http://communications.science.ankara.edu.tr

Commun.Fac.Sci.Univ.Ank.Ser. A2-A3
Volume 64, Number 1, Pages 1–20 (2022)
DOI:10.33769/aupse.1020175
ISSN 1303–6009 E-ISSN 2618–6462

Research Article; Received: November 07, 2021; Accepted: January 04, 2022

3D RECONSTRUCTION OF CORONARY ARTERIES USING DEEP NETWORKS FROM SYNTHETIC X-RAY ANGIOGRAM DATA

İbrahim ATLI[1] and O. Serdar GEDİK[1]

[1]Department of Computer Engineering, Ankara Yildirim Beyazit University,
Ankara, TURKEY

Abstract. Cardiovascular disease (CVD) is one of the most common health problems that are responsible for one-third of all deaths around the globe. Although X-Ray angiography has deficiencies such as two-dimensional (2D) representation of three dimensional (3D) structures, vessel overlapping, noisy background, the existence of other tissues/organs in images, etc., it is used as the gold standard technique for the diagnosis and in some cases treatment of CVDs. To overcome the deficiencies, great efforts have been drawn on retrieval of actual 3D representation of coronary arterial tree from 2D X-ray angiograms. However, the proposed algorithms are based on analytical methods and enforce some constraints. With the evolution of deep neural networks, 3D reconstruction from images can be achieved effectively. In this study, we propose a new data structure for the representation of objects in a tubular shape for 3D reconstruction of arteries using deep learning. Moreover, we propose a method to generate synthetic coronaries from data of real subjects. Then, we validate tubular shape representation using 3 typical deep learning architectures with synthetic X-ray data we produced. The input to deep learning architectures is multi-view segmented X-Ray images and the output is the structured tubular representation. We compare results qualitatively in terms of visual appearance and quantitatively in terms of Chamfer Distance and Mean Squared Error. The results demonstrate that tubular representation has promising performance in 3D reconstruction of coronaries. We observe that convolutional neural network (CNN) based architectures yield better 3D reconstruction performance with 9.9e-3 on Chamfer Distance. On the other hand, LSTM-based network fails to learn the coronary tree structure and we conclude that LSTMs are not appropriate for auto-regression problems as depicted in this study.

Keywords. 3D Reconstruction, coronary artery tree, deep learning, synthetic coronary dataset.

✉ iatli@ybu.edu.tr-Corresponding author; gedik@ybu.edu.tr
iD 0000-0003-0393-2332; 0000-0002-1863-8614.

©2022 Ankara University

1. INTRODUCTION

Cardiovascular disease (CVD) is one of the serious health problems which may lead to death if not treated appropriately. According to the research conducted by World Health Organization (WHO), it is reported that an estimated 17.9 million people die (31% of all deaths worldwide) each year from CVDs [1]. Precise diagnosis and treatment have the potential to increase the quality of life of patients and decrease the rate of deaths. There exist many imaging methods utilized for the diagnosis of CVDs such as Magnetic Resonance Imaging (MRI), Digital Subtraction Angiography (DSA), Computed Tomography Angiography (CTA) and XRay Angiography (XRA). Among them XRA imaging modality is a gold standard technique used in clinics [2, 3]. However, XRA provides 2-dimensional (2D) images to monitor blockages, aneurysms, narrowing, malformations and other blood vessel problems in veins, arteries and organs in screens. Interventional cardiologists capture images from different views of angle [4] and examine these images to make a diagnosis. They decide on treatment planning by investigating the anatomical features of arteries. In some cases, it is possible to treat blocked vessels during the angiogram, but further examination is recommended after surgery to diagnose a variety of vascular problems including aneurysm, vascular malformations, deep vein thrombosis, aortic arc (problems in the arteries that branch off the aorta), peripheral artery disease and renovascular conditions [5]. These tasks are not trivial due to the following factors: (1) The images are generally in low quality and contrast; (2) XRA imaging is limited by inherent 2D representation of 3D structures. It creates 2D projection images and the projection operation causes a substantial amount of 3D/4D anatomy information of the coronary artery to be lost. (3) Cardiologists need to combine all images mentally in their minds to resolve the problem and its location; (4) the accurate diagnosis heavily relies on the experience of cardiologists, and (5) this creates a subjective evaluation; (6) Finally, the procedure of X-Ray imaging is not convenient to be repeated multiple times because it needs surgical intervention and injection of contrast (radioactive) material directly to arteries. Although tackling these problems manually is possible, it takes hours of processing and evaluation for each patient. It is a cumbersome, repetitive and operator-dependent task. Decreasing intra or inter-operator variability and improving the quality of diagnosis in terms of time and accuracy is possible with an automated tool. In this manner, a system generating 3D model of heart vessels can provide objective, reproducible and relatively easier evaluation. Such a system offers a new and accurate approach in diagnosis and has the potential to guide clinical decisions. Therefore, the development of automatic and accurate vessel-tree reconstruction from angiograms is highly desirable. The aim of this study is to develop a fast, robust and automated deep-learning (DL) tool for 3D reconstruction of coronary artery tree from X-Ray angiograms to provide clinical assistance.

3D reconstruction with deep learning methods is a topic of high interest and researchers improve algorithms and data structures as the number of 3D data-sets increases. We identify that previous works have only focused on analytical solutions for 3D reconstruction of coronary arteries as reviewed in Section 2. Although 3D reconstruction can be achieved effectively using deep learning, there is no 2D or 3D publicly available coronary artery data-set in literature. In this paper, we present a synthetic 3D data-set (5000 left, 5000 right coronary artery) and make following contributions:

- We propose a novel data structure for the representation of 3D tubular shapes.
- We have validated that our data structure works well in 3D reconstruction of coronaries using 3 basic deep networks both qualitatively and quantitatively.
- We introduce a method to generate synthetic 3D coronaries from 3D computerized tomography data of real subjects,
- We show deep networks can be utilized in 3D reconstruction of coronaries. This is the first study that uses DL in 3D reconstruction of coronaries,
- We demonstrate that long-short-term-memory (LSTM) is not a proper choice for auto-regression problems as in [6],
- The proposed method has the potential to be used in clinics due to its decent running-time.

The rest of the paper is organized as follows: Section 2 presents related work. Section 3 introduces the materials and proposed method. In Section 4, experimental results are presented. Finally, we draw our conclusions, discuss the limitation of the work and mention future directions in Section 5.

2. Related work

There are various 3D coronary reconstruction methods that stem from different types of coronary imaging systems. The studies presented in [3, 7–13] utilize traditional (single-plane) XRA and [14–18] utilize biplane XRA. We focus on these XRA methods and examine them into 2 main categories based on the technique they follow: (1) Calibrated and (2) non-calibrated (calibration-free) reconstruction. Then, we continue to present learning based methods specifically. Since 3D reconstruction of coronary arterial tree with DL has not been addressed in the literature, we review DL-based methods for only 3D object reconstruction. We present them in groups according to the 3D representation they use.

