AN OPTICAL FLOW BASED APPROACH TO VALIDATING DYNAMIC STRUCTURAL FINITE ELEMENT MODELS OF BIOLOGICAL ORGANS USING 4D MEDICAL IMAGES – THE AORTIC VALVE AS AN EXAMPLE

by

EMMA GIBNEY

Thesis submitted to the University of Ottawa
In partial fulfillment of the requirements for the
Master of Applied Science in Biomedical Engineering

The Ottawa-Carleton Institute for Biomedical Engineering
University of Ottawa
November 2021

ABSTRACT

Recent developments within the biomedical engineering field of using finite element methods to analyze biological structures has resulted in a need for a standardized method to validate these models. The purpose of this thesis was to develop a system to effectively and efficiently validate biological finite element models using 4D medical images. The aortic valve was chosen as the biological model for testing as any solution that could manage the complexity of the valve's motion would likely work for simpler biological models. The proposed validation method involved 3 steps: estimating a voxel displacement field using a direct method of 3D motion estimation, converting the voxel displacement field into a nodal displacement field, and validating the results of a finite element model by comparing the nodal displacement field of the finite element model to the nodal displacement field from the medical images. The proposed validation method was implemented using synthetic 4D CT images of an aortic valve based on an existing finite element model, where the ground truth was the results of the existing finite element model. Three different direct motion estimation methods were implemented within the first step of the method and compared. The three methods were: 3D Horn-Schunck optical flow, 3D Brox optical flow, and demons method. The addition of a multilevel scheme with a variable scale constant was integrated into each of these motion estimation methods so that larger magnitudes of displacement could by captured. It was found that Horn-Schunck optical flow was best able to capture the motion of the aortic valve throughout a cardiac cycle. The proposed method of validation was able to track the aorta nodes effectively through an entire cardiac cycle and was able to track leaflet nodes through large displacements until the valve closed. Although the general trend of the motion of the aortic valve was captured by the validation method using synthetic medical images, node-to-node comparison was not entirely reliable. Comparison of the general trend was still superior to the current validation methods for biological finite element methods as it considered the motion of the entire structure.

ACKNOWLEDGEMENTS

I would like to thank my supervisor, Dr. Michel Labrosse, for providing guidance and feedback throughout this project. Thanks also to my family and friends for their continued support and encouragement during this time.

1 TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1 INTRODUCTION

1.1 RATIONALE

The analysis of biological organs in dynamics is important because it has the potential to impact how personalized healthcare can be approached. For example: if the dynamics of an individual's aortic valve can be understood and recreated as a simulation it can be used as a digital twin [1]. In theory, virtual surgeries carried out on a digital twin of an aortic valve could provide the surgeon with possible, and optimal scenarios for the restoration of the valve function in the case of aortic valve repair [1]. As a foundation for constructing digital twins, finite element analysis has been a tool in engineering design since the 1950s [2]. Finite element analysis is used to simulate physical phenomena. The geometry of interest is discretized into hundreds, or thousands of smaller pieces called finite elements. The elements are connected to adjacent elements through nodes. Nodes are coordinate locations where the degrees of freedom (e.g. displacements) are defined. Given a force applied at one node in the geometry, the resulting stress and strain can be determined at every other node based the mathematical equations of how the nodes relate to each other. Finite element analysis is useful in mechanical engineering applications because the stress and strain fields of complex geometries can be solved quickly. As computing has gotten more powerful over the years, the applications of finite element analysis have broadened to include biological analysis [2]. While finite element analysis has proved valuable in mechanical engineering, there is still hesitancy from medical professionals that it can be effectively used as a tool for assessing biological organs. Validation is a critical element in all finite element analyses. It is particularly important for biological applications where the complexity of the material properties increases the chance of error [2]. The gold standard for validation of biological finite element models is in-vivo strain gauges [3]. When modeling biological tissues using finite element analysis, in-vivo strain gauges are often not feasible. Instead, in-vitro experiments of simplified models are used. Simplified in-vitro experiments are good to obtain general information about an organ for validation but are not useful for validation patient-specific models of organs. For models of a specific person's tissue, 4-dimensional (4D) medical images such as CT, ultrasound, or MRI, are often the only available information. 4D medical images are 3D medical images with time being the fourth dimension. Currently, there is no standard method for using these medical images to validate biological finite element models. This was the motivation of the present work. There are two options to use medical images to validate dynamic finite element models: segmentation at different times, or motion estimation. By comparison, segmentation is time-consuming, and does not necessarily track the same image features over time, which could raise problems for the computation of strains. For this reason, a motion estimation method was the focus of the present work.

1.2 Objective

Motivated by the lack of a standard effective method for the validation of dynamic biological finite models, the objective of this thesis is to work towards a method of validation for such models using patient specific 4D medical imaging. The initial objective is to create an effective validation method using synthetic 4D medical images obtained from an existing dynamic finite element model of an aortic valve. The purpose of using an existing finite element model to create synthetic medical images is for the existing model to act as the known ground truth. If real images were used initially, there would be no way to validate the validation method because there would be no known results to compare to. With synthetic images, the effectiveness of the validation method can then be determined by comparing the results of the validation method to the ground truth from the existing model. The other advantage to using synthetic images is there is no noise, or alignment issues. The aortic valve was chosen as the biological model to test on as any solution that could manage the complexity and magnitude of the valve's motion would likely be transferable to more simple biological models.

1.3 Knowledge Gap

Currently, no method exists for using 4D medical images to validate biological finite element models. A concept for such a method was the main knowledge gap. The concept of the proposed method can be broken down into components: determining the voxel displacement field, converting the voxel displacement field into a nodal displacement field, and determining weather the finite element nodal displacement field agrees with the ground truth. In our context, determining the voxel displacement field involved identifying a motion estimation technique that could capture the range of motion of the aortic valve. The motion estimation techniques chosen were the 3D Horn-Schunck optical flow within a pyramid scheme, the 3D Brox optical flow, and the demons method within a pyramid scheme. Open-source files existed for the implementation of 3D Horn-Schunck optical flow, and the theory for the extension of Horn-Schunck optical flow from 2D images to 3D images was documented [4] [5]. The theory and implementation of the demons method was documented and built into a function [6] [7]. The theory for the extension of Brox optical flow from 2D images to 3D images was documented. However, no open-source code or built-in functions implemented the theory [8]. The theory of the implementation of Brox also included the theory of the addition of a multiscale approach with a variable scale constant to capture larger magnitudes of pixel displacement was documented for 2D cases by Brox [9], and documented for 3D cases by Chen [8]. The implementation of the multi-scale approach existed in open-source code for 2D images, but not 3D images. The knowledge gaps relating to the motion estimation methods were: the implementation of the 3D Brox optical flow method, and the integration of the multiscale approach with a variable scale constant to both the 3D Horn-Schunck optical flow method, and the demons method. There was also a knowledge gap in how to convert a voxel displacement field into a nodal displacement field.

1.3.1 Proposed Improvement 1 – The Concept of a Method to Validate Finite Element Models using Medical Images

The basis of this work is the lack of a sufficient, or standard validation method for biological finite element models. Often the only information available for validation is medical images. A method of validation using only medical images is proposed. The method involves the following steps. First, use a motion estimation method to determine a voxel displacement field between the base image (the image that the finite element model is segmented from) and any subsequent image. Second, convert the voxel displacement field into a nodal displacement field. Third, compare the results of the determined nodal displacement field to the results of the nodal displacement field from the finite element model.

1.3.2 Proposed Improvement 2 - Using a Multiscale Approach with a Variable Scale Constant with 3D Horn-Schunck Optical Flow, and Demons Method

The motion estimation methods of 3D Horn-Schunck optical flow, and demons method are unable to capture the motion of the aortic valve analysing only the images at the given size and resolution. For this reason, a pyramid scheme, or multilevel scheme, will be added to increase the magnitude of displacements that can be captured. The multilevel scheme with a variable scale constant is a concept used in Brox optical flow. The same principle can be applied to the other motion estimation methods as well. The built-in MATLAB function for the demons method of motion estimation does have a built-in pyramid scheme; however, it does not have a variable scale constant. Without the variable scale constant, it is not effective in capturing the motion of an aortic valve.

1.3.3 Proposed Improvement 3 – Implementation of 3D Brox Optical Flow

The theory of the conversion of Brox optical flow from 2D to 3D was outlined by Chen [15]. The 3D implementation is a proposed improvement in this work. Brox optical flow adds an additional constraint to Horn-Schunck optical flow to increase the accuracy of the motion estimation method.

1.3.4 Proposed Improvement 4 -Tuning Optical Flow Parameters Separately for Valve Leaflets and Aorta

The biological structure of focus in this thesis was the aortic valve. Due to its complexity, the optical flow input variables, alpha and gamma, will be tuned separately for the aorta and the valve leaflets in order to ensure optimal performance. A tuning process involving comparison of warped and original images will be theorized and implemented. Using multiple values for alpha and gamma within an analysis is unique to this thesis and only possible due to the conversion from voxel displacement field to nodal displacement field.

1.3.5 Proposed Improvement 5 – Method for Converting a Voxel Displacement Field into a Nodal Displacement Field

Part of the implementation of the proposed validation method is the step of converting a voxel displacement field into a nodal displacement field. A method for converting a voxel displacement field into a nodal displacement field is proposed for this work. The nodal displacement field is overlayed on the voxel displacement field. Nodes are assigned to voxels based on the brightness intensity of the voxels, and the nodes proximity to the information containing voxels.

