



Weakly Supervised Deep Learning for Aortic Valve Finite Element Mesh Generation from 3D CT Images

Daniel H. Pak¹(✉), Minliang Liu², Shawn S. Ahn¹, Andrés Caballero²,
John A. Onofrey³, Liang Liang⁴, Wei Sun², and James S. Duncan^{1,3}

¹ Biomedical Engineering, Yale University, New Haven, CT, USA
daniel.pak@yale.edu

² Biomedical Engineering, Georgia Institute of Technology, Atlanta, GA, USA

³ Radiology and Biomedical Imaging, Yale School of Medicine, New Haven, CT, USA

⁴ Computer Science, University of Miami, Miami-Dade County, FL, USA

Abstract. Finite Element Analysis (FEA) is useful for simulating Transcatheter Aortic Valve Replacement (TAVR), but has a significant bottleneck at input mesh generation. Existing automated methods for imaging-based valve modeling often make heavy assumptions about imaging characteristics and/or output mesh topology, limiting their adaptability. In this work, we propose a deep learning-based deformation strategy for producing aortic valve FE meshes from noisy 3D CT scans of TAVR patients. In particular, we propose a novel image analysis problem formulation that allows for training of mesh prediction models using segmentation labels (i.e. weak supervision), and identify a unique set of losses that improve model performance within this framework. Our method can handle images with large amounts of calcification and low contrast, and is compatible with predicting both surface and volumetric meshes. The predicted meshes have good surface and correspondence accuracy, and produce reasonable FEA results.

Keywords: Weakly supervised deep learning · Shape deformation · Aortic valve modeling

1 Introduction

Transcatheter Aortic Valve Replacement (TAVR) is an emerging minimally invasive treatment option for aortic stenosis [22]. In recent years, studies have explored Finite Element Analysis (FEA) for simulating TAVR from pre-operative patient images, and have shown promising results for predicting patient outcome and finding better treatment strategies [3]. However, there exists a significant bottleneck at producing FE meshes from patient images, as the manual process takes several hours for each patient and requires expert knowledge about the anatomy as well as the meshing techniques and requirements.

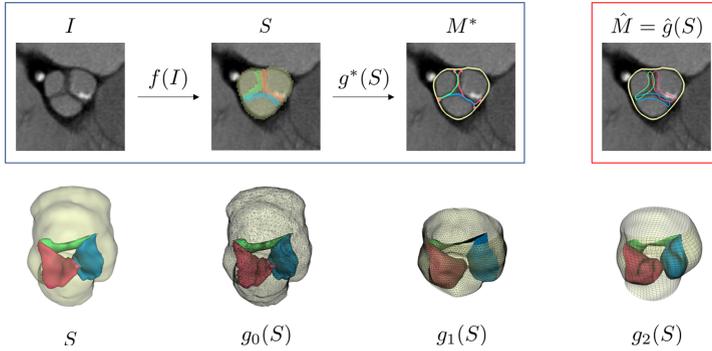


Fig. 1. Top row: $I \rightarrow S \rightarrow M$ progression, where blue box shows the desired sequence and red box shows a faulty output using marching cubes, a common choice for $\hat{g}(S)$. Bottom row: different meshes using the same segmentation based on the choice of $g(S)$. Y: aortic root/wall and R, G, B: valve leaflets. (Color figure online)

Automated solutions for aortic valve modeling have been proposed. Ionasec et al. [7] and Liang et al. [11] used a combination of landmark and boundary detection driven by intensity-based features to produce valve meshes with a predefined topology. Pouch et al. [16] used multi-atlas segmentation and medial modeling to match the template mesh to the predicted segmentation. Ghesu et al. [5] used marginal space deep learning to localize an aortic wall template mesh and subsequently deformed it along the surface normal. Although various approaches have demonstrated success with valve modeling, they have drawbacks such as heavy reliance on intensity changes along the valve structures, extensive assumptions about valve geometry, and limited adaptability to FE meshes due to assumptions about the output mesh topology.

To address these issues, we propose a deep learning-based deformation strategy for predicting FE meshes from noisy 3D CT scans of TAVR patients. Our main contributions are three-fold: (1) We propose a novel image analysis problem formulation that allows for weakly supervised training of image-to-mesh prediction models, where training is performed with segmentation labels instead of mesh labels. (2) We make minimal assumptions in defining this formulation, so it can easily adapt to various imaging conditions and desired output mesh topology (even from surface mesh to volumetric mesh). (3) We identify a unique set of losses that improves model performance within this framework.