In calibrated methods, camera calibration matrix is formed in order to map points in 3D to their corresponding 2D projections. Calibration parameters are estimated by optimizers [3, 9–11] or using the phantom object on which metal balls are attached [19]. Centerlines [7–9, 14] or corresponding 2D points [3, 11] are used in determining calibration parameters and estimating 3D model of coronary artery tree. [3] and [9] follow the same approach in finding external parameters of the camera (R: rotation, t: translation) using constrained nonlinear optimization algorithm given five or more correspondence points. The works [9–11, 16] use epipolar matching and triangulation methods during 3D reconstruction of coronaries after calibration. Apart from these studies, [18] utilizes bundle adjustment method and [13, 20] utilize graph-cut based sparse stereo method for 3D reconstruction. The works of [7, 8] use deformable model and update contours using active contour algorithms with back-projection external energy. Similarly, [3] proposes active contour model and [14] uses Fourier deformable model with projective external energy. Although most of the studies assume heart is rigid during 3D reconstruction, [3, 13, 14, 17] include motion caused by respiration and heartbeat into their methods and obtain 4D model of coronary artery tree. In [8,10,11,13,15], X-ray images are synchronized using electrocardiography (ECG) recordings to ease 3D reconstruction process.

Calibration-free solution of [18] utilizes Random Sample Consensus (RANSAC) [21] method and eight-point algorithm to estimate the fundamental matrix for 3D reconstruction. In this way, the method internally achieves self-calibration in XRA image sequences. RANSAC method produces reliable correspondences during matching of centerline points. The work of [12] does not require calibration and uses a non-linear optimization algorithm to estimate projection matrix directly by making some assumptions on rotation of C-arm. The method tries to minimize the error between centerline points and back-projection of 3D reconstructed points.

It is clearly seen that classical 3D reconstruction methods depend on mathematical derivations between 2D and 3D geometry using some assumptions (pin-hole camera model). This process requires camera parameters to be known or estimated. For the ill-posed situations, where camera parameters are not known, feature correspondences are utilized to achieve self-calibration [18]. However, recent 3D reconstruction literature shows that objects can be reconstructed without calibration by the utilization of deep learning methods. Tremendous amount of progress has been drawn in last 5 years in learning-based 3D reconstruction. The problem remains challenging because there is not enough 3D data for training and there are many ways to represent 3D models such as point clouds, meshes, volumetric grids (voxels), etc. The proposed methods change according to the representations of 3D models. Moreover, although DL based methods are not utilized in the 3D reconstruction of arteries, in this part we focus on DL based object reconstruction methods. Thus, we divide 3D object reconstruction literature into 4 main categories based on their 3D representations as following: (1) point-cloud [22–26], (2) volumetric grids [27–34], (3) polygonal mesh [35, 36], (4) other specific data structures [25, 37, 38].

Point cloud representation is one of the most popular object representation techniques in computer graphics and computer vision. Data is simple to read, write and interpret in this format. However, the number of points should be limited if this format is used in deep learning architectures due to pre-determined output size. CAPNet is introduced in [22] and method uses an encoder-decoder architecture to obtain point cloud reconstructions from 2D image(s). Arbitrary views of reconstructed point clouds are projected onto 2D plane to refine point clouds with known camera parameters. Fan et al. [23] propose a network, PointOutNet, for generating 3D objects from a single image by obtaining its own conditional shape sampler. The output shape is 1024x3 in PointOutNet. Zamorski et al. [24] propose adversarial auto-encoders to learn 3D latent space of compact representation of 3D point clouds and generate 3D shape out of it. In [25], a deep learning method is proposed for reconstructing 3D shapes from 2D sketches in the form of line drawings. The network produces depth and normal map of input sketches and these outputs are transformed into a dense point cloud representing a 3D reconstruction of the input sketch. In [26], neural networks are trained to produce corresponding depth maps and silhouettes from input images. From these silhouettes and depth maps 3D point cloud is generated and refined.

Volumetric grid representation is generally preferred in the related literature because outputs of deep learning architectures are definite in terms of size. In this method, objects are placed into a 3D grid and grid size is constant. In [27], a recurrent network based solution is proposed for 3D object reconstruction from single or multi-view images. The architecture, called 3D-R2N2, takes a 127x127 input image and generates features from that image using auto-encoder network. The output of auto-encoder network is fed into gated recurrent unit (GRU) based recurrent network to hold necessary features in its internal memory. Finally, 3D reconstruction of arbitrary viewpoints of images is achieved in 3D occupancy grid (32x32x32 voxel grid) with a decoder network. Training and testing are achieved using ShapeNet [39], PASCAL VOC 2012 [40] and online 3D databases. In [32], an enhanced version of 3D-R2N2 is proposed by changing regular voxel values into weighted representation. In weighted representation, voxels are defined as integer in the range of (-53, +53) instead of 0 and 1. Neighbor voxels are taken into account with 3x3x3 kernel filter in their representation. It is claimed that training time is also improved in this way [32]. [28] and [29] propose deep learning architecture using Octree [135] structure for obtaining higher resolution of a 3D model in grid voxel representation. By using Octree, 3D models can be generated efficiently in terms of memory and calculation because the architecture does not have cubic complexity for 3D object (for instance:256*256*256 or 512*512*512). The neural network learns Octree structure in [29]. Wang et al. [31] design O-CNN, a novel octree data structure, to efficiently store the octant information occupied by 3D surfaces. The surface of objects are predicted at different scale of voxel blocks with octrees in [33] by using an encoder-decoder network. In order to incorporate the physics formations such as perspective geometry and occlusion, Markov Random Fields with ray-potentials are explicitly modeled with CNN in [34]. Yan et al. propose an encoder-decoder network for 3D object reconstruction (output size 32*32*32) without 3D supervision by including perspective projection in the loss function [30]. In this study, the intrinsic and extrinsic parameters are known for the camera.

Polygonal mesh representation is another format used mostly in computer graphics, games and animations. Wang et al. [35] generate meshes by deforming an ellipsoid into the desired shape from a single color image using convolutional neural network (CNN). Distance between camera and objects are assumed to be fixed. Pontes et al. [36] find the similar objects in the image from the 3D database using classification index and deform this object with the parameters that are produced using neural network. The method can reconstruct 3D shape of the object from single image. We include other 3D representations into problem specific part because these are mostly data-specific formats. For instance, [37] utilizes encoder-decoder architecture for constructing RGB+D of a given single image view of a car or a chair. Point cloud and mesh are generated from multi-view RGB+D output of the network. They train their network with ShapeNet [39]. Lun et al. [25] propose a deep learning method for automatically translating hand-drawn sketches into 2D images representing surface depth and normal across several output views. Encoder-decoder network is utilized for generation of 12 normal maps and corresponding depths. 3D point cloud and polygonal mesh of object are obtained by fusing these depth and normal predictions later. A relatively new study presented in [38], introduces 3D representation of an object with defining 64D vector. The vector makes a 3D object predictable from 2D images.