1.4 CONTRIBUTIONS

This thesis introduces the concept of a method to validate biological finite element models using 4D medical images. A direct method of motion estimation is used to determine a voxel displacement field. The medical images are made up of voxels, so the motion estimation methods used result in a voxel displacement field. The finite element model is made up of nodes, so in order for the two results to be compared, the voxel displacement field must be converted into a nodal displacement field. The nodal displacement field can be directly compared to the finite element analysis results. Three methods of motion estimation are evaluated as candidates to determine the voxel displacement field: 3D Horn-

Schunck optical flow, 3D Brox optical flow, and the demons method, all three within a pyramid scheme. The parameter inputs into the motion estimation methods are selected by optimization within the validation method.

1.5 Thesis Organization

The organization of this thesis is as follows. Section 2 introduces and reviews the relevant background topics in detail. These topics include existing methods of validation of biological finite element models, image processing techniques for motion estimation, and detailed overviews of the implementation of Horn-Schunck optical flow, Brox optical flow, and the demons method for motion estimation. Section 3 describes the method used to determine the nodal displacements of a finite element model from ideal synthetic medical images using the three methods of motion estimation. Section 4 shows the results of the method in comparison to the ground truth. Section 5 is a discussion on the validity, representativeness and predictive power of the results.

2 LITERATURE REVIEW

The concept of creating a validation method for finite element models using medical images requires background information and an overview of the literature from a variety of topics. The first topic to review is the current practice of validation for biological finite element models. Section 2.1 is an overview of the current validation methods of biological finite element models in both a general sense in Section 2.1.1, and specific to the aortic valve in Section 2.1.2. The second topic to review is the current image processing techniques for motion estimation. Section 2.2 covers a brief overview about the different categories of methods for motion estimation. Section 2.2.1 gives a detailed explanation about the different types of optical flow methods with subsections that break down the details of the application of Horn-Schunck optical flow, and Brox optical flow. Section 2.2.2 provides an overview of the demons algorithm as an alternative method of motion estimation. Finally, Section 1.3 outlines the knowledge gap in the relation between validation of finite element analysis, and the use current motion estimation methods.

2.1 VALIDATION OF BIOLOGICAL FINITE ELEMENT MODELS

Finite element analysis was introduced as a technique for engineering design in the 1950's, and by the 1970's the solid and fluid mechanics principles were being applied to biological tissues [2]. Due to the complex material properties of biological tissues, traditional approaches of describing engineering materials were not sufficient to represent the biological tissues [2]. For this reason, constitutive models were developed for specific tissues in an attempt to describe the material, however the added complexity and specificity increased the chance for errors [2]. The increased potential for errors highlights the need for verification and validation of biological finite element models. The American Society of Mechanical Engineers released a guide in 2006 outlining the expectation for verification and validation in computational solid mechanics, including finite element analysis [10]. Validation is defined as, “The process of determining the degree to which a model is an accurate representation of the real world from the perspective of the intended uses of the model” [10]. In this guide it is stated that, “validation must assess the predictive capability of the model in the physical realm of interest” [10]. Validation is the process of comparing simulation outcomes to experimental outcomes [3]. If the comparisons are acceptable, the model is considered validated and can be trusted in its intended use [10]. In the real-world, models are often created for applications where no experimental data could exist. In these cases, it is acceptable to assume a model to be validated if it can effectively predict a simplified case [10]. This is a common occurrence with finite element models for biological applications. The simplified models are quantified in-vitro experiments for in-vivo systems [2]. The comparison of the validation experiment and simulation results is measured by validation metrics, which are quantitative values with an associated confidence [10]. Section 2.1.1 will provide an overview of current methods researchers are using to meet the validation criteria for finite element models of different types of biological tissue. Section 2.1.2 will provide an overview of the current methods researchers are using to meet the validation criteria of finite element models of aortic valves specifically.

2.1.1 Current Validation Methods of Biological Finite Element Models

The results of finite element models of biological systems are more susceptible to errors compared to strictly mechanical structures because the validity of the solutions is dependent on input variables of the biological systems that are more complex to describe [3]. These variables include geometry, material properties, applied forces, and constraints [3]. Some biologists have been suspicious of finite element methods due to the reliance on complex inputs, which is why validation is an important component. In vivo strain gauges are the gold standard for validation of biological finite element models [3]. An in vivo strain gauge is able to measures strain on complex structures which can be directly compared to in silico results [3]. If the in vivo model and the in-silico model deform the same way under the same forces, the results can be considered valid. In vivo strain gauges have been used successfully on animal bones and dental applications as a validation tool for finite element modeling [11] [12]. Gao et al was able to use in vivo strain gauges on rat tibias to evaluate strain distribution under axial compressive loads [11]. Reddy et al. used in vivo strain gauges on molar implants to explore the application of finite element modeling in dentistry [12]. Unfortunately, for many biological applications of finite element modeling, it is too invasive to place in-vivo strain gauges for them to be a viable validation tool [3].

In vitro strain gauges can also be used as an alternative way of validating the finite element inputs of geometry and material properties [3]. In vitro strain gauges can also be difficult to bond to different types of biological materials without affecting them structurally [3]. In certain applications in vitro strain gauges can effectively be used as a validation tool. Takacs et al. used in vitro strain gauges to validate a finite element model of an osteoporotic human dry femoral bone [13].

As an alternative to strain gauges, full field strain analysis methods can be used as a validation method for finite element models [3]. Speckle interferometry is a noncontact system used to find three-dimensional data on stress and strain [3]. Groning et al used digital speckle pattern interferometry to measure strains in an ex vivo human mandible as a validation method for a finite element model [14].

2.1.2 Current Validation Methods for Finite Element Models of the Aortic Valve

Compared to motion of other biological tissue, the motion of the aortic valve is very complex. The magnitude of displacement of the valve leaflets with respect to the size of the valve is large, and the deformation of the leaflets as they open and close is also large. These factors make the aortic valve difficult to model using finite element methods, as well as make the models difficult to validate thoroughly. The following sections review examples of how researchers who have modeled aortic valves using finite element methods have validated the results using in vitro experiments.

Ghareie et al. modeled a polymeric aortic valve's non-linear structural deformation under physiological conditions using computational methods [15]. They also completed in vitro testing of the polymeric aortic valve to validate their findings [15]. The researchers used three methods to validate the results.

First, a quantitative comparison between the leaflets' displacement from the experimental data and the simulation model as shown in Figure 1 [15]. The researchers matched images of the valve opening from the in-vitro experiment at specific time steps throughout the simulated cardiac cycle with the computational model at the same time step and assessed whether the shape of the valve openings were comparable [15].

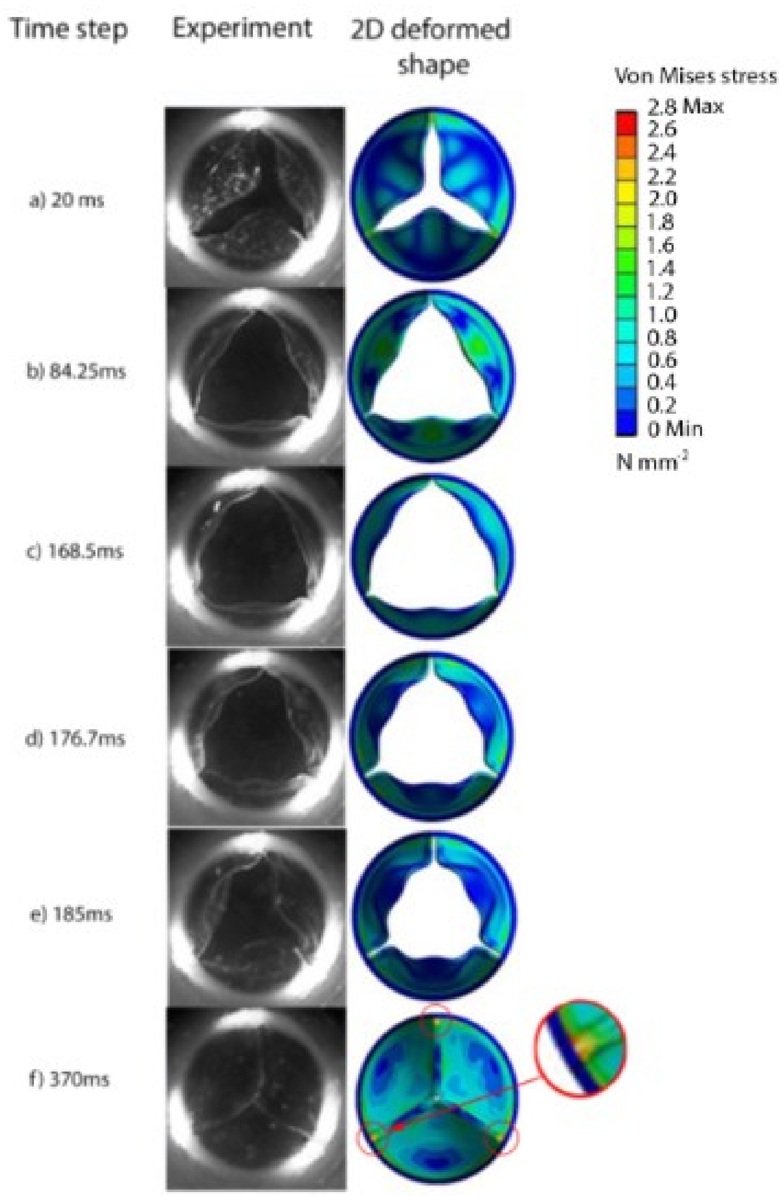


Figure 1: Qualitative Comparison between leaflets' displacement from the invitro data (left) and finite element predictions (right) [15]

Second, three metrics for distance measurements were defined as the distance between tips of each of the three leaflets and the center of the valve [15]. These distance metrics are shown in Figure 2 [15]. The measurements were tracked over the cycle of the experiment by a high-speed camera and plotted against time [15]. Similarly, the same distance measurements were tracked in the computational simulation and plotted on the same graph as shown in Figure 3 [15].