2 Methods

2.1 Possible Problem Formulation: Meshing from Segmentation

Let I be an image, and let S and M be the corresponding segmentation and mesh outputs, respectively. Considering the sequential generation steps $I \rightarrow S \rightarrow M$, we can define two mapping functions $f(I) = S$ and $g(S) = M$ (Fig. 1).

The most common choices for $f(I)$ and $g(S)$ are the inside volume a structure and marching cubes, respectively. However, for thin structures such as valve leaflets, naïve surface meshing fails to provide the desired open surface meshes with accurate attachment points (Fig. 1). It is also problematic for the aortic wall, which requires tube-like openings. Therefore, to obtain the correct surface, we must extract structural information from S via curve fitting [11], medial modeling [16], or manual labeling. This makes it extremely difficult to define a general $g(S)$ without making heavy assumptions about the anatomy and output mesh even when provided with manually defined S during test time.

2.2 Proposed Problem Formulation: Mesh Template Matching

Instead of solving for $g(S)$, we propose a problem formulation that we refer to as Mesh Template Matching (MTM), summarized schematically in Fig. 2. Here, we find the deformation field ϕ^* :

$$\phi^* = \arg \min_{\phi} \mathcal{L}(M, M_0, \phi) \quad (1)$$

where M and M_0 are target and template meshes, respectively. We use a convolutional neural network (CNN) as our function approximator $h_{\theta}(I; S_0, M_0) = \phi$ where I is the image, S_0 is the segmentation template (paired to M_0), and θ is the network parameters. Then, we solve for θ that minimizes the loss:

$$\theta^* = \arg \min_{\theta} [\mathbb{E}_{(I, M) \sim D} [\mathcal{L}(M, M_0, h_{\theta}(I; S_0, M_0))]] \quad (2)$$

where D is the training set distribution. We propose two different variations of MTM, as detailed below.

2.2.1 MTM

For the vanilla MTM, we initially defined \mathcal{L} from Eq. 1 as:

$$\mathcal{L}(M, M_0, \phi) = \mathcal{L}_{acc}(M, \phi(M_0)) + \lambda \mathcal{L}_{smooth}(\phi) \quad (3)$$

where $\phi(M_0)$ is the deformed template, \mathcal{L}_{acc} is the spatial accuracy loss, and \mathcal{L}_{smooth} is the field smoothness loss with a scaling hyperparameter λ . From here, we removed the need for ground truth M with the following steps:

$$\arg \min_{\phi} \mathcal{L}(M, M_0, \phi) = \arg \min_{\phi} [\mathcal{L}_{acc}(M, \phi(M_0)) + \lambda \mathcal{L}_{smooth}] \quad (4)$$

$$= \arg \min_{\phi} [\mathcal{L}_{acc}(g^*(S), \phi(g^*(S_0))) + \lambda \mathcal{L}_{smooth}] \quad (5)$$

$$\approx \arg \min_{\phi} [\mathcal{L}_{acc}(\hat{g}(S), \phi(\hat{g}(S_0))) + \lambda \mathcal{L}_{smooth}] \quad (6)$$

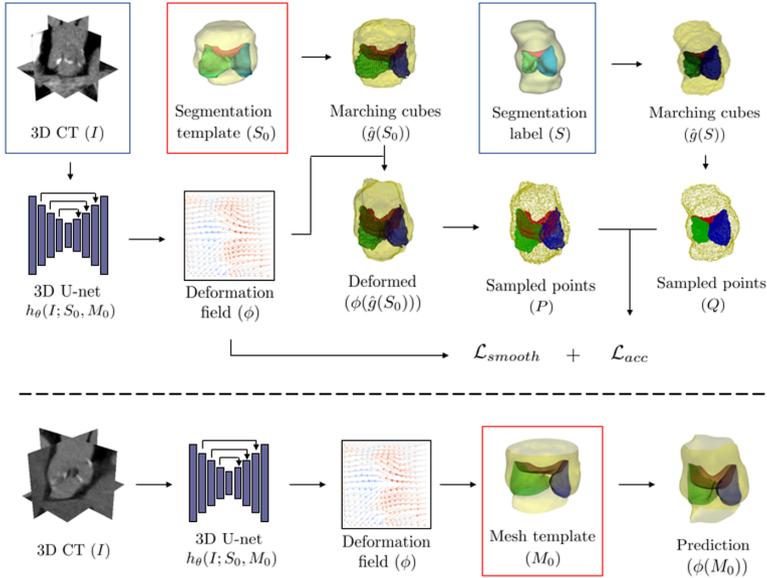


Fig. 2. Training (above dotted line) and inference steps (below dotted line) for the vanilla MTM. Blue box represents paired image-segmentation training samples and red box represents fixed templates. Note that M_0 can freely change topology as long as its surface is in close proximity to $\hat{g}(S_0)$ surface. MTMgeo is similar, but instead of \mathcal{L}_{smooth} , we calculate \mathcal{L}_{geo} using $\phi(M_0)$ during training. (Color figure online)