3. Material and Methods

Three-dimensional reconstruction itself is a challenging problem due to various requirements and constraints. The algorithms proposed in the literature follow similar analytical methods like finding correspondences between images, estimating camera parameters, using epipolar constraints, etc. as mentioned earlier in Section 2. There are many parameters in analytical solutions. On the other hand, deep learning architectures with appropriate data structures can eliminate the parameters that analytical methods use.

In this study, we propose a simple data structure for tubular shaped objects and validate this representation in reconstruction of coronaries. To the best of our knowledge, this is the first study that employs deep learning in the reconstruction of coronary arteries. Moreover, we present a method to generate a synthetic data-set and make it publicly available for researchers. This section presents details of synthetic data-set generation and proposed data structure along with some simple deep networks.

3.1. **Synthetic 3D Data Generation.**

Deep network architectures require plenty of labeled training data in order to learn input and output relationships. If the number of elements in the training set is lower, the network memorizes input-output relation, called overfitting. In this case, the network performance (the quality of output) degrades in tests. To generalize network for better test results, training should be accomplished using a relatively large data-set. Thus, we randomly generate 10000 3D models from real coronaries of nine subjects from [41]. We also generate centerlines and corresponding radius in order to use them in the proposed method.

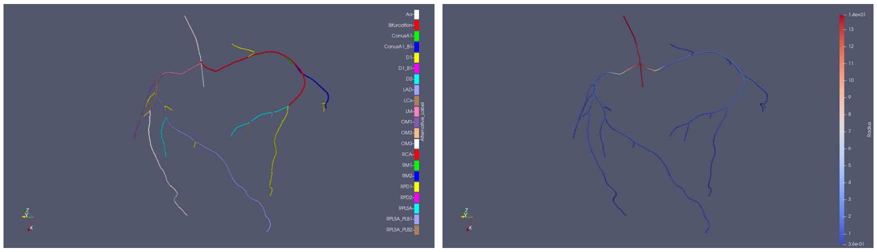
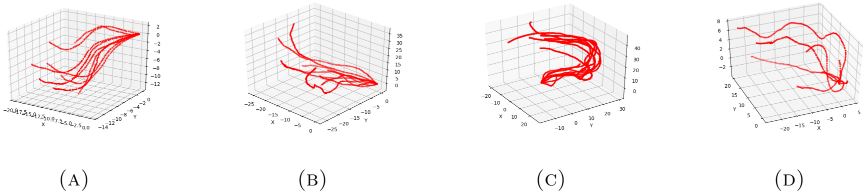


Figure 1. Coronary arteries of one of the subject in [41]. Branches are shown on the left and corresponding radii are shown on the right. The units are in mm for radii.

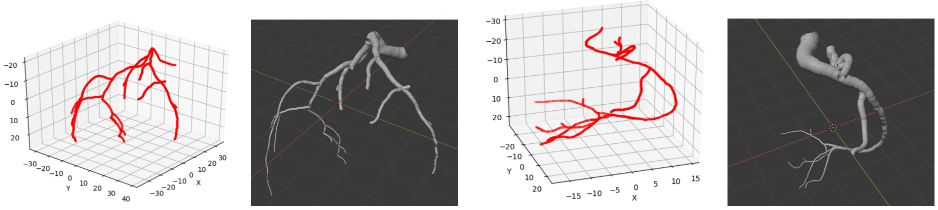
Left coronary artery (LCA) and right coronary artery (RCA) are branched off from the aorta to supply blood to the different parts of heart muscles. White colors on the left in Fig. 1 show aorta (Ao), red colors show main RCA branch and pink colors show the start of LCA. Typically, X-Ray angiogram is applied to one side of the arterial tree. Thus, we separate LCA and RCA branches during generation of synthetic data-set. In this way, networks can specifically learn anatomical properties of LCA and RCA structures. Another advantage of separating main branches is that the output size of the network becomes smaller. This will yield less computation and faster prediction time. Later, the combined version of the arterial tree can also be produced using the same logic presented here.



(A) (B) (C) (D)

Figure 2. Some of the branching patterns of coronary arterial tree. (A) and (B) branches of left main (LM) and left circumflex (LCx) arteries on LCA. (C) and (D) are RCA and the acute marginal branch (M1) on RCA.

We do not utilize the actual 3D data (centerline and radius) from [41]. We create random vessel trees by selecting one of the corresponding branches from 9 subjects. Additionally, some of the branches follow a similar path and we add them to the same group to obtain a random pool. We determine similarities by looking at visually multiple samples of branches. Vessels with similar patterns are gathered into the same group. In this way, we increase the variety of coronary trees that can be encountered in real cases. Example branching patterns for some of LCA and RCA branches from 9 subjects can be examined in Fig. 2.



(A) (B) (C) (D)

Figure 3. The coronary trees synthetically produced from our method. (A) shows 3D centerline points of (B) for LCA. (C) shows 3D centerline points of (D) for RCA.

We introduce more randomization to the vessel generation procedure to expand diversity of coronaries. There is a possibility to deviate 3D coordinates of centerline points and their corresponding radii. Some branches are also determined to be added in the coronary tree with an 80% possibility from the related random pool. We also add the possibility to change the distance between two consecutive centerline points. In this way, we can produce various coronary vessel forms that may resemble real coronary trees. It is a very low possibility to generate the same vessel tree in this configuration. Some example outputs from vessel generating software are illustrated in Fig. 3 with their 3D centerlines.

In order to obtain 3D model of coronaries, we perform a sampling operation around the center and corresponding radii. Since this process requires vector operations in 3D space, we provide implementation details of tubular shaped representation and its transformation into 3D objects in Appendix A. Once the 3D coronary is obtained, 2D images of the model should be produced from different views. Simulating C-arm machine behavior to generate images similar to X-Ray angiograms is described in detail in Appendix B.

3.2. Proposed Networks.

We propose 3 deep learning-based architectures for 3D reconstruction: (1) Multi-view fully CNN, (2) Time distributed auto-encoder CNN and (3) Auto-encoder followed by LSTM. The input to the networks is multi-view coronary vessel tree and the output is the sequential data structure we proposed for tubular shapes (see details in Appendix A).