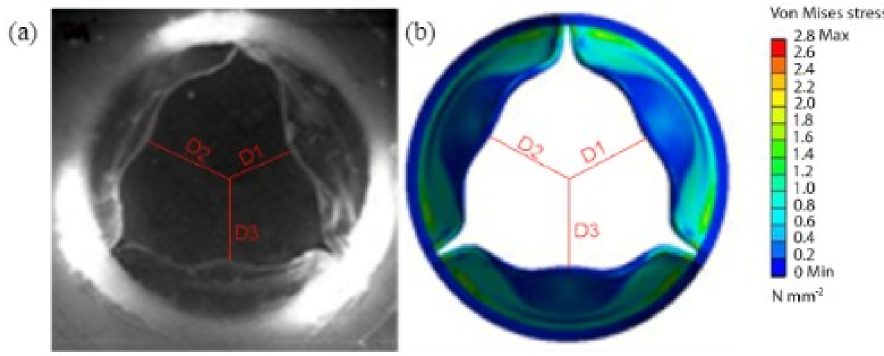


Figure 2: Three distance measurement metrics for displacement comparisons between experimental and simulated aortic valves [15]

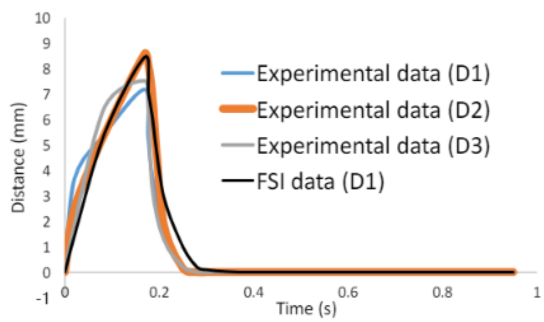
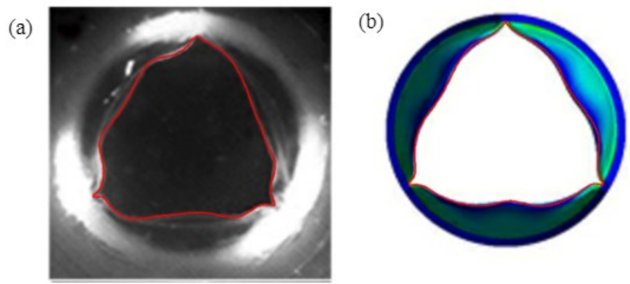


Figure 3: Comparison of displacement measurement metrics between experimental and simulated aortic valves vs time [15]

Lastly, a comparison of the geometric orifices' areas shown in Figure 4 was used as a validation method [15]. The geometric orifice area at peak systole were measured in the in-vitro experiment and compared to the predicted area from the simulated valve [15].



GOA experimental (cm^2)	Predicted GOA (cm^2)	Percentage error (%)
3.13	3.31	5.7

Figure 4: Geometric orifice area (GOA) comparison of experimental and simulated results at peak systole [15]

Sodhani et al. studied artificial textile reinforced aortic valves using simulation to study the fluid structure interaction [16]. They also completed in-vitro tests with aortic flow and pressure based on the standard [16]. The results for displacement were validated by comparison to the in-vitro experimental results using 4 methods. The first method was a comparison of simulation flow rate over a cycle vs. in vitro experiment flow rate over a cycle as shown in part (a) of Figure 5 [16]. The second method was a comparison of simulation geometric orifice area over a cycle vs. in vitro experiment geometric orifice area over a cycle as shown in part (b) of Figure 5 [16]. This was similar to the validation strategy by Ghareie et al. [15]. The third method was a comparison of radial displacements of three points on the valve at 9 times throughout the cycle compared to the in-vitro experiment valve deformation as shown in part (c) of Figure 5 [16]. The final method was a visual overlay of simulation valve displacements and experimental valve displacements as shown in part (d) of Figure 5 [16].

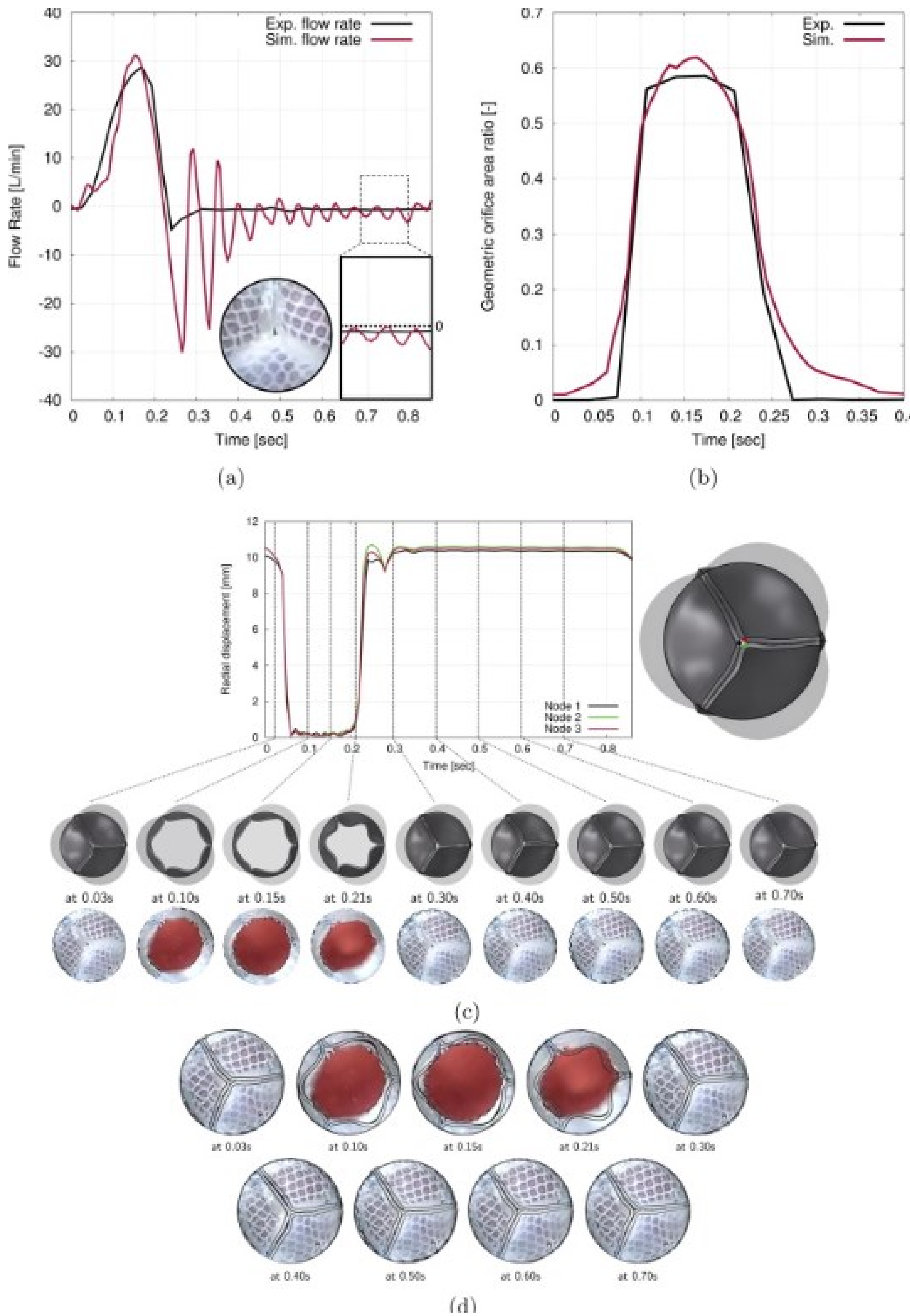


Figure 5: Validation methods: (a) comparison of simulation flow rate over a cycle vs. in-vitro experiment flow rate over a cycle (b) comparison of simulation geometric orifice area over a cycle vs. in-vitro experiment geometric orifice area over a cycle (c) comparison of radial displacements of three points on the valve at 9 times throughout the cycle compared to the in-vitro experiment valve deformation (d) visual overlay of simulation valve displacements and experimental valve displacements [16]

2.1.2.1 *Limitations of Current Validation Methods*

Current biological finite element studies, particularly of the aortic valve, have implemented validation methods that use simple performance criteria as a way to validate their findings. Simple performance criteria should not be considered sufficient validation because they leave the possibility of having met all the individual criteria while still having a poor model. For example, if the validation method for a model of an aortic valve only considers the criteria of geometric orifice area, and normal distance from the tips of the valve leaflets to the center of the valve, the simulated valve could be twice the height of the real valve and would be considered a valid model. The limitations of the current validation methods for biological finite element models reveal the need for a more robust method of validation.