where g^* is the ideal meshing function (for which the topology is defined by the template mesh), \hat{g} is marching cubes, and S and S_0 are target and template segmentation volumes. Our key approximation step (Eq. 6) makes two important assumptions: (1) g^* mesh surface is in close proximity to \hat{g} mesh surface by Euclidean distance and (2) ϕ is smooth with respect to Euclidean space. The first assumption is reasonable because ground truth meshes are often created using segmentation labels as intermediate steps, and the second assumption is enforced by \mathcal{L}_{smooth} and the choice of ϕ (discussed further in Sect. 2.3).

Common choices for the spatial accuracy loss are mean surface distance (e.g. Chamfer distance) and volume overlap (e.g. Dice). Since Dice and other segmentation losses are typically lenient towards errors in segmentation boundaries (and we need accuracy at boundaries for meshes), we used the Chamfer distance:

$$\mathcal{L}_{acc}(P, Q) = \frac{1}{|P|} \sum_{\mathbf{p} \in P} \min_{\mathbf{q} \in Q} \|\mathbf{p} - \mathbf{q}\|_2^2 + \frac{1}{|Q|} \sum_{\mathbf{q} \in Q} \min_{\mathbf{p} \in P} \|\mathbf{q} - \mathbf{p}\|_2^2 \quad (7)$$

where P and Q are points sampled on surfaces of $\phi(\hat{g}(S_0))$ and $\hat{g}(S)$, respectively. We experimented with adding Dice [13] to the loss, but observed no significant improvement in performance.

For field smoothness loss, we used the bending energy term to penalize non-affine fields [19]:

$$\mathcal{L}_{smooth} = \frac{1}{V} \int_0^X \int_0^Y \int_0^Z \left[\left(\frac{\partial^2 \phi}{\partial x^2} \right)^2 + \left(\frac{\partial^2 \phi}{\partial y^2} \right)^2 + \left(\frac{\partial^2 \phi}{\partial z^2} \right)^2 + 2 \left(\frac{\partial^2 \phi}{\partial xy} \right)^2 + 2 \left(\frac{\partial^2 \phi}{\partial xz} \right)^2 + 2 \left(\frac{\partial^2 \phi}{\partial yz} \right)^2 \right] dx dy dz \quad (8)$$

where V is the total number of voxels and X, Y, Z are the number of voxels in each dimension. We experimented with adding gradient magnitude [1] and field magnitude [10] to the loss, but observed no significant improvement in performance.

2.2.2 MTMgeo

For the second variation of MTM, referred to as MTMgeo, we replaced \mathcal{L}_{smooth} with \mathcal{L}_{geo} to preserve various desired geometric qualities of the deformed template mesh:

$$\mathcal{L}(M, M_0, \phi) = \mathcal{L}_{acc}(M, \phi(M_0)) + \mathcal{L}_{geo}(M_0, \phi(M_0)) \quad (9)$$

where we apply identical steps as Eq. 4–7 to get:

$$\arg \min_{\phi} \mathcal{L}(M, M_0, \phi) \approx \arg \min_{\phi} [\mathcal{L}_{acc}(P, Q) + \mathcal{L}_{geo}] \quad (10)$$

The geometric quality loss is a weighted sum of three different losses:

$$\mathcal{L}_{geo}(M_0, \phi(M_0)) = \lambda_0 \mathcal{L}_{norm} + \lambda_1 \mathcal{L}_{lap} + \lambda_2 \mathcal{L}_{edge} \quad (11)$$

where λ_i are scaling hyperparameters, \mathcal{L}_{norm} is face normal consistency loss, \mathcal{L}_{lap} is Laplacian smoothing loss, and \mathcal{L}_{edge} is edge correspondence loss.

$$\mathcal{L}_{norm} = \frac{1}{|\mathcal{N}_f|} \sum_{(\mathbf{n}_{fi}, \mathbf{n}_{fj}) \in \mathcal{N}_f} 1 - \frac{\langle \mathbf{n}_{fi}, \mathbf{n}_{fj} \rangle}{\|\mathbf{n}_{fi}\|_2 \|\mathbf{n}_{fj}\|_2} \quad (12)$$