3.2.1. *Multi-view CNN.*

We propose fully CNN network as shown in Fig. 4 for the construction of 3D synthetic coronaries. In this architecture, multi-view images are provided to the network as channels. Thus, we name it as Multi-view Fully CNN (MvFCNN). We add residual connections to carry features from previous layers to deeper layers. This approach generally prevents loss of information and performs better performance on the given task. The convolution filter size for each layer for

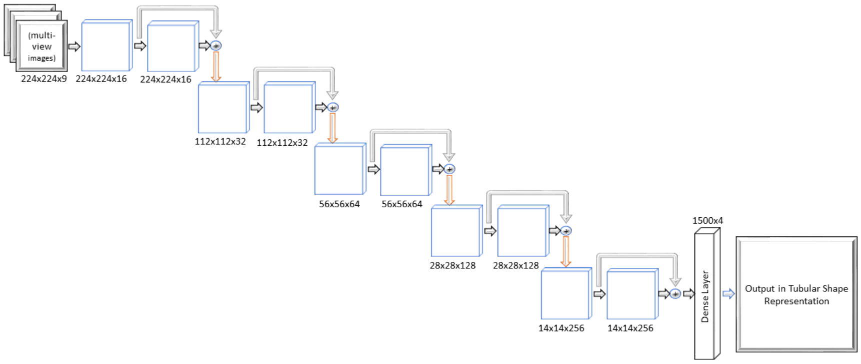


FIGURE 4. Multi-view fully CNN 3D reconstruction network. Activation function followed by each 3x3 convolution layer is Leaky-ReLU. Gray and orange arrows show residual connections and down-sampling with strided convolution (s=2), respectively.

MvFCNN architecture is 3x3. Downsampling is achieved using convolution operation with a stride of 2. In this way, the architecture continues to update weights while shrinking the size of the input (or embedding information from inputs) in intermediate layers. We utilize residual connections to transfer hierarchical representations to deeper layers of the network. In the end, we utilize fully connected layer to transform embedded information into the data structure we defined for the tubular shapes. Finally, we convert tubular shape representation into 3D objects using some math described in A.

3.2.2. *Time-distributed Auto-encoder CNN.* We use the same MvFCNN architecture of the previous section, but modify input and output layers to build a larger network as shown in Fig. 5. The differences are that (1) input images are separately supplied to the network instead of giving them as channels and (2) we remove the fully-connected layer at the end. In this way, each image is encoded into a feature space with the same auto-encoder. The features coming from different images are combined and used to produce our structured output with fully-connected layer. Fig. 5 visualizes this process clearly. In this figure, images of the same coronary tree are denoted as $X_1, X_2, \ldots, X_n$. These images are supplied to the same network to produce corresponding embedded features $E_1, E_2, \ldots, E_n$. Finally, embedded features are fed into a dense layer to produce our structured output.

For simplicity, we abbreviate time-distributed auto-encoder network as "TDAEn-Dense". In this naming convention, TD stands for time-distributed, AEn stands for auto-encoder and Dense is the final layer.

3.2.3. *Auto-encoder LSTM Decoder.* We use an auto-encoder network to embed features into a latent-space, and pass these features into memory cells of LSTM to hold necessary information that is coming from the current input. We remove the

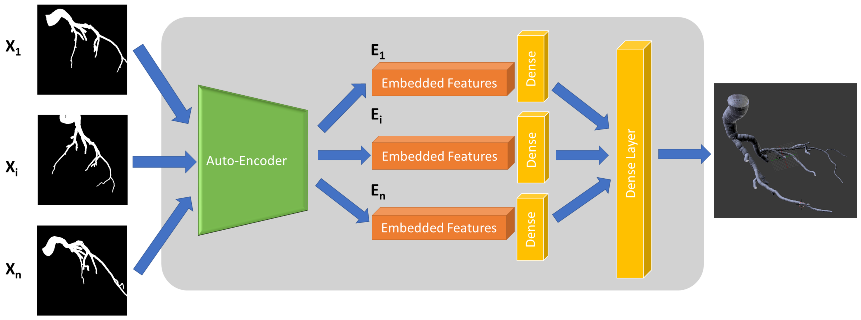


FIGURE 5. Time-distributed auto-encoder network. Input images are provided to the network in series.

fully connected part of the network in Fig. 5 and replace it with LSTM. In order to extract embedded information in LSTM cells, we place a decoder network at the end of LSTM. In this configuration, LSTM learns to hold important features in its internal memory and updates its hidden state according to incoming inputs. The network is abbreviated as En-LSTM-Dec in this study. The LSTM cell size is set to 1500.

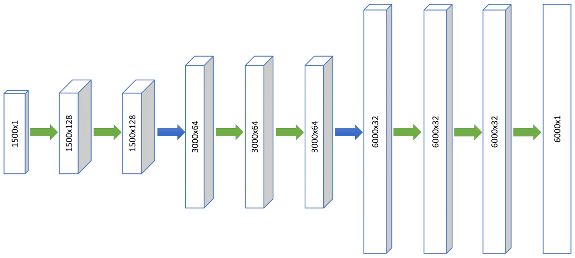


FIGURE 6. The decoder part for 3D reconstruction network.

The decoder part plays an important role in the architecture because the entire input sequence is now represented in LSTM's memory cells as a fixed-length vector. This encoded vector should be transformed into desired output sequences. The intuition here is that the decoder learns this complex representation and extracts the embedded information that LSTM holds. Thus, the output of LSTM is provided to the input of the decoder part in Fig. 6 to decode the embedded information within it.

3.3. **Performance Measures.** We evaluate the performance of 3D reconstruction predictions for 3D deep network architectures using Chamfer Distance (CD). It is utilized in ShapeNet's [39] shape reconstruction challenge. However, this measure works with point cloud representation. Therefore, we use center points in our structured data definition of 3D tubular objects for quantitative comparisons. CD is the evaluation criterion for estimating 3D points in our data representation. However; our networks also predict the radius of each point. To compare network performance quantitatively, we use mean squared error (MSE) to evaluate average error on the radius as follows:

$$E_R(r,\widetilde{r}) = \frac{1}{M}\frac{1}{N}\sum_{s}\sum_{i}(r_{si} - \widetilde{r_{si}})^2 \tag{1}$$

where M is the number of test samples, N is the number of points in sample s, r_{si} and $\widetilde{r_{si}}$ are the reference (ground-truth) and predicted radius values in test sample s for the point at i.

4. Experimental Results

In this section, we show the validation of our 3D reconstruction methods using synthetic data-set we produced from real subjects as mentioned earlier in Section 3.1. We share the results of several experiments conducted on networks and compare their reconstruction performances in terms of CD and MSE. We present network comparisons under three subsections. The first section compares predictions quantitatively using statistical measures. The second section compares them qualitatively and finally, the last section compares the average running time.

