pulsed-mode projections (PMPs), the breath-hold command, and the actual CTA scan. Note that the gantry rotation angle, ECG signal, and PMPs are shown for illustration purpose. The spacing is not exact.

Figure 6 shows example CT images and the corresponding compartment segmentation and blood flow propagation labels. The color coding reflects the integer labels from 100 to 799.

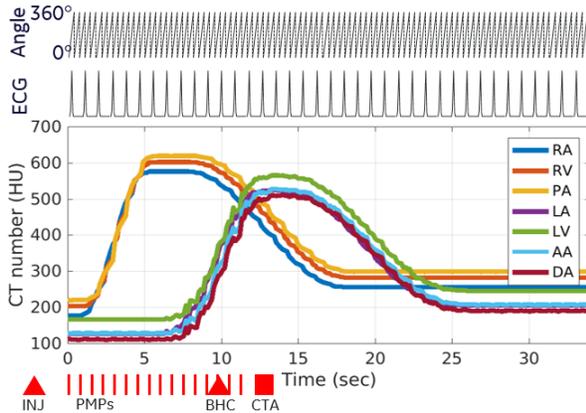


Fig. 5: Example of a CT exam instance as a function of time. From top to bottom: gantry rotation angle, ECG signal, average CT number in the 7 cardiac compartments, and the timing of injection (INJ), pulsed-mode projections, (PMPs), breath-hold command (BHC), and CTA scan.

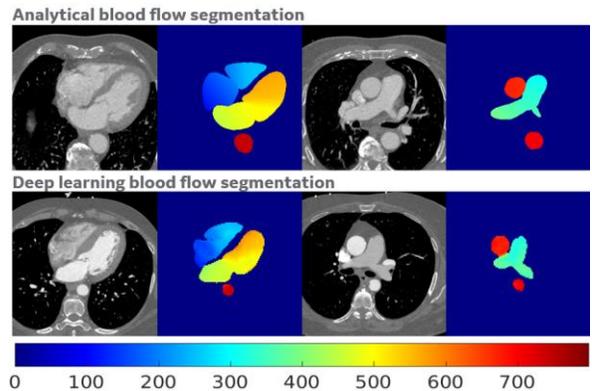


Fig. 6: Four example CT images and corresponding segmentation and blood flow propagation labels: the two training examples show the analytically computed labels (top row) and the two test examples show the inferred values (bottom row).

The top row shows two examples of CT images and the corresponding the labels, which were analytically generated. The bottom row shows two test CT images with the inferred segmentation and integer labels. The cardiac regions were quite accurately segmented, and the blood flow propagation labels in seven compartments were visually similar to the training examples from the top row.

Figure 7 shows a sequence of AP projections, comparing real measurements (left) and virtual projections (right). The top row shows the first actual projections in the sequence. The next 4 rows show the difference images of the subsequent projections, i.e.: subtracting the first projection in the respective sequence. In the real data, the heart rate was 58.5 bpm and the cardiac phase was 0.89, 0.17, 0.44, 0.71, and 0.99 %RR from the top to bottom. The simulated data was generated using used these same parameters. Bolus insertion and motion augmentation were skipped. The simulated projections visually match the real projections quite well. More work is needed to verify the accuracy of the virtual bolus insertion.

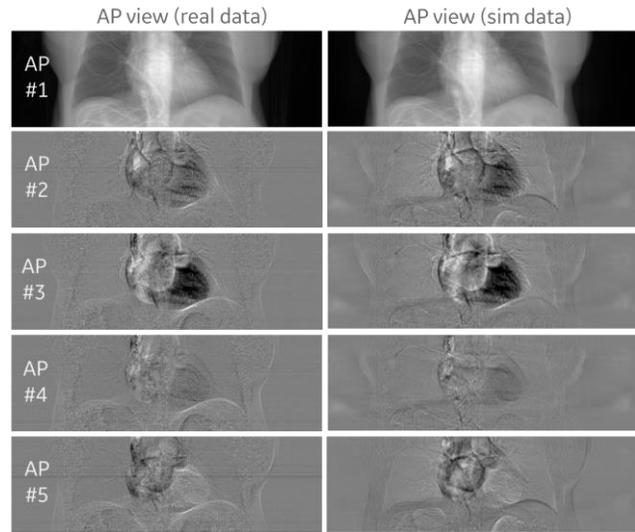


Fig. 7: Comparison between real projections (left) and simulated projections (right) for a sequence of AP projections. The first row in the sequence shows the first AP projection. The next rows show subsequent AP projections after subtracting the first projection in the sequence.

4. CONCLUSION

In this paper, we proposed an innovative method to generate large amounts of virtual but clinically realistic cardiac CT projection data. Our approach used a five-dimensional model of the cardiac CT volume derived from multi-phase 3D cardiac CT images with programmable bolus dynamics and cardiac phases. This dataset could be used to train a real-time deep learning network which determines the optimal scan time from raw data without ECG and traditional bolus tracing or timing bolus.

ACKNOWLEDGEMENT

Research reported in this publication was supported by the NIH/NHLBI grant R01HL153250. The content is solely the responsibility of the authors and does not necessarily represent the official views of the NIH.

REFERENCES

- [1] B. Ohnesorge and T. Flohr, “Principles of Multi-slice Cardiac CT Imaging,” in *Multi-slice and Dual-source CT in Cardiac Imaging*, 2nd ed., Springer, Berlin/Heidelberg, 2007, pp. 71–126.
- [2] B. De Man, E. Haneda, J. Pack, and B. Claus, “Data-based scan gating,” U.S. Patent 10736594, Aug. 11, 2020
- [3] J. Hsieh, S. Sirohey, R. Nilsen, S. Narayanan, and G. Cao, “Systems and methods for imaging phase selection for computed tomography imaging,” U S Patent 9517042, Dec. 13, 2016
- [4] O. Ronneberger, P. Fischer, and T. Brox, “U-Net: Convolutional Networks for Biomedical Image Segmentation,” arXiv:1505.04597
- [5] B. De Man and S. Basu, “Distance-driven projection and backprojection in three dimensions,” *Phys. Med. Biol.* 49(11), 2463–2475 (2004).