\mathbf{n}_{fi} is the normal vector calculated at a given face and \mathcal{N}_f is the set of all pairs of neighboring faces' normals within $\phi(M_0)$.

$$\mathcal{L}_{lap} = \frac{1}{|V|} \sum_{\mathbf{v}_i \in \phi(V)} \left\| \frac{1}{|\mathcal{N}(\mathbf{v}_i)|} \sum_{\mathbf{v}_j \in \mathcal{N}(\mathbf{v}_i)} \mathbf{v}_i - \mathbf{v}_j \right\|_2 \quad (13)$$

$\mathcal{N}(\mathbf{v}_i)$ represents neighbors of \mathbf{v}_i . The norm represents the magnitude of the differential coordinates, which approximates the local mean curvature.

$$\mathcal{L}_{edge} = \frac{1}{|\varepsilon|} \sum_{(\mathbf{v}_i, \mathbf{v}_j) \in \varepsilon} \left(\frac{\|\mathbf{v}_i - \mathbf{v}_j\|_2}{\max_{(\mathbf{v}'_i, \mathbf{v}'_j) \in \varepsilon} \|\mathbf{v}'_i - \mathbf{v}'_j\|_2} - \frac{\|\phi(\mathbf{v}_i) - \phi(\mathbf{v}_j)\|_2}{\max_{(\mathbf{v}'_i, \mathbf{v}'_j) \in \varepsilon} \|\phi(\mathbf{v}'_i) - \phi(\mathbf{v}'_j)\|_2} \right)^2 \quad (14)$$

Our proposed edge correspondence loss is different from the edge length loss [6, 23] in that it allows meshes to change sizes more freely, as long as the edge length ratio stays consistent before and after deformation. This is beneficial for medical FE meshing tasks where (1) patients have organs of different sizes and (2) consistent edge ratio helps convert between triangular and quadrilateral faces. The latter is important because quadrilateral faces are desired in FEA for more accurate simulation results, but many mesh-related algorithms and libraries are only compatible with triangular faces.

2.3 Deformation Field

As mentioned in Sect. 2.2.1, the choice of ϕ is important for applying the key approximation step in MTM. Our approach is to learn a diffeomorphic B-spline deformation field from which we interpolate displacement vectors at each node of the template mesh. By requiring the result to adhere to topology-preserving diffeomorphism, we help prevent mesh intersections that can commonly occur with node-specific displacements. Also, when the field is calculated in the reverse direction for deforming a template image/segmentation to prevent hole artifacts, the invertible property of diffeomorphism allows for much more accurate field calculation. The B-spline aspect helps generate smooth fields, which prevents erratic changes in field direction or magnitude for nearby interpolation points.

3 Experiments and Results

3.1 Data Acquisition and Preprocessing

We used an aortic valve dataset consisting of 88 CT scans from 74 different patients, all with tricuspid valves. Of the 88 total scans, 73 were collected from transcatheter aortic valve replacement (TAVR) patients at the Hartford hospital, and the remaining 15 were from the training set of the MM-WHS public dataset [25]. From the Hartford images, we randomly chose some patients to include more than one random time point from the ~ 10 time points collected over the cardiac cycle. The 88 images were randomly split into 40, 10, 38 for training, validation, and testing sets, respectively, with no patient overlap between the training and testing sets. We pre-processed all CT scans by converting to Hounsfield units (HU), thresholding, and renormalizing to $[0, 1]$. We resampled all images to fix the spatial resolution at $1 \times 1 \times 1 \text{ mm}^3$, and cropped and rigidly aligned them using three manually annotated landmark points to focus on the aortic valve, resulting in final images with $[64, 64, 64]$ voxels.