2.2 Image Processing Techniques for Motion Estimation

4D medical images provide invaluable information in terms of how organs displace and deform over time. Therefore, it is important to discuss motion estimation. Motion estimation is the concept that consecutive frames of a video will be similar, aside from the changes caused by objects in the frame moving [17]. There are two basic branches of motion estimation: feature based methods, and direct methods [17]. Feature based methods extract visual features such as corners or textures and track those features across consecutive frames [17]. The positive aspect of using feature-based motion estimation is that it can effectively track larger motion where there is significant displacement of the pixels across frames [17]. The negative aspect of using feature-based motion estimation is that it does not track all the points, only features of interest [17]. This results in a sparse motion field [17]. For this reason, it will not be investigated further. Direct methods track the motion of each pixel of an image based on image brightness variations through time and space [17]. The positive aspects of using direct methods for motion estimation is that the result is a dense motion field which contains information about the displacement of each pixel in the image [17]. The negative aspect of using a direct method for motion estimation is it can have difficulty capturing larger displacements (typically on the order of magnitude of tens of pixels) [17]. Optical flow and demons method, discussed next, are both considered direct methods for motion estimation.

2.2.1 Optical Flow

Optical flow is the apparent motion distribution of an image based on spatiotemporal image brightness patterns [18]. Optical flow is a technique that has been used in computer science for years to determine motion fields of sequential imaged [4]. Optical flow is a common tool that has been used in a wide array of industries. Some of the current industrial applications of optical flow include; driver assistance systems, video game controllers, and movie special effects [19]. The method to compute a displacement field between 2D images was introduced by Horn and Schunck in 1981 [4]. In their method, a global solution was proposed using the minimization of an energy function [4]. The energy function combines the assumption of brightness constancy and a smooth transition of the velocity field detailed in Section 2.2.1.1.3 [18]. Since Horn-Schunck's work, many other methods for optical flow have been published. Lucas and Kanade proposed a local solution using specific sub regions of a 2D image [20]. Brox proposed a global solution with improved capability to track large displacement fields [9]. Brox's method added the gradient constancy assumption as another constraint to the energy function in the minimization equation [9]. The assumption states that the gradient of the image grey value does not vary over time, which allows for slight changes in brightness to be accounted for [9]. Brox also implemented a multiscale approach which uses coarse to fine estimation of motion to ensure a global minimum to the energy functional is found rather than a local minimum [9]. A more detailed overview of the Brox method for optical flow is given in Section 2.2.1.2. These methods have since been extended to 3D image fields [4, 8]. This is useful in the field of medical imaging with MRI, CT, and ultrasound data, as relevant herein.

2.2.1.1 Horn-Schunck Optical Flow Method

Horn and Schunck introduced a method of determining optical flow in two-dimensional sequential images [18]. Their method was based on their observation that there are two components of flow velocity [18]. First, the basis of optical flow, the brightness constancy assumption, otherwise known as the grey value constancy assumption which is detailed in Section 2.2.1.1.1. The second assumption, unique at the time to the Horn-Schunck model, was the smoothness constraint which is detailed in Section 2.2.1.1.2 [18]. They combined these functions into an energy functional, and were able to solve for the flow field by minimizing the energy functional by an iterative method detailed in Section 2.2.1.1.3 [18].

2.2.1.1.1 Brightness Constancy Assumption

The brightness constancy assumption assumes that the brightness intensity of a pixel remains constant over a small interval of time [18]. In other terms, the gray value of a pixel does not change when it moves to another location [8]. This assumption is shown in Eq. 1 where I represents the brightness intensity, u and v are the displacements in the x and y directions respectively, and t is the time. Using Taylor series expansion, the displacement vector of any given pixel can be related to the spatial and temporal gradients of the brightness intensity of the pixel by Eq. 2 [18]. In the real world, the brightness intensity cannot be expected to remain constant as image noise is unavoidable. This was accounted for by the energy function in Eq. 3 that can be minimized from the flow vectors across the entire flow field [18].

$$I(x+u, y+v, t+1) = I(x, y, t) \qquad \text{Eq. 1}$$

$$\frac{\partial I}{\partial x}u + \frac{\partial I}{\partial y}v = \frac{-dI}{dt}, i.e. \nabla I \cdot \begin{bmatrix} u \\ v \end{bmatrix} = \frac{-dI}{dt} \qquad \text{Eq. 2}$$

$$E_{data}(u, v) = \int \left(\frac{\partial I}{\partial x}u + \frac{\partial I}{\partial y}v + \frac{\partial I}{\partial t}\right)^2 dxdydt \qquad \text{Eq. 3}$$

2.2.1.1.2 Smoothness Assumption

The single constraint of brightness constancy is not enough to solve for the two unknowns, *u* and *v*, so a second constraint was introduced. The smoothness constraint is also useful in negating any issues with the aperture problem which occurs if only the flow in the normal direction to the gradient is estimated, or if the gradient in the brightness disappears [18]. The smoothness constraint imposes that the velocities of neighbouring pixels should vary smoothly [18]. This was applied by using an energy minimization function shown in Eq. 4 [18].

$$E_{smooth}(u,v) = \int \left(\frac{\partial u}{\partial x}\right)^2 + \left(\frac{\partial u}{\partial y}\right)^2 + \left(\frac{\partial v}{\partial x}\right)^2 + \left(\frac{\partial v}{\partial y}\right)^2 dxdydt \qquad \text{Eq. 4}$$

2.2.1.1.3 The Energy Functional

The Horn-Schunck method energy function was derived by combining Eq. 3 and Eq. 4 as shown in Eq. 5 to obtain Eq. 6. A variable α is added as a regularization constant that can vary the degree of smooth velocity transitions [18]. Optical flow of a given set of images can be solved by minimizing the energy function in an iterative manner.

$$E(u,v) = E_{data}(u,v) + \alpha E_{smooth}(u,v) \qquad \text{Eq. 5}$$

$$E(u,v) = \int \left\{ \left(\frac{\partial I}{\partial x}u + \frac{\partial I}{\partial y}v + \frac{\partial I}{\partial t}\right)^2 + \alpha\left(\left(\frac{\partial u}{\partial x}\right)^2 + \left(\frac{\partial u}{\partial y}\right)^2 + \left(\frac{\partial v}{\partial x}\right)^2 + \left(\frac{\partial v}{\partial y}\right)^2\right)\right\} dxdy \qquad \text{Eq. 6}$$

The Horn-Schunck has been extended to 3D cases using the energy function shown in Eq. 7 [4]. In the 3D case, w represents the displacement in the third-dimension, z. This equation can similarly be written as Eq. 8, where the spatial variant, ∇= $[\partial_x \;\; \partial_y \;\; \partial_z]^T$ [4].

$$\begin{aligned} E(u,v,w) = \int \Bigg\{ &\left(\frac{\partial I}{\partial x}u + \frac{\partial I}{\partial y}v + \frac{\partial I}{\partial z}w + \frac{\partial I}{\partial t}\right)^2 \\ &+ \alpha\left(\left(\frac{\partial u}{\partial x}\right)^2 + \left(\frac{\partial u}{\partial y}\right)^2 + \left(\frac{\partial u}{\partial z}\right)^2 + \left(\frac{\partial v}{\partial x}\right)^2 + \left(\frac{\partial v}{\partial y}\right)^2 + \left(\frac{\partial v}{\partial z}\right)^2 + \left(\frac{\partial w}{\partial x}\right)^2 + \left(\frac{\partial w}{\partial y}\right)^2 + \left(\frac{\partial w}{\partial z}\right)^2\right)\Bigg\} dxdydz \end{aligned} \qquad \text{Eq. 7}$$

$$E(u,v,w) = \int \left\{ \left(\frac{\partial I}{\partial x}u + \frac{\partial I}{\partial y}v + \frac{\partial I}{\partial z}w + \frac{\partial I}{\partial t}\right)^2 + \alpha(\|\nabla u\|^2 + \|\nabla v\|^2 + \|\nabla w\|^2)\right\} dxdydz \qquad \text{Eq. 8}$$

In order to minimize the integrand of the energy function, the derivatives with respect to u, v, and w are set to zero as shown in Eq. 9, Eq. 10, and Eq. 11 [4]. In a finite difference sense, the Laplacian of u, $\nabla^2 u$, can be approximated as $\bar{u} - u$, where $\bar{u}$ is the 7X7X7 voxel average displacement [18]. This approximation is based on the definition of the Laplacian, the divergence of the gradient, which relates the value of a point to the value of its neighbours or the voxel average. Similarly, $\nabla^2 v = \bar{v} - v$, and

$\nabla^2 w = \overline{w} - w$ [4]. Each of the 3 equations, Eq. 9, Eq. 10, and Eq. 11, can be written in terms of their displacement component, u, v, and w respectively. With three equations and three unknowns the solution to the system of equations is shown in Eq. 12, Eq. 13, and Eq. 14 [4].