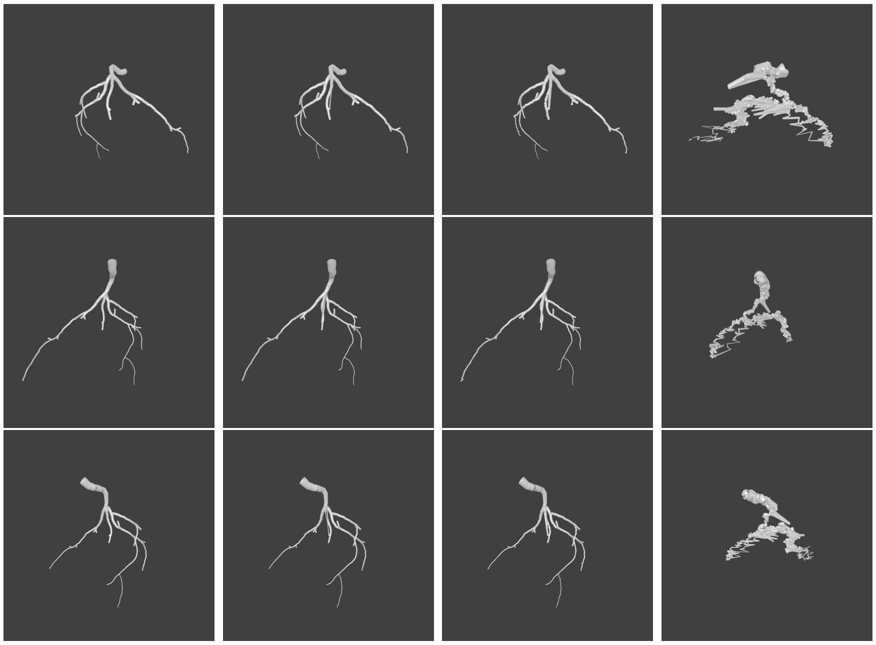
4.1. **Quantitative Comparison for 3D Reconstruction.** We quantify the performance of each network proposed for multi-view 3D reconstruction of coronary arteries and report the results in terms of CD and MSE. CD is used for the evaluation of center-line points in 3D and MSE is used for the evaluation of radius estimations. The lower values of both metrics indicate better 3D reconstruction that networks estimate.

There are 3 different 3D reconstruction networks described in Section 3.2. Table 1 shows the results of each network. In order to show network comparisons fairly and clearly, we present results with statistical measures; namely, mean (μ) and standard deviation (σ).

Table 1. Reconstruction performances of networks proposed in this study.

Model Name	**Chamfer Distance**		**Radius Error**	
	μ	σ	μ	σ
MvFCNN	0.00998328	0.00299526	0.00000580	0.00000226
TDAEn_Dense	0.03288964	0.02404985	0.00014654	0.00039439
En_LSTM_Dec	1.54436864	0.17509677	0.29157797	0.15928418

As shown in Table 1, MvFCNN outperforms others. The second architecture which achieves relatively good performance is TDAEn_Dense. These two architectures have no memory structure and data is provided explicitly as part of their input. However; in LSTM-based architecture, data is provided in series and it is expected to remember past observations. We observe that the architecture with internal memory (LSTM) have the largest errors in both metrics. The reasons why LSTM-based network performs poorly on 3D reconstruction might be that (1) the number of epoch or (2) memory units are not enough (it was 500 epoch and 1500 memory unit for LSTM in this study) for them to learn hidden representations or (3) they are not suited for auto-regression type problems as stated in [6]. Increasing the number of training epochs or using a stack of LSTMs may have improved the results but the network sizes are quite large as in the current configuration. Prediction time increases as the number of parameters are increased. Thus, we do not add more layers to memory-based network. We perform an additional test by increasing the number of units in LSTMs to 3000, but it does not improve the reconstruction performance for at least learning the vessel tree structure.



(A) Ground-Truth (B) MvFCNN (C) TDAEn_Dense (D) En_LSTM_Dec

Figure 7. The prediction of proposed networks for qualitative comparisons (test data o00015). Rows show the same object from different views.

4.2. **Qualitative Comparison for 3D Reconstruction.** We depict the reconstruction performances of each network in Fig. 7. It is clear from the figure that LSTM-based network produces worse predictions than CNN based networks. We observe that they fail to learn to estimate 3D positions of centers and corresponding radius of coronaries. The reasons why LSTM-based networks failed are discussed previously in Section 4.1.

If we compare predictions qualitatively, we observe that MvFCNN and TDAEn_Dense networks perform well on 3D reconstruction. There is no difference visually worth mentioning but quantitative comparison in Table 1 shows that MvFCNN produces, on average, 3.3 times better than TDAEn_Dense in CD metric and 25.27 times better in radius estimates. En_LSTM_Dec has the worst 3D reconstruction performance. It fails to construct the structure of coronary artery anatomy (see Fig. 7). We consider this is because LSTMs are not appropriate choice for auto-regression type problem.

4.3. **Execution Time.** The experiments for running time have been implemented on two Linux-based machines. The first one is a notebook computer with the hardware specifications: Intel(R) Core (TM) i7-4700HQ CPU@2.40GHz and 16GB 1600 MHz DDR3 RAM. We use this machine for testing CPU-based platforms. The second one is a server machine where training is achieved. The hardware specifications of this machine are as follows: Xeon(R) CPU E5-2667v4 (3.20GHz) CPU, 128GB of RAM and NVIDIA Tesla P100 Graphics Processing Unit (GPU). We use this machine for testing GPU-based platforms.

In order to compare network complexities, the total number of trainable parameters and the number of CNN layers used in each network are shared in Table 2. TDAEn_Dense appears to have the most parameters but this is because it uses copies of network to calculate prediction and combines all results in dense layer (see Fig. 5). Therefore, the number of parameters in dense layer increases. The latent size for LSTM based network is 1500, however, the internal structure of LSTM is more complex than just a single metric. It has 5 different gate calculations for updating its memory and output. The running time complexities validates this computational complexity as demonstrated in Table 3.

Table 2. Total number of parameters for each network.

Metric / Model Name	**MvFCNN**	**TDAEn_Dense**	**En_LSTM_Dec**
Num. of Convolutional Layers	14	20	28
Num. of Dense Layer	1	1	0
Num. of Trainable Parameters	304,201,808	626,246,480	311,528,065

We measure prediction time on test data for each network and calculate its average over 3000 samples. Table 3 reports the average running-time on CPU and GPU based platforms for a single prediction.

Since memory-based (LSTM) network are larger (more complex, more weights, more calculation), it produces prediction slower than the other two networks. If we compare run-time for different platforms, GPU based platform produces at least

Table 3. Run-time comparison of proposed networks on CPU and GPU based platforms.

Model Name	CPU (seconds)	GPU (seconds)	Ratio (CPU / GPU)
MvFCNN	0.033588	0.001923	17.47
TDAEn_Dense	0.120523	0.004286	28.12
En_LSTM_Dec	0.243891	0.005843	41.74

17.47 times faster than CPU based platform on average. This gain increases to 41.74 for larger and more complex deep networks. The fastest network appears to be MvFCNN with approximately 30 and 520 predictions per second on CPU and GPU, respectively.

5. Conclusion

In this paper, we have proposed deep-learning based 3D reconstruction method from 2D X-Ray angiograms for coronary arterial tree. Previously, this problem is solved using analytical methods in the related literature but such methods have some limitations discussed earlier in Section 1. We have solved the same problem with a learning-based approach. In this study, for the first time in literature, we demonstrate that coronary arteries can be reconstructed using 3D deep networks. We have proposed variations of 3D-R2N2 network [27] and multi-view CNN based networks. 3D-R2N2 network produces voxel-grids and the resolution is very low. It is not appropriate for our problem. Thus, we modify object representation and present well-defined data-structure for 3D representation of objects in tubular/cylindrical shapes described in Section A. We have shown that networks can learn this representation and show promising results. We prove the validation of the data-structure and architectures proposed in this study for 3D reconstruction of coronaries.