We focused on 4 aortic valve components: the aortic root/wall (for segmentation/mesh, respectively) and the 3 leaflets. The ground truth segmentation labels (for training) and mesh labels (for testing) for all 4 components were obtained via manual and semi-automated annotations by human experts in 3D slicer [4]. We first obtained the segmentation labels using the brush tool and the robust statistics segmenter. Then, using the segmentation as guidance, we manually defined

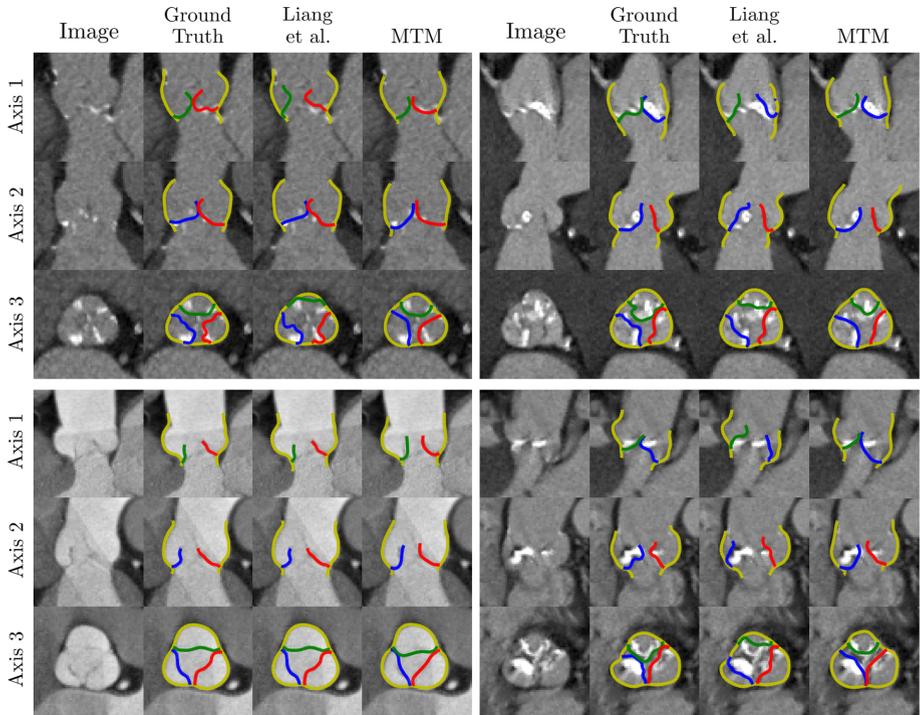


Fig. 3. Cross sectional display of CT images and meshes at 3 different viewing planes. Each block of images (separated by white space) represents a different test set patient. Y: aortic wall and R, G, B: valve leaflets (Color figure online)

the boundary curves of the aortic wall and the leaflets, while denoting key landmark points of commissures and hinges. We then applied thin plate spline to deform mesh templates from a shape dictionary to the defined boundaries, and further adjusted nodes along the surface normals, using a combination of manually marked points on surface and intensity gradient-based objective, similar to [11]. The surface mesh template was created by further processing one of the ground truth meshes with a series of remeshing and stitching steps. The final template surface mesh was a single connected component mesh with 12186 nodes and 18498 faces, consisting of both triangular and quadrilateral faces.

3.2 Implementation Details

We used Pytorch [15] to implement a variation of a 3D U-net [18] as our function approximator $h_\theta(I; S_0, M_0)$. Since the network architecture is not the focus of this paper, we kept it consistent between all deep learning-based methods for fair comparison. The basic Conv unit was Conv3D-InstanceNorm-LeakyReLU, and the network had 4 encoding layers of ConvStride2-Conv with residual connections and dropout, and 4 decoding layers of Concatenation-Conv-Conv-Up-

sampling-Conv. The base number of filters was 16, and was doubled at each encoding layer and halved at each decoding layer. The output of the U-net was a $3 \times 64 \times 64 \times 64$ tensor, which we interpolated to obtain a $3 \times 24 \times 24 \times 24$ tensor that we then used to displace the control points of a 3D diffeomorphic third-order B-spline transformation to obtain a dense displacement field using the Airlab registration library [20]. The Chamfer distance and geometric quality losses were implemented using Pytorch3D [17] with modifications. We used the Adam optimizer [9] with a fixed learning rate of $1e-4$, batch size of 1, and 2000 training epochs. The models were trained with B-spline deformation augmentation step over 2000 epochs, resulting in 80k training samples and around 24 h of training time on a single NVIDIA GTX 1080 Ti.

3.3 Evaluation Metrics

For spatial accuracy, we evaluated the Chamfer distance (mean surface accuracy) and the Hausdorff distance (worst-case surface accuracy) between ground truth meshes and predicted surface meshes. The Chamfer distance was calculated using 10k sampled points on each mesh surface using Pytorch3D [17], and the Hausdorff distance was calculated using the meshes themselves using IGL [8]. For correspondence error, we evaluated the Euclidean distance between hand-labeled landmarks (3 commissures and 3 hinges) and specific node positions on the predicted meshes (e.g. commissure #1, was node #81 on every predicted mesh, etc.). Additionally, for bare minimum FEA viability, we used CGAL [12] to check for self-intersections and degenerate edges/faces of predicted volumetric meshes with no post-processing.