$$\frac{dE}{du} = 2\frac{\partial I}{\partial x}\left(u\frac{\partial I}{\partial x} + v\frac{\partial I}{\partial y} + w\frac{\partial I}{\partial z} + \frac{\partial I}{\partial t}\right) + 2\alpha(\nabla^2 u) = 0 \quad \text{Eq. 9}$$

$$\frac{dE}{dv} = 2\frac{\partial I}{\partial y}\left(u\frac{\partial I}{\partial x} + v\frac{\partial I}{\partial y} + w\frac{\partial I}{\partial z} + \frac{\partial I}{\partial t}\right) + 2\alpha(\nabla^2 v) = 0 \quad \text{Eq. 10}$$

$$\frac{dE}{dw} = 2\frac{\partial I}{\partial z}\left(u\frac{\partial I}{\partial x} + v\frac{\partial I}{\partial y} + w\frac{\partial I}{\partial z} + \frac{\partial I}{\partial t}\right) + 2\alpha(\nabla^2 w) = 0 \quad \text{Eq. 11}$$

$$\left(\left(\frac{\partial I}{\partial x}\right)^2 + \left(\frac{\partial I}{\partial y}\right)^2 + \left(\frac{\partial I}{\partial z}\right)^2 + \alpha^2\right)(u - \bar{u}) = -\left(\frac{\partial I}{\partial x}\right)\left(\left(\frac{\partial I}{\partial x}\right)\bar{u} + \left(\frac{\partial I}{\partial y}\right)\bar{v} + \left(\frac{\partial I}{\partial z}\right)\overline{w} + \left(\frac{\partial I}{\partial t}\right)\right) \quad \text{Eq. 12}$$

$$\left(\left(\frac{\partial I}{\partial x}\right)^2 + \left(\frac{\partial I}{\partial y}\right)^2 + \left(\frac{\partial I}{\partial z}\right)^2 + \alpha^2\right)(v - \bar{v}) = -\left(\frac{\partial I}{\partial y}\right)\left(\left(\frac{\partial I}{\partial x}\right)\bar{u} + \left(\frac{\partial I}{\partial y}\right)\bar{v} + \left(\frac{\partial I}{\partial z}\right)\overline{w} + \left(\frac{\partial I}{\partial t}\right)\right) \quad \text{Eq. 13}$$

$$\left(\left(\frac{\partial I}{\partial x}\right)^2 + \left(\frac{\partial I}{\partial y}\right)^2 + \left(\frac{\partial I}{\partial z}\right)^2 + \alpha^2\right)(w - \overline{w}) = -\left(\frac{\partial I}{\partial z}\right)\left(\left(\frac{\partial I}{\partial x}\right)\bar{u} + \left(\frac{\partial I}{\partial y}\right)\bar{v} + \left(\frac{\partial I}{\partial z}\right)\overline{w} + \left(\frac{\partial I}{\partial t}\right)\right) \quad \text{Eq. 14}$$

These equations can be manipulated into an iterative solution scheme by using the average displacements of the voxels $\bar{u}$, $\bar{v}$, and $\overline{w}$ at step n, to estimate the displacement of each voxel u, v, and w at the next step, n+1 [4]. This is shown in Eq. 15, Eq. 16, and Eq. 17.

$$u^{n+1} = \bar{u}^n - \frac{\left(\frac{\partial I}{\partial x}\right)\left(\left(\frac{\partial I}{\partial x}\right)\bar{u}^n + \left(\frac{\partial I}{\partial y}\right)\bar{v}^n + \left(\frac{\partial I}{\partial z}\right)\bar{w}^n + \left(\frac{\partial I}{\partial t}\right)\right)}{\left(\left(\frac{\partial I}{\partial x}\right)^2 + \left(\frac{\partial I}{\partial y}\right)^2 + \left(\frac{\partial I}{\partial z}\right)^2 + \alpha^2\right)} \qquad \text{Eq. 15}$$

$$v^{n+1} = \bar{v}^n - \frac{\left(\frac{\partial I}{\partial y}\right)\left(\left(\frac{\partial I}{\partial x}\right)\bar{u}^n + \left(\frac{\partial I}{\partial y}\right)\bar{v}^n + \left(\frac{\partial I}{\partial z}\right)\bar{w}^n + \left(\frac{\partial I}{\partial t}\right)\right)}{\left(\left(\frac{\partial I}{\partial x}\right)^2 + \left(\frac{\partial I}{\partial y}\right)^2 + \left(\frac{\partial I}{\partial z}\right)^2 + \alpha^2\right)} \qquad \text{Eq. 16}$$

$$w^{n+1} = \bar{w}^n - \frac{\left(\frac{\partial I}{\partial z}\right)\left(\left(\frac{\partial I}{\partial x}\right)\bar{u}^n + \left(\frac{\partial I}{\partial y}\right)\bar{v}^n + \left(\frac{\partial I}{\partial z}\right)\bar{w}^n + \left(\frac{\partial I}{\partial t}\right)\right)}{\left(\left(\frac{\partial I}{\partial x}\right)^2 + \left(\frac{\partial I}{\partial y}\right)^2 + \left(\frac{\partial I}{\partial z}\right)^2 + \alpha^2\right)} \qquad \text{Eq. 17}$$

2.2.1.1.4 Limitations of the Horn-Schunck Method

Horn-Schunck optical flow is limited in its capacity to measure large displacements [18]. The algorithm often gets stuck in a local minimum [18]. Large displacements are considered displacements greater than 1 voxel [18]. This limitation is due to the constraint equation, Eq. 2, only being valid when the partial derivatives can be approximated correctly [9]. When the motion field is small enough, the gradient of the image is smooth, and the constraint equation can be applied [9]. The solution of a multiscale strategy to deal with large displacements in optical flow was proposed by Brox [9].
Another difficulty associated with Horn-Schunck optical flow is selecting the smoothing parameter, α. The smoothing parameter is a trade-off between satisfying the brightness intensity constraint and the smoothness of flow constraint [21]. Often, this parameter is set by hand, which limits the ability to generalize the method for unknown data. Ng and Solo proposed a data-driven method for choosing smoothing parameters in optical flow problems [21]. Their method used a stochastic model and an unbiased risk estimator [21].

2.2.1.2 Brox Optical Flow Method

In 2004, Brox built on the Horn-Schunck method to introduce a high accuracy optical flow estimation based on a theory for warping of 2D images [9]. In this model, the constraints of the grey value constancy assumption, and the smoothness assumption from the Horn-Schunck model are used [14]. These assumptions are detailed in Sections 2.2.1.1.1, and 2.2.1.1.2 respectively. A third assumption, the gradient constancy assumption is also implemented, detailed in Section 2.2.1.2.1 [14]. Another addition in the Brox method was the implementation of a multiscale approach [14]. The multiscale approach involves downsizing the images to evaluate the displacement field in a coarse to fine iterative structure [14]. This method is detailed in Section 2.2.1.2.5.

2.2.1.2.1 Gradient Constancy Assumption

The gradient constancy assumption, as shown in Eq. 18, imposes that the gradient of the image grey value is constant [14]. Adding this criterion allows for small variations in the grey value which is not accounted for by the grey value constancy assumption [9]. It is useful in tracking translatory motion [9]. In the equation $\nabla = \begin{bmatrix} \partial_x \\ \partial_y \end{bmatrix}$, the spatial gradient [14].

$$\nabla I(x, y, t) = \nabla I(x + u, y + v, t + 1) \qquad \text{Eq. 18}$$

2.2.1.2.2 2D Energy Functional

The energy functional can be derived by combining the 3 following assumptions: the brightness constancy assumption, the smoothness assumption, and the gradient constancy assumption [14]. The data term minimizes the error in the brightness intensity constancy, and gradient constancy as shown in Eq. 19 [14]. Here, γ is the weighting term between the two assumptions. To decrease the influence of outliers a concave function $\psi(s^2) = \sqrt{s^2 + \epsilon}$ is applied in Eq. 20 [14]. s^2 represents the integrand of either the data term or the smoothness term. ϵ was initialized as 0.001 by Brox [9]. As in the Horn-Schunck method, a smoothness term is shown in Eq. 21, and the total energy term to be minimized is shown in Eq. 22 [14].

$$E_{data}(u,v) = \int |I(x+u,y+v,t+1) - I(x,y,t)|^2 + \gamma|\nabla I(x+u,y+v,t+1) - \nabla I(x,y,t)|^2 dxdy \quad \text{Eq. 19}$$

$$E_{data}(u,v) = \int_{\Omega} \psi(|I(x+u,y+v,t+1) - I(x,y,t)|^2 + \gamma|\nabla I(x+u,y+v,t+1) - \nabla I(x,y,t)|^2) dxdy \quad \text{Eq. 20}$$

$$E_{smooth}(u,v) = \int \psi(|\nabla_3 u|^2 + |\nabla_3 v|^2) dxdy \quad \text{Eq. 21}$$

$$E(u,v) = E_{data}(u,v) + \alpha E_{smooth}(u,v) \quad \text{Eq. 22}$$

2.2.1.2.3 2D Euler-Lagrange Equations

The energy functional shown in Eq. 22 is nonlinear, and therefore difficult to minimize. The Euler Lagrange method for minimization is used [14]. The energy functional is used as the input function into the Euler-Lagrange equations, f =$E(u,v)$ [14]. The general expression of Euler-Lagrange equations is shown in Eq. 23 and Eq. 24 [14]. The abbreviations, Eq. 25 to Eq. 32, are used to improve readability, in the derived Euler-Lagrange equations; Eq. 33 and Eq. 34 [14]. Satisfying the constraints of the Euler-Lagrange equations can be used to determine the functions u, v that are the minimiser [9].