In this study, 3D reconstruction task is designed as a regression problem in which output contains continuous values. We conduct experimental tests to quantitatively evaluate and compare the performance of networks. We observe that CNN-based networks yield better 3D reconstruction performances in terms of CD and MSE metrics. The results for memory-based network (LSTM) are worse than CNN only-based networks and memory-based network fails to construct coronary tree anatomy. We consider this is because memory-based networks must remember past observations thanks to their internal cells whereas CNN-based networks are provided this data explicitly. We conclude that LSTMs are not appropriate for auto-regression problems as stated in [6].

We show and validate our 3D reconstruction method in this study using a synthetic data-set produced from real subjects. We believe that 3D reconstruction with deep learning architecture can be improved further with the evaluation of technologies and development of variants of network architectures. We believe that the topic (DL-based 3D reconstruction) becomes a trending problem shortly which is already started with object reconstruction.

5.1. **Limitations and Future Work.** We consider 3D reconstruction problem as a regression problem in this study. We train networks to estimate 6000 floating point numbers from their inputs. This is a difficult task. Instead, a deformable model can be used to facilitate the 3D reconstruction problem, as the anatomy of the heart vessels is specific. This kind of approach will decrease the number of parameters in networks and we believe that the performance will improve on 3D reconstruction.

Secondly, one may use transfer learning method to train (fine-tuning) our network on real coronary data. Since there is no publicly available X-Ray coronary data-set, we hope this study encourages researchers to share their X-Ray angiogram data with the community.

Vessel construction method presented in this study can also be applicable to other forms of vessels such as retinal images. Researchers are encouraged to try tubular shape representation in their work.

Author Contribution Statements The authors jointly worked on the study. All authors read and approved the final copy of the manuscript.

Declaration of Competing Interests The authors declare that there is no conflict of interest regarding the publication of this article.

Acknowledgement We would like to thank Professor Pablo Javier Blanco for sharing his research data (3D coronary centerlines) with us. This data enabled us to progress rapidly in deep-learning-based 3D reconstruction in our work.

This study is based on Ph.D. thesis of İ. Atli and is supported by Ankara Yildirim Beyazit University BAP, Turkey under the project number 3809.

References

[1] World health organization, `https://www.who.int/news-room/fact-sheets/detail/cardiovascular-diseases-(cvds)`, Accessed: 2020-07-22.

[2] Xiao, R., Yang, J., Fan, J., Ai, D., Wang, G., Wang, Y., Shape context and projection geometry constrained vasculature matching for 3d reconstruction of coronary artery, *Neurocomputing, 195* (2016), 65–73, https://dx.doi.org/10.1016/j.neucom.2015.08.110.

[3] Zheng, S., Meiying, T., Jian, S., Sequential reconstruction of vessel skeletons from x-ray coronary angiographic sequences, *Comput. Med. Imaging Graph., 34* (5) (2010), 333–345, https://dx.doi.org/10.1016/j.compmedimag.2009.12.004.

[4] Fazlali, H. R., Karimi, N., Soroushmehr, S. M. R., Sinha, S., Samavi, S., Nallamothu, B., Najarian, K., Vessel region detection in coronary x-ray angiograms, In *Proc. - Int. Conf. Image Process. ICIP (ICIP)* (2015), IEEE, pp. 1493–1497, https://dx.doi.org/0.1109/ICIP.2015.7351049.

[5] Medical radiation, https://www.medicalradiation.com/types-of-medical-imaging/imaging-using-x-rays/angiography/, Accessed: 2020-07-23.

[6] Gers, F. A., Eck, D., Schmidhuber, J., Applying lstm to time series predictable through time-window approaches, In *Neural Nets WIRN Vietri-01*, Springer, 2002, pp. 193–200, https://dx.doi.org/10.1007/978-1-4471-0219-9_20.

[7] Cong, W., Yang, J., Ai, D., Chen, Y., Liu, Y., Wang, Y., Quantitative analysis of deformable model-based 3-d reconstruction of coronary artery from multiple angiograms, *IEEE Trans. Biomed. Eng., 62* (8) (2015), 2079–2090, https://dx.doi.org/10.1109/TBME.2015.2408633.

[8] Yang, J., Cong, W., Chen, Y., Fan, J., Liu, Y., Wang, Y., External force back-projective composition and globally deformable optimization for 3-d coronary artery reconstruction, *Phys. Med. Biol., 59* (4) (2014), 975, https://dx.doi.org/10.1088/0031-9155/59/4/975.

[9] Chen, S. J., Carroll, J. D., 3-d reconstruction of coronary arterial tree to optimize angiographic visualization, *IEEE Trans. Med. Imag., 19* (4) (2000), 318–336, https://dx.doi.org/10.1109/42.848183.

[10] Chen, S.-Y., Carroll, J. D., Kinematic and deformation analysis of 4-d coronary arterial trees reconstructed from cine angiograms, *IEEE Trans. Med. Imag., 22* (6) (2003), 710–721, https://dx.doi.org/10.1109/TMI.2003.814788.

[11] Andriotis, A., Zifan, A., Gavaises, M., Liatsis, P., Pantos, I., Theodorakakos, A., Efstathopoulos, E. P., Katritsis, D., A new method of three-dimensional coronary artery reconstruction from x-ray angiography: Validation against a virtual phantom and multislice computed tomography, *Catheter. Cardiovasc. Interv., 71* (1) (2008), 28–43, https://dx.doi.org/10.1002/ccd.21414.

[12] Yang, J., Wang, Y., Liu, Y., Tang, S., Chen, W., Novel approach for 3-d reconstruction of coronary arteries from two uncalibrated angiographic images, *IEEE Trans. Image Process., 18* (7) (2009), 1563–1572, https://dx.doi.org/10.1109/TIP.2009.2017363.

[13] Liu, X., Hou, F., Hao, A., Qin, H., A parallelized 4d reconstruction algorithm for vascular structures and motions based on energy optimization, *Vis. Comput., 31* (11) (2015), 1431–1446, https://dx.doi.org/10.1007/s00371-014-1024-4.

[14] Sarry, L., Boire, J.-Y., Three-dimensional tracking of coronary arteries from biplane angiographic sequences using parametrically deformable models, *IEEE Trans. Med. Imag., 20* (12) (2001), 1341–1351, https://dx.doi.org/10.1109/42.974929.

[15] Cañero, C., Vilariño, F., Mauri, J., Radeva, P., Predictive (un) distortion model and 3-d reconstruction by biplane snakes, *IEEE Trans. Med. Imag., 21* (9) (2002), 1188–1201, https://dx.doi.org/10.1109/TMI.2002.804421.