3.4 Comparison with an Image Intensity Gradient-Based Approach

We compared our method against the semi-automated version of [11], which uses manually delineated boundaries to first position the template meshes and refines them using an image gradient-based objective (Fig. 3). This approach performs very well under ideal imaging conditions, where there are clear intensity changes along the valve components, but it tends to make large errors in images with high calcification and low contrast. On the other hand, MTM does not particularly favor one condition or another, as long as enough variations exist in training images. However, this could also mean that it could make errors for “easy” images if the model has not seen a similar training image. We chose to not include Liang et al. [11] for evaluation metric comparisons because (1) it uses manually delineated boundaries, which skews both surface distance and correspondence errors and (2) we used some of its meshes as ground truth without any changes, when the images are in good condition (e.g. Fig. 3 bottom left).

3.5 Comparison with Other Deformation Strategies

We chose three deformation-based methods for comparison: Voxelmorph [1], U-net + Robust Point Matching (RPM) [2], and TETRIS [10] (Table 1, Fig. 4).

Table 1. Summary of evaluation metrics, mean \pm standard deviation across all test set patients and all 4 valve components. “Bad” column: percentage of predicted test set meshes with self-intersection or degeneracy with no post-processing. Lower is better for all metrics.

	Chamfer (mm)	Hausdorff (mm)	Commissure error (mm)	Hinge error (mm)	Bad (%)
Voxelmorph	3.01 \pm 1.38	9.51 \pm 3.04	6.80 \pm 3.93	7.15 \pm 4.36	100.00
U-net + RPM	0.99 \pm 0.32	4.57 \pm 1.59	2.25 \pm 1.27	2.38 \pm 1.15	76.32
TETRIS	0.95 \pm 0.30	4.35 \pm 1.99	1.84 \pm 1.10	2.32 \pm 1.38	5.26
MTMgeo (ours)	0.82 \pm 0.26	4.48 \pm 2.43	1.49 \pm 0.94	2.33 \pm 1.07	10.53
MTM (ours)	0.80 \pm 0.24	4.04 \pm 1.86	1.57 \pm 1.05	2.30 \pm 1.36	0.00

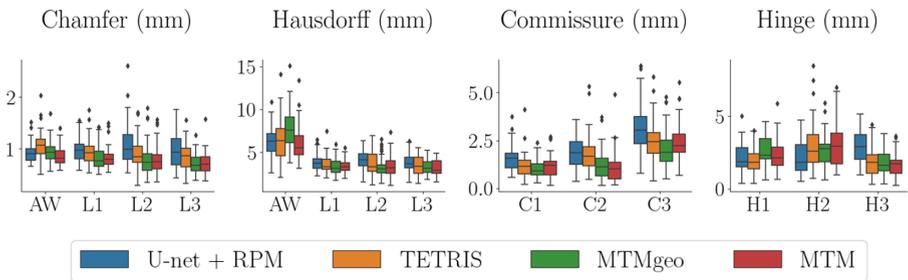


Fig. 4. Component-specific evaluation metrics (summarized in Table 1) for the four best performing methods. AW: aortic wall, L1, L2, L3: 3 leaflets, C1, C2, C3: 3 commissure points, H1, H2, H3: 3 hinge points

Voxelmorph [1] uses a CNN-predicted deformation field to perform intensity-based registration with an atlas image. Given a paired image-mesh template, we first trained for image registration and used the resulting field to deform the mesh to the target image. Since there is no guidance for the network to focus on the valve components, the resulting deformation is optimized for larger structures around the valve rather than the leaflets, leading to poor mesh accuracy.

U-net + RPM [2, 18] is a sequential model where we trained a U-net for voxel-wise segmentation and used its output as the target shape for registration with a template mesh. RPM (implemented in [14]) performs optimization during test time, which requires longer time and expert knowledge for parameter tuning during model deployment. It also produces suboptimal results, possibly due to the segmentation output and point sampling not being optimized for matching with the template mesh.