$$f_u - \frac{df_{u_x}}{dx} - \frac{df_{u_y}}{dy} = 0 \quad \text{Eq. 23}$$

$$f_v - \frac{df_{v_x}}{dx} - \frac{df_{v_y}}{dy} = 0 \quad \text{Eq. 24}$$

$$I_x = \partial_x I(x+u,y+v,t+1) \quad \text{Eq. 25}$$

$$I_y = \partial_y I(x+u,y+v,t+1) \quad \text{Eq. 26}$$

$$I_T = I(x+u,y+v,t+1) - I(x,y,t) \quad \text{Eq. 27}$$

$$I_{xx} = \partial_{xx} I(x+u,y+v,t+1) \quad \text{Eq. 28}$$

$$I_{xy} = \partial_{xy} I(x+u,y+v,t+1) \quad \text{Eq. 29}$$

$$I_{yy} = \partial_{yy} I(x+u,y+v,t+1) \quad \text{Eq. 30}$$

$$I_{xt} = \partial_x I(x+u,y+v,t+1) - \partial_x I(x,y,t) \quad \text{Eq. 31}$$

$$I_{yt} = \partial_y I(x+u, y+v, t+1) - \partial_y I(x, y, t) \qquad \text{Eq. 32}$$

$$\psi'\left({I_t}^2 + \gamma\left({I_{xt}}^2 + {I_{yt}}^2\right)\right) \cdot \left(I_x I_t + \gamma\left(I_{xx} I_{xt} + I_{xy} I_{yt}\right)\right) - \alpha\, div\, (\psi'(|\nabla_3 u|^2 + |\nabla_3 v|^2)\nabla_3 u) = 0 \qquad \text{Eq. 33}$$

$$\psi'\left({I_t}^2 + \gamma\left({I_{xt}}^2 + {I_{yt}}^2\right)\right) \cdot \left(I_y I_t + \gamma\left(I_{yy} I_{yt} + I_{xy} I_{xt}\right)\right) - \alpha\, div\, (\psi'(|\nabla_3 u|^2 + |\nabla_3 v|^2)\nabla_3 v) = 0 \qquad \text{Eq. 34}$$

2.2.1.2.4 3D Extension

The Brox method was extended to be used for 3D images by Chen [8]. The energy functional terms are shown in Eq. 35 and Eq. 36 [8]. The general expression for the Euler-Lagrange equations is shown Eq. 38, Eq. 39, and Eq. 40 [15]. The Euler-Lagrange equations are shown in Eq. 54, Eq. 55, and Eq. 56, using the abbreviated form shown in Eq. 41 to Eq. 53 [15].

$$E_{data}(u, v, w) = \int_\Omega \psi(|I(x+u, y+v, z+w, t+1) - I(x, y, z, t)|^2 + \gamma|\nabla I(x+u, y+v, z+w, t+1) - \nabla I(x, y, z, t)|^2)dxdydz \qquad \text{Eq. 35}$$

$$E_{smooth}(u, v, w) = \int \psi(|\nabla_3 u|^2 + |\nabla_3 v|^2 + |\nabla_3 w|^2)dxdydz \qquad \text{Eq. 36}$$

$$E(u, v) = E_{data}(u, v, w) + \alpha E_{smooth}(u, v, w) \qquad \text{Eq. 37}$$

$$f_u - \frac{df_{u_x}}{dx} - \frac{df_{u_y}}{dy} - \frac{df_{u_z}}{dz} = 0 \qquad \text{Eq. 38}$$

$$f_v - \frac{df_{v_x}}{dx} - \frac{df_{v_y}}{dy} - \frac{df_{v_z}}{dz} = 0 \qquad \text{Eq. 39}$$

$$f_w - \frac{df_{w_x}}{dx} - \frac{df_{w_y}}{dy} - \frac{df_{w_z}}{dz} = 0 \qquad \text{Eq. 40}$$

$$I_x = \partial_x I(x+u, y+v, z+w, t+1) \qquad \text{Eq. 41}$$

$$I_y = \partial_y I(x+u, y+v, z+w, t+1) \qquad \text{Eq. 42}$$

$$I_z = \partial_z I(x+u, y+v, z+w, t+1) \qquad \text{Eq. 43}$$

$$I_T = I(x+u, y+v, z+w, z+w, t+1) - I(x, y, z, t) \qquad \text{Eq. 44}$$

$$I_{xx} = \partial_{xx} I(x+u, y+v, z+w, t+1) \quad \text{Eq. 45}$$

$$I_{xy} = \partial_{xy} I(x+u, y+v, z+w, t+1) \quad \text{Eq. 46}$$

$$I_{xz} = \partial_{xz} I(x+u, y+v, z+w, t+1) \quad \text{Eq. 47}$$

$$I_{yy} = \partial_{yy} I(x+u, y+v, z+w, t+1) \quad \text{Eq. 48}$$

$$I_{yz} = \partial_{yz} I(x+u, y+v, z+w, t+1) \quad \text{Eq. 49}$$

$$I_{zz} = \partial_{zz} I(x+u, y+v, z+w, t+1) \quad \text{Eq. 50}$$

$$I_{xt} = \partial_{x} I(x+u, y+v, z+w, t+1) - \partial_x I(x, y, z, t) \quad \text{Eq. 51}$$

$$I_{yt} = \partial_{y} I(x+u, y+v, z+w, t+1) - \partial_y I(x, y, z, t) \quad \text{Eq. 52}$$

$$I_{zt} = \partial_{z} I(x+u, y+v, z+w, t+1) - \partial_z I(x, y, z, t) \quad \text{Eq. 53}$$

$$\psi'\left({I_t}^2 + \gamma\left({I_{xt}}^2 + {I_{yt}}^2 + {I_{zt}}^2\right)\right) \cdot \left(I_x I_t + \gamma\left(I_{xx} I_{xt} + I_{xy} I_{yt} + I_{xz} I_{zt}\right)\right) - \alpha \, div \left(\psi'(|\nabla_3 u|^2 + |\nabla_3 v|^2 + |\nabla_3 w|^2) \nabla_3 u\right) = 0 \quad \text{Eq. 54}$$

$$\psi'\left({I_t}^2 + \gamma\left({I_{xt}}^2 + {I_{yt}}^2 + {I_{zt}}^2\right)\right) \cdot \left(I_y I_t + \gamma\left(I_{yx} I_{xt} + I_{yy} I_{yt} + I_{yz} I_{zt}\right)\right) - \alpha \, div \left(\psi'(|\nabla_3 u|^2 + |\nabla_3 v|^2 + |\nabla_3 w|^2) \nabla_3 v\right) = 0 \quad \text{Eq. 55}$$

$$\psi'\left({I_t}^2 + \gamma\left({I_{xt}}^2 + {I_{yt}}^2 + {I_{zt}}^2\right)\right) \cdot \left(I_z I_t + \gamma\left(I_{zx} I_{xt} + I_{zy} I_{yt} + I_{zz} I_{zt}\right)\right) - \alpha \, div \left(\psi'(|\nabla_3 u|^2 + |\nabla_3 v|^2 + |\nabla_3 w|^2) \nabla_3 w\right) = 0 \quad \text{Eq. 56}$$

2.2.1.2.5 Multiscale Approach

Given the resolution levels using current medical imaging techniques, it is expected for images of an aortic valve over a cardiac cycle to have displacement magnitude of greater than one voxel over the time between two sequential images. Optical Flow algorithms, particularly the Horn-Schunck technique, have difficulty computing large displacements (by definition, large is considered as greater than one voxel). This happens because the algorithm frequently gets stuck finding a local minimum of an energy functional in the process of looking for a global minimum [9]. This can be counteracted by the use of a multiscale approach. Figure 6 shows the overview of the application of a pyramidal scheme for optical flow. It uses down sampling with coarse to fine optical flow estimation to refine the displacement field over a series of iterations [9]. Two original images are downsized *N* times by a scale factor of *S* [9]. This creates a sequence of images of decreasing size. As the images get smaller, each pixel represents a larger and blurrier area. Through the layers of increasingly blurred images, a smoothed sequence of the same image is created. Figure 7 shows an example of the smoothed sequence of images that make up the levels of the pyramid, 20 pyramid levels are shown with a scale factor of 0.95. The optical flow algorithm is used between the two smallest images, which have the largest area represented by each pixel, to determine a flow field, $(u,v)^N$ [9]. This run should produce the unique minimum of the smoothed problem, which ideally is similar to the global minimum of the original problem [8]. The flow field is then scaled up by $\frac{1}{s}$ and resized to the size of the image on the next level [9]. The flow field is then used to warp the second image of the next level in the pyramid. This acts as a course adjustment causing the image to appear to look closer to the first image of the next level. The optical flow displacement field is continually refined by this process throughout the levels of the pyramid. The coarse refinement of the displacement field becomes fine refinement of the displacement field as the algorithm moves through the levels of the pyramid, similar to the adjustment knobs on a microscope. The same principle can be applied in 3D images.

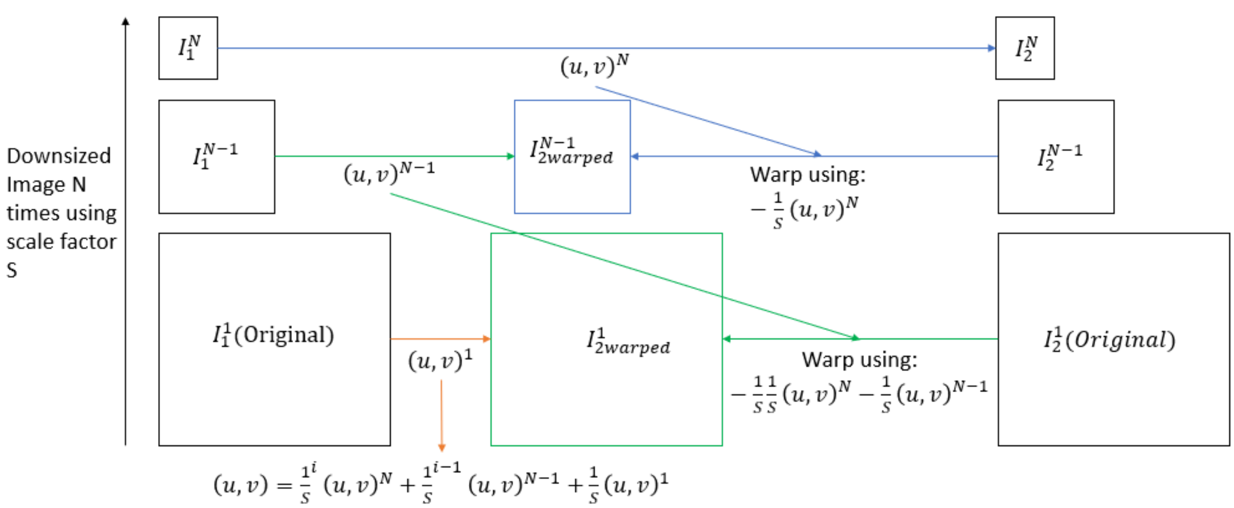


Figure 6: Pyramidal Structure See text for details.