[16] Hoffmann, K. R., Sen, A., Lan, L., Chua, K.-G., Esthappan, J., Mazzucco, M., A system for determination of 3d vessel tree centerlines from biplane images, *Int. J. Card. Imag., 16* (5) (2000), 315–330, https://dx.doi.org/10.1023/A:1026528209003.

[17] Shechter, G., Devernay, F., Coste-Manière, E., Quyyumi, A., McVeigh, E. R., Three-dimensional motion tracking of coronary arteries in biplane cineangiograms, *IEEE Trans. Med. Imag., 22* (4) (2003), 493–503, https://dx.doi.org/10.1109/TMI.2003.809090.

[18] Fallavollita, P., Cheriet, F., Optimal 3d reconstruction of coronary arteries for 3d clinical assessment, *Comput. Med. Imaging Graph., 32* (6) (2008), 476–487, https://dx.doi.org/10.1016/j.compmedimag.2008.05.001.

[19] Wiesent, K., Barth, K., Navab, N., Durlak, P., Brunner, T., Schuetz, O., Seissler, W., Enhanced 3-d-reconstruction algorithm for c-arm systems suitable for interventional procedures, *IEEE Trans. Med. Imag., 19* (5) (2000), 391–403, https://dx.doi.org/10.1109/42.870250.

[20] Liao, R., Luc, D., Sun, Y., Kirchberg, K., 3-d reconstruction of the coronary artery tree from multiple views of a rotational x-ray angiography, *Int. J. Card. Imag., 26* (7) (2010), 733–749, https://dx.doi.org/10.1007/s10554-009-9528-0.

[21] Torr, P. H., Murray, D. W., The development and comparison of robust methods for estimating the fundamental matrix, *Int. J. Comput. Vis., 24* (3) (1997), 271–300, https://dx.doi.org/10.1023/A:1007927408552.

[22] Navaneet, K., Mandikal, P., Agarwal, M., Babu, R. V., Capnet: Continuous approximation projection for 3d point cloud reconstruction using 2d supervision, In *Proceedings of the AAAI Conference on Artificial Intelligence* (2019), vol. 33, pp. 8819–8826, https://dx.doi.org/10.1609/aaai.v33i01.33018819.

[23] Fan, H., Su, H., Guibas, L. J., A point set generation network for 3d object reconstruction from a single image, In *CVPR* (2017), pp. 605–613.

[24] Zamorski, M., Zieba, M., Klukowski, P., Nowak, R., Kurach, K., Stokowiec, W., Trzciński, T., Adversarial autoencoders for compact representations of 3d point clouds, *Comput. Vis. Image Underst., 193* (2020), 102921, https://dx.doi.org/10.1016/j.cviu.2020.102921.

[25] Lun, Z., Gadelha, M., Kalogerakis, E., Maji, S., Wang, R., 3d shape reconstruction from sketches via multi-view convolutional networks, In *Proc. - 2017 Int. Conf. 3D Vis. (3DV)* (2017), IEEE, pp. 67–77, https://dx.doi.org/10.1109/3DV.2017.00018.

[26] Arsalan Soltani, A., Huang, H., Wu, J., Kulkarni, T. D., Tenenbaum, J. B., Synthesizing 3d shapes via modeling multi-view depth maps and silhouettes with deep generative networks, In *CVPR* (2017), pp. 1511–1519.

[27] Choy, C. B., Xu, D., Gwak, J., Chen, K., Savarese, S., 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction, In *ECCV* (2016), Springer, pp. 628–644, https://dx.doi.org/10.1007/978-3-319-46484-8_38.

[28] Riegler, G., Osman Ulusoy, A., Geiger, A., Octnet: Learning deep 3d representations at high resolutions, In *CVPR* (2017), pp. 3577–3586.

[29] Tatarchenko, M., Dosovitskiy, A., Brox, T., Octree generating networks: Efficient convolutional architectures for high-resolution 3d outputs, In *Proc. IEEE Int. Conf. Comput. Vis.* (2017), pp. 2088–2096.

[30] Yan, X., Yang, J., Yumer, E., Guo, Y., Lee, H., Perspective transformer nets: Learning single-view 3d object reconstruction without 3d supervision, In *Adv. Neural Inf. Process. Syst.* (2016), pp. 1696–1704.

[31] Wang, P.-S., Liu, Y., Guo, Y.-X., Sun, C.-Y., Tong, X., O-cnn: Octree-based convolutional neural networks for 3d shape analysis, *ACM Trans. Graph., 36* (4) (2017), 1–11, https://dx.doi.org/10.1145/3072959.3073608.

[32] Xie, H., Yao, H., Sun, X., Zhou, S., Tong, X., Weighted voxel: a novel voxel representation for 3d reconstruction, In *Proceedings of the 10th International Conference on Internet Multimedia Computing and Service* (2018), pp. 1–4, https://dx.doi.org/10.1145/3240876.3240888.

[33] Häne, C., Tulsiani, S., Malik, J., Hierarchical surface prediction for 3d object reconstruction, In *Proc. - 2017 Int. Conf. 3D Vis. (3DV)* (2017), IEEE, pp. 412–420, https://dx.doi.org/10.1109/3DV.2017.00054.

[34] Paschalidou, D., Ulusoy, O., Schmitt, C., Van Gool, L., Geiger, A., Raynet: Learning volumetric 3d reconstruction with ray potentials, In *CVPR* (2018), pp. 3897–3906.

[35] Wang, N., Zhang, Y., Li, Z., Fu, Y., Liu, W., Jiang, Y.-G., Pixel2mesh: Generating 3d mesh models from single rgb images, In *ECCV* (2018), pp. 52–67.

[36] Pontes, J. K., Kong, C., Sridharan, S., Lucey, S., Eriksson, A., Fookes, C., Image2mesh: A learning framework for single image 3d reconstruction, In *Asian Conference on Computer Vision* (2018), Springer, pp. 365–381, https://dx.doi.org/10.1007/978-3-030-20887-5_23.

[37] Tatarchenko, M., Dosovitskiy, A., Brox, T., Multi-view 3d models from single images with a convolutional network, In *ECCV* (2016), Springer, pp. 322–337, https://dx.doi.org/10.1007/978-3-319-46478-7_20.

[38] Girdhar, R., Fouhey, D. F., Rodriguez, M., Gupta, A., Learning a predictable and generative vector representation for objects, In *ECCV* (2016), Springer, pp. 484–499, https://dx.doi.org/10.1007/978-3-319-46466-4_29.

[39] Chang, A. X., Funkhouser, T., Guibas, L., Hanrahan, P., Huang, Q., Li, Z., Savarese, S., Savva, M., Song, S., Su, H., et al., Shapenet: An information-rich 3d model repository, *arXiv preprint arXiv:1512.03012* (2015).

[40] Everingham, M., Van Gool, L., Williams, C., Winn, J., Zisserman, A., The pascal visual object classes challenge 2012 (voc2012), *Results* (2012).