TETRIS [10] uses a CNN-predicted deformation field to deform a segmentation prior to optimize for segmentation accuracy. Using a paired segmentation-mesh template, we first trained for template-deformed segmentation and used the resulting field to deform the mesh. Since the field is not diffeomorphic and calculated in the reverse direction to prevent hole artifacts, we used VTK’s

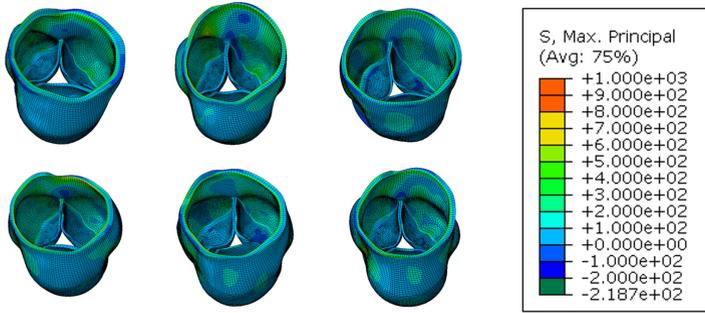


Fig. 5. FEA results using MTM-predicted meshes from 6 test set patients. Values indicate maximum principal stress in the aortic wall and leaflets during diastole.

implementation of Newton’s method [21] to get the inverse deformation field for the template mesh. The inaccuracies at the segmentation boundaries and errors due to field inversion lead to suboptimal performance.

MTM consistently outperforms all other deformation strategies in terms of spatial accuracy and produces less degenerate meshes (Table 1, Fig. 4). The accuracy values are also comparable to those in [7, 11], which use cleaner images, heavier assumptions about valve structure, and/or ground truth meshes for training. MTMgeo arguably performs similarly to MTM, suggesting that we may be able to replace field smoothness penalty with other mesh-related losses to refine the results for specific types of meshes. This may be especially useful for training with volumetric meshes, where we might want to dynamically adjust the thickness of different structures based on the imaging characteristics.

3.6 FEA Results

Volumetric FE meshes were produced by applying the MTM-predicted deformation field to a template volumetric mesh, which was created by applying a simple offset + stitch operation from the template surface mesh. We set aortic wall and leaflet thicknesses to 1.5 cm and 0.8 cm, respectively, and used C3D15 and C3D20 elements. FEA simulations were performed with an established protocol, similar to those in [11, 24]. Briefly, to compute stresses on the aortic wall and leaflets during valve closing, an intraluminal pressure ($P = 16$ kPa) was applied to the upper surface of the leaflets and coronary sinuses, and a diastolic pressure ($P = 10$ kPa) was applied to the lower portion of the leaflets and intervalvular fibrosa. The maximum principal stresses in the aortic wall and leaflets were approximately 100–500 kPa (Fig. 5), consistent with previous studies [11, 24]. This demonstrates MTM-predicted meshes’ suitability for FEA simulations.

3.7 Limitations and Future Works

Although MTM shows promise, it has much room for improvement. First, the current setup requires 3 manual landmark points during preprocessing for cropping and rigid alignment. We will pursue end-to-end learning using 3D whole-heart images via region proposal networks, similar to [13]. Second, our model does not produce calcification meshes, which are important for proper simulation because calcification and valve components have different material properties. We will need a non-deformation strategy for predicting calcification meshes since their size and position vary significantly between patients. Third, the restriction to smooth and diffeomorphic field prevents large variations in valve shapes. We will continue exploring the possibility of extending our framework to node-specific displacement vectors.

4 Conclusion

We presented a weakly supervised deep learning approach for predicting aortic valve FE meshes from 3D patient images. Our method only requires segmentation labels and a paired segmentation-mesh template during training, which are easier to obtain than mesh labels. Our trained model can predict meshes with good spatial accuracy and FEA viability.

Acknowledgments and Conflict of Interest. This work was supported by the NIH R01HL142036 grant. Dr. Wei Sun is a co-founder and serves as the Chief Scientific Advisor of Dura Biotech. He has received compensation and owns equity in the company.