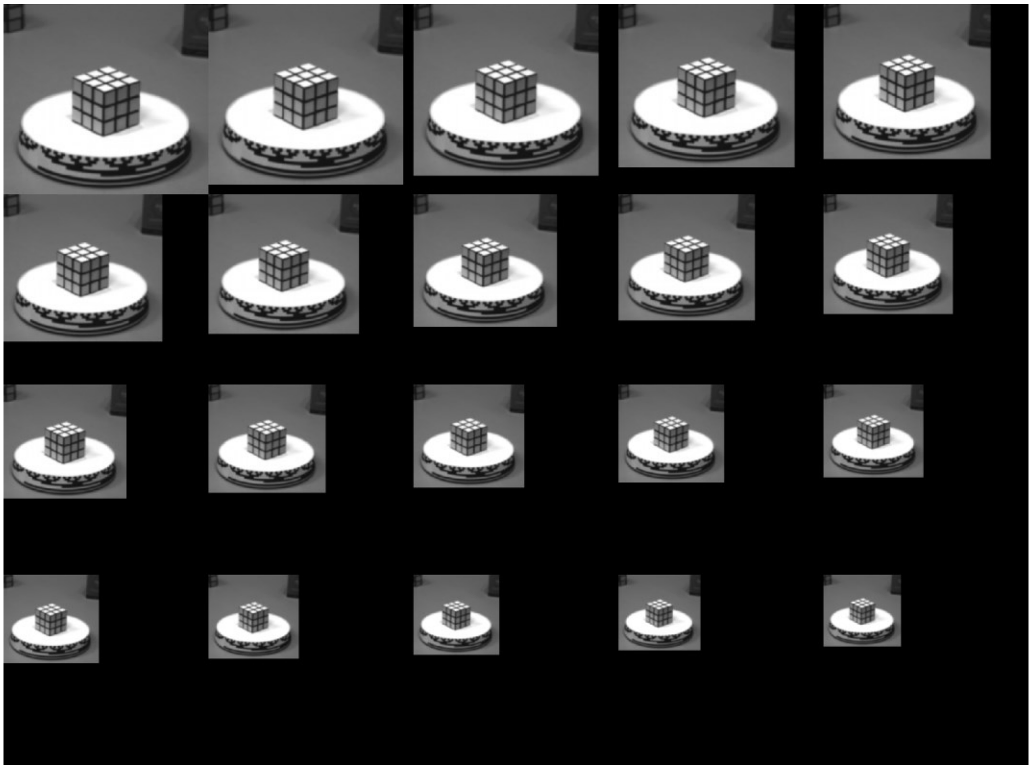


Figure 7: Example of smoothed sequence of images that make up pyramid levels (N=20, S=0.95). see text for details.

2.2.1.2.6 Numerical Implementation

Fixed point iteration is used in combination with the multiscale approach to deal with the nonlinearity of the argument ***d*=** (*u,v,w,*1) of the Euler-Lagrange equations Eq. 54 to Eq. 56 [8]. A graphical example of how the fixed point iteration method is used in 1D is shown in Figure 8. Two functions are graphed: y=cos(x), and y=x. The fixed point iteration method is used to find the intersection of the two functions, where x=cos(x). An initial guess of x=-1 is chosen, and the function cos(x) is evaluated using the initial guess, 0.54 in this case. The x-value of the other function, y=x, at y=0.54 is found and used as the x-value estimate for the next step. This process is repeated until the x-value converges within a given level of error. The same principle can be applied to more complicated functions that make up optical flow.

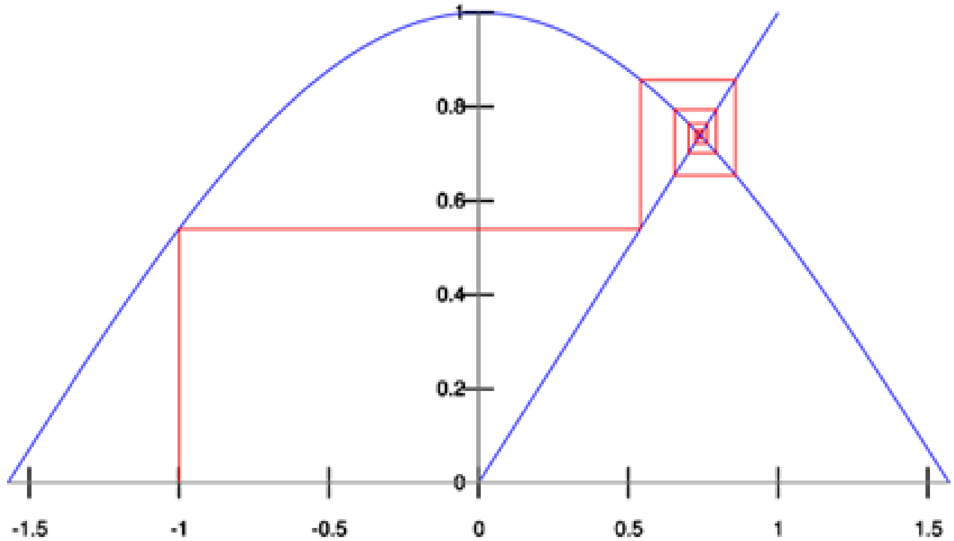


Figure 8: Example of fixed point iteration in 1D to find the intersection point between the functions y=cos(x) and y=x [22]

Starting with the coarsest pyramid level, the smallest image, fixed point iterations are implemented. The iteration is indexed by the variable k, given a solution $\boldsymbol{d}^k$ at level k, the solution for $\boldsymbol{d}^{k+1}$can be determined using Eq. 57, Eq. 58, and Eq. 59 [8]. Once the fixed point $\boldsymbol{d}^k$ is found the process is repeated for the next finest pyramid level using $\boldsymbol{d}^k$ as an initialization for the fixed point at the new level.

$$\begin{aligned}\psi'\left(\left(I_t^{k+1}\right)^2+\gamma\left(\left(I_{xt}^{k+1}\right)^2+\left(I_{yt}^{k+1}\right)^2+\left(I_{zt}^{k+1}\right)^2\right)\right) \\ \cdot\left(I_x^k I_t^{k+1}+\gamma\left(I_{xx}^k I_{xt}^{k+1}+I_{xy}^k I_{yt}^{k+1}+I_{xz}^k I_{zt}^{k+1}\right)\right) \\ -\alpha\, div\left(\psi'\left(\left|\nabla_3 u^{k+1}\right|^2+\left|\nabla_3 v^{k+1}\right|^2+\left|\nabla_3 w^{k+1}\right|^2\right)\nabla_3 u^{k+1}\right)=0\end{aligned} \quad \text{Eq. 57}$$

$$\begin{aligned}\psi'\left(\left(I_t^{k+1}\right)^2+\gamma\left(\left(I_{xt}^{k+1}\right)^2+\left(I_{yt}^{k+1}\right)^2+\left(I_{zt}^{k+1}\right)^2\right)\right) \\ \cdot\left(I_y^k I_t^{k+1}+\gamma\left(I_{yx}^k I_{xt}^{k+1}+I_{yy}^k I_{yt}^{k+1}+I_{yz}^k I_{zt}^{k+1}\right)\right) \\ -\alpha\, div\left(\psi'\left(\left|\nabla_3 u^{k+1}\right|^2+\left|\nabla_3 v^{k+1}\right|^2+\left|\nabla_3 w^{k+1}\right|^2\right)\nabla_3 v^{k+1}\right)=0\end{aligned} \quad \text{Eq. 58}$$

$$\psi'\left(\left(I_t^{k+1}\right)^2+\gamma\left(\left(I_{xt}^{k+1}\right)^2+\left(I_{yt}^{k+1}\right)^2+\left(I_{zt}^{k+1}\right)^2\right)\right)$$
$$\cdot\left(I_z^k I_t^{k+1}+\gamma\left(I_{zx}^k I_{xt}^{k+1}+I_{zy}^k I_{yt}^{k+1}+I_{zz}^k I_{zt}^{k+1}\right)\right)$$
$$-\alpha\ div\ \left(\psi'\left(\left|\nabla_3 u^{k+1}\right|^2+\left|\nabla_3 v^{k+1}\right|^2+\left|\nabla_3 w^{k+1}\right|^2\right)\nabla_3 w^{k+1}\right)=0 \quad \text{Eq. 59}$$

At this point the smoothness term is fully implicit, and the data term is semi-implicit [8]. These equations are still nonlinear because of the variable I^{k+1}, and nonlinear function ψ'. Taylor series expansion is used to deal with the nonlinearity of I^{k+1} by converting variables in the form I^{k+1} to the form of I^k as shown in Eq. 60 to Eq. 63 [8].