[41] Blanco, P. J., Bulant, C. A., Müller, L. O., Talou, G. M., Bezerra, C. G., Lemos, P., Feijóo, R. A., Comparison of 1d and 3d models for the estimation of fractional flow reserve, *Sci. Rep., 8* (1) (2018), 1–12, https://dx.doi.org/10.1038/s41598-018-35344-0.

[42] Horn, F., Leghissa, M., Kaeppler, S., Pelzer, G., Rieger, J., Seifert, M., Wandner, J., Weber, T., Michel, T., Riess, C., et al., Implementation of a talbot-lau interferometer in a clinical-like c-arm setup: A feasibility study, *Sci. Rep., 8* (1) (2018), 1–11, https://dx.doi.org/10.1038/s41598-018-19482-z.

Appendix

Appendix A. Transformation of Tubular Shapes to 3D

There are various ways to represent objects in 3D and 3D representation matters in terms of 3D machine learning. A compatible data structure with deep learning method can improve reconstruction performance of networks. Thus, we define a new data structure for holding tubular shapes specifically. In this structure, the center of circles and their radii are used to represent tubular shapes. The center points must come sequentially to obtain the path for a continuous tubular shape (see Fig. 8). In this way, we can find the direction of each consecutive circle using[1]:

$$\vec{n} = \frac{\overrightarrow{C_i C_{i+1}}}{\left\|\overrightarrow{C_i C_{i+1}}\right\|} = \frac{C_{i+1} - C_i}{\|C_{i+1} - C_i\|} \tag{2}$$

The segment shown in Fig. 8 starts with its center point (C_0) in 3D space and ends with the corresponding radius (r_0) of the circle centered at C_0 for the first line. We assume that this segment is continuous until the end of the segment sign (?, ?, ?, -1) appears. We represent the end of the segment with ?, ?, ? for indicating any value for center point and a negative value for radius (it is -1 in the example). The direction of tubular segments can easily be derived (using Eq. 2) from the continuity assumption in which center points must come in order (see Fig. 8).

In order to recover tubular shape, we need to sample points around centers with a given radius as in Fig. 8. $\vec{u}$ and $\vec{v}$ vectors are on the surface of the circle and can be derived with a little effort using equations 3, 4, 5 and 6. The variables n_x and n_y in the equations show unit normal values in X and Y directions, respectively. For deeper understanding of vectors in 3D space along with 3D reconstruction, please have a look at this book[2].

$$\vec{n} = \langle x, y, z \rangle \tag{3}$$

[1] Olver, Peter J., Chehrzad Shakiban, and Chehrzad Shakiban. Applied linear algebra. Prentice Hall, 2006

[2] Andrew, Alex M. "Multiple view geometry in computer vision." Kybernetes (2001)

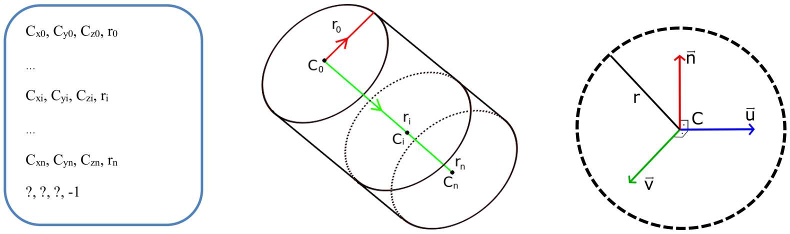

$C_{x0}, C_{y0}, C_{z0}, r_0$
...
$C_{xi}, C_{yi}, C_{zi}, r_i$
...
$C_{xn}, C_{yn}, C_{zn}, r_n$
?, ?, ?, -1
r_0
C_0
r_i
C_i
r_n
C_n
$\vec{n}$
r
C
$\vec{u}$
$\vec{v}$

FIGURE 8. Tubular shape representation for one segment. C_i and r_i show center and radius for the point at location i. For sampling tubular shapes around centers (on the right), $\vec{n}$ shows unit normal to the surface, $\vec{u}$ and $\vec{v}$ are perpendicular unit vectors lie on the surface of the circle.

$$\vec{t} = \langle n_x, -n_y, 0 \rangle \tag{4}$$

$$\vec{u} = \vec{n} \times \vec{t} \tag{5}$$

$$\vec{v} = \vec{n} \times \vec{u} \tag{6}$$

We parametrize the circle using these vectors ($\vec{u}$ and $\vec{v}$) around each center point C_i to obtain represented 3D cylindrical shape. In our tests, we sampled each circle with 20 points. If desired, this can be increased for a better resolution of tubular shapes. Parametrization over the angle, θ, can be achieved as follows:

$$P_{s_i} = (r * cos\theta)\vec{u} + (r * sin\theta)\vec{v} + C_i \tag{7}$$

where P_{s_i} is an array of sampled points for $0 \leq \theta \leq 2\pi$ with a step of $2\pi/20$ and C_i is i^{th} center point in the segment.

APPENDIX B. MULTI-VIEW IMAGE GENERATION

Remembering the main goal posed at the beginning of this study, we propose deep-learning based 3D reconstruction method for coronary arteries from 2D X-Ray angiograms to provide an objective, reproducible clinical assistance. To achieve this goal, we have generated 3D coronaries synthetically with the procedure mentioned in Section 3.1. The next step is to produce multi-view images of coronaries. We have created a tool for producing multiple images from different views of angle like in X-Ray C-arm machine. We simulate 2 axis rotations demonstrated in Fig. 9. The coordinates of the camera (head point of C-arm) after rotations are calculated using spherical coordinates as follows:

$$\langle C_x, C_y, C_z \rangle = \langle \rho \sin(\theta), \rho \cos(\theta), \rho \sin(\theta) \cos(\psi) \rangle \tag{8}$$

where ρ is the distance between the camera and the geometric center of objects (coronaries). We use Blender software[3] to produce multiple-view images (n=3) of coronary arterial tree. Since operators capture X-Ray images in variety of angles to visualize blood vessels clearly, we use random rotations during the generation of multi-view images.

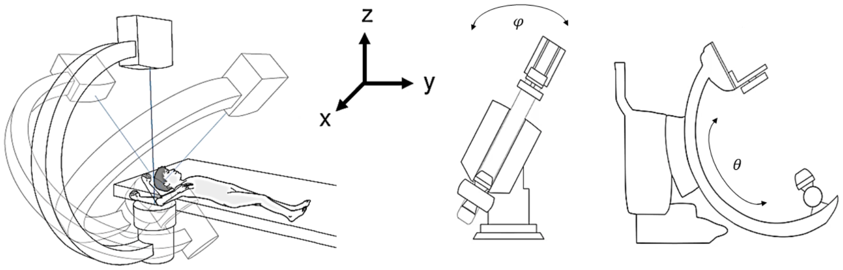


FIGURE 9. C-arm rotation axis simulation. θ and ψ represent rotations in y-axis and x-axis, respectively [42].

[3] www.blender.org