References

1. Balakrishnan, G., Zhao, A., Sabuncu, M.R., Guttag, J., Dalca, A.V.: Voxelmorph: a learning framework for deformable medical image registration. *IEEE Trans. Med. Imaging* **38**(8), 1788–1800 (2019)
2. Chui, H., Rangarajan, A.: A new point matching algorithm for non-rigid registration. *Comput. Vis. Image Underst.* **89**(2–3), 114–141 (2003)
3. Dowling, C., Firoozi, S., Brecker, S.J.: First-in-human experience with patient-specific computer simulation of TAVR in bicuspid aortic valve morphology. *JACC Cardiovasc. Interv.* **13**(2), 184–192 (2020)
4. Fedorov, A., et al.: 3D slicer as an image computing platform for the quantitative imaging network. *Magn. Reson. Imaging* **30**(9), 1323–1341 (2012)
5. Ghesu, F.C., et al.: Marginal space deep learning: efficient architecture for volumetric image parsing. *IEEE Trans. Med. Imaging* **35**(5), 1217–1228 (2016)
6. Gkioxari, G., Malik, J., Johnson, J.: Mesh R-CNN. In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 9785–9795 (2019)
7. Ionasec, R.I., et al.: Patient-specific modeling and quantification of the aortic and mitral valves from 4-D cardiac CT and tee. *IEEE Trans. Med. Imaging* **29**(9), 1636–1651 (2010)

8. Jacobson, A., Panozzo, D., et al.: libigl: A simple C++ geometry processing library (2018). <https://libigl.github.io/>
9. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint [arXiv:1412.6980](https://arxiv.org/abs/1412.6980) (2014)
10. Lee, M.C.H., Petersen, K., Pawlowski, N., Glocker, B., Schaap, M.: Tetris: template transformer networks for image segmentation with shape priors. *IEEE Trans. Med. Imaging* **38**(11), 2596–2606 (2019)
11. Liang, L., et al.: Machine learning-based 3-D geometry reconstruction and modeling of aortic valve deformation using 3-D computed tomography images. *Int. J. Numer. Methods Biomed. Eng.* **33**(5), e2827 (2017)
12. Lorient, S., Rouxel-Labbé, M., Tournois, J., Yaz, I.O.: Polygon mesh processing. In: CGAL User and Reference Manual. CGAL Editorial Board, 5.1.1 edn. (2020). <https://doc.cgal.org/5.1.1/Manual/packages.html#PkgPolygonMeshProcessing>
13. Pak, D.H., Caballero, A., Sun, W., Duncan, J.S.: Efficient aortic valve multilabel segmentation using a spatial transformer network. In: 2020 IEEE 17th International Symposium on Biomedical Imaging (ISBI), pp. 1738–1742. IEEE (2020)
14. Papademetris, X., et al.: Bioimage suite: an integrated medical image analysis suite: an update. *Insight J.* **2006**, 209 (2006)
15. Paszke, A., et al.: Automatic differentiation in pytorch (2017)
16. Pouch, A.M., et al.: Medially constrained deformable modeling for segmentation of branching medial structures: application to aortic valve segmentation and morphometry. *Med. Image Anal.* **26**(1), 217–231 (2015)
17. Ravi, N., et al.: Accelerating 3D deep learning with pytorch3d. [arXiv:2007.08501](https://arxiv.org/abs/2007.08501) (2020)
18. Ronneberger, O., Fischer, P., Brox, T.: U-Net: convolutional networks for biomedical image segmentation. In: Navab, N., Hornegger, J., Wells, W.M., Frangi, A.F. (eds.) MICCAI 2015, Part III. LNCS, vol. 9351, pp. 234–241. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-24574-4_28
19. Rueckert, D., Sonoda, L.I., Hayes, C., Hill, D.L., Leach, M.O., Hawkes, D.J.: Non-rigid registration using free-form deformations: application to breast MR images. *IEEE Trans. Med. Imaging* **18**(8), 712–721 (1999)
20. Sandkühler, R., Jud, C., Andermatt, S., Cattin, P.C.: Airlab: Autograd image registration laboratory. arXiv preprint [arXiv:1806.09907](https://arxiv.org/abs/1806.09907) (2018)
21. Schroeder, W.J., Lorensen, B., Martin, K.: The Visualization Toolkit: An Object-Oriented Approach to 3D Graphics. Kitware, New York (2004)
22. Sun, W., Martin, C., Pham, T.: Computational modeling of cardiac valve function and intervention. *Annu. Rev. Biomed. Eng.* **16**, 53–76 (2014)
23. Wang, N., Zhang, Y., Li, Z., Fu, Y., Liu, W., Jiang, Y.G.: Pixel2Mesh: generating 3D mesh models from single RGB images. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 52–67 (2018)
24. Wang, Q., Primiano, C., McKay, R., Kodali, S., Sun, W.: CT image-based engineering analysis of transcatheter aortic valve replacement. *JACC Cardiovasc. Imaging* **7**(5), 526–528 (2014)
25. Zhuang, X., Shen, J.: Multi-scale patch and multi-modality atlases for whole heart segmentation of MRI. *Med. Image Anal.* **31**, 77–87 (2016)