$$I_t^{k+1}\approx I_t^k+I_x^k du^k+I_y^k dv^k+I_z^k dw^k \quad \text{Eq. 60}$$

$$I_{xt}^{k+1}\approx I_{xt}^k+I_{xx}^k du^k+I_{xy}^k dv^k+I_{xz}^k dw^k \quad \text{Eq. 61}$$

$$I_{yt}^{k+1}\approx I_{yt}^k+I_{yx}^k du^k+I_{yy}^k dv^k+I_{yz}^k dw^k \quad \text{Eq. 62}$$

$$I_{zt}^{k+1}\approx I_{zt}^k+I_{zx}^k du^k+I_{zy}^k dv^k+I_{zz}^k dw^k \quad \text{Eq. 63}$$

In these equations, $u^{k+1}=u^k+du^k$, $v^{k+1}=v^k+dv^k$, and $w^{k+1}=w^k+dw^k$ [8]. The unknowns at the level k+1 (u^{k+1}, v^{k+1}, and w^{k+1}) are now in terms of known solutions at level k (u^k, v^k, and w^k) and the unknown changes at levels k (du^k, dv^k, and dw^k) [8]. Eq. 64 and Eq. 65 can be written for better readability. The solution at level *k* is shown in Eq. 66 to Eq. 68 [8].

$$
\begin{aligned}
(\psi')^k_{Data} = \psi' \Big(& \left(I^k_t + I^k_x du^k + I^k_y dv^k + I^k_z dw^k\right)^2 \\
& + \gamma \Big(\left(I^k_{xt} + I^k_{xx} du^k + I^k_{xy} dv^k + I^k_{xz} dw^k\right)^2 \\
& + \left(I^k_{yt} + I^k_{yx} du^k + I^k_{yy} dv^k + I^a_{yz} dw^k\right)^2 \\
& + \left(I^k_{zt} + I^k_{zx} du^k + I^k_{zy} dv^k + I^k_{zz} dw^k\right)^2 \Big) \Big)
\end{aligned}
\qquad \text{Eq. 64}
$$

$$
(\psi')^k_{Smooth} = \psi' \left(\left|\nabla_3\left(u^k + du^k\right)\right|^2 + \left|\nabla_3\left(v^k + dv^k\right)\right|^2 + \left|\nabla_3\left(w^k + dw^k\right)\right|^2 \right) \qquad \text{Eq. 65}
$$

$$
\begin{aligned}
(\psi')^k_{Data} \cdot \Big(& I^k_x\left(I^k_t + I^k_x du^k + I^k_y dv^k + I^k_z dw^k\right)\Big) + \gamma (\psi')^k_{Data} \\
& \cdot \Big(I^k_{xx}\left(I^k_{xt} + I^k_{xx} du^k + I^k_{xy} dv^k + I^k_{xz} dw^k\right) \\
& + I^k_{xy}\left(I^k_{yt} + I^k_{xy} du^k + I^k_{yy} dv^k + I^k_{yz} dw^k\right) \\
& + I^k_{xz}\left(I^k_{zt} + I^k_{xz} du^k + I^k_{zy} dv^k + I^k_{zz} dw^k\right)\Big) \\
& - \alpha\, div\, \left((\psi')^k_{Smooth} \nabla_3\left(u^k + du^k\right)\right) = 0
\end{aligned}
\qquad \text{Eq. 66}
$$

$$
\begin{aligned}
(\psi')^k_{Data} \cdot \Big(& I^k_y\left(I^k_t + I^k_x du^k + I^k_y dv^k + I^k_z dw^k\right)\Big) + \gamma (\psi')^k_{Data} \\
& \cdot \Big(I^k_{yx}\left(I^k_{xt} + I^k_{xx} du^k + I^k_{xy} dv^k + I^k_{xz} dw^k\right) \\
& + I^k_{yy}\left(I^k_{yt} + I^k_{xy} du^k + I^k_{yy} dv^k + I^k_{yz} dw^k\right) \\
& + I^k_{yz}\left(I^k_{zt} + I^k_{xz} du^k + I^k_{zy} dv^k + I^k_{zz} dw^k\right)\Big) \\
& - \alpha\, div\, \left((\psi')^k_{Smooth} \nabla_3\left(v^k + dv^k\right)\right) = 0
\end{aligned}
\qquad \text{Eq. 67}
$$

$$
\begin{aligned}
(\psi')^k_{Data} \cdot \Big(& I^k_z\left(I^k_t + I^k_x du^k + I^k_y dv^k + I^k_z dw^k\right)\Big) + \gamma (\psi')^k_{Data} \\
& \cdot \Big(I^k_{zx}\left(I^k_{xt} + I^k_{xx} du^k + I^k_{xy} dv^k + I^k_{xz} dw^k\right) \\
& + I^k_{zy}\left(I^k_{yt} + I^k_{xy} du^k + I^k_{yy} dv^k + I^k_{yz} dw^k\right) \\
& + I^k_{zz}\left(I^k_{zt} + I^k_{xz} du^k + I^k_{zy} dv^k + I^k_{zz} dw^k\right)\Big) \\
& - \alpha\, div\, \left((\psi')^k_{Smooth} \nabla_3\left(w^k + dw^k\right)\right) = 0
\end{aligned}
\qquad \text{Eq. 68}
$$

The final nonlinearity to address is the function ψ'. The function, $\psi(s^2) = \sqrt{s^2 + \epsilon}$, is intentionally convex such that a unique minimum solution exists [8]. A second fixed point iteration is used to deal with the non-linearity. The second iteration is the inner iteration and is indexed by the variable *l.* This iteration loop runs within the outer iteration loop *k.* The resulting linear systems of equations is shown in Eq. 69 to Eq. 71 [8].

$$
\begin{aligned}
(\psi')^{k,l}_{Data} \cdot \Big(I^k_x\big(I^k_t + I^k_x du^{k,l+1} + I^k_y dv^{k,l+1} + I^k_z dw^{k,l+1}\big)\Big) + \gamma(\psi')^{k,l}_{Data} \\
\cdot \Big(I^k_{xx}\big(I^k_{xt} + I^k_{xx} du^{k,l+1} + I^k_{xy} dv^{k,l+1} + I^k_{xz} dw^{k,l+1}\big) \\
+ I^k_{xy}\big(I^k_{yt} + I^k_{xy} du^{k,l+1} + I^k_{yy} dv^{k,l+1} + I^k_{yz} dw^{k,l+1}\big) \\
+ I^k_{xz}\big(I^k_{zt} + I^k_{xz} du^{k,l+1} + I^k_{zy} dv^{k,l+1} + I^k_{zz} dw^{k,l+1}\big)\Big) \\
- \alpha\ div\ \Big((\psi')^{k,l}_{Smooth} \nabla_3\big(u^k + du^{k,l+1}\big)\Big) = 0
\end{aligned}
\qquad \text{Eq. 69}
$$

$$
\begin{aligned}
(\psi')^{k,l}_{Data} \cdot \Big(I^k_y\big(I^k_t + I^k_x du^{k,l+1} + I^k_y dv^{k,l+1} + I^k_z dw^{k,l+1}\big)\Big) + \gamma(\psi')^{k,l}_{Data} \\
\cdot \Big(I^k_{yx}\big(I^k_{xt} + I^k_{xx} du^{k,l+1} + I^k_{xy} dv^{k,l+1} + I^k_{xz} dw^{k,l+1}\big) \\
+ I^k_{yy}\big(I^k_{yt} + I^k_{xy} du^{k,l+1} + I^k_{yy} dv^{k,l+1} + I^k_{yz} dw^{k,l+1}\big) \\
+ I^k_{yz}\big(I^k_{zt} + I^k_{xz} du^{k,l+1} + I^k_{zy} dv^{k,l+1} + I^k_{zz} dw^{k,l+1}\big)\Big) \\
- \alpha\ div\ \Big((\psi')^{k,l}_{Smooth} \nabla_3\big(v^k + dv^{k,l+1}\big)\Big) = 0
\end{aligned}
\qquad \text{Eq. 70}
$$

$$
\begin{aligned}
(\psi')^{k,l}_{Data} \cdot \Big(I^k_z\big(I^k_t + I^k_x du^{k,l+1} + I^k_y dv^{k,l+1} + I^k_z dw^{k,l+1}\big)\Big) + \gamma(\psi')^{k,l}_{Data} \\
\cdot \Big(I^k_{zx}\big(I^k_{xt} + I^k_{xx} du^{k,l+1} + I^k_{xy} dv^{k,l+1} + I^k_{xz} dw^{k,l+1}\big) \\
+ I^k_{zy}\big(I^k_{yt} + I^k_{xy} du^{k,l+1} + I^k_{yy} dv^{k,l+1} + I^k_{yz} dw^{k,l+1}\big) \\
+ I^k_{zz}\big(I^k_{zt} + I^k_{xz} du^{k,l+1} + I^k_{zy} dv^{k,l+1} + I^k_{zz} dw^{k,l+1}\big)\Big) \\
- \alpha\ div\ \Big((\psi')^{k,l}_{Smooth} \nabla_3\big(w^k + dw^{k,l+1}\big)\Big) = 0
\end{aligned}
\qquad \text{Eq. 71}
$$

Brox et al. used SOR iterations to solve the linear system of equations [9]. Faisal and Barron suggested using Cramer's rule to solve the linear systems of equations as they were unable to implement the SOR iteration method [23].

2.2.1.2.7 Solving the linear system

The general expression of a system of three linear equations in matrix form is shown in Eq. 72 [8]. The solution of any given linear expression, *X, Y and Z,* is shown in Eq. 73 to Eq. 75 [8].

$$
\begin{bmatrix} A & B & C \\ D & E & F \\ G & G & I \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} J \\ K \\ L \end{bmatrix} \qquad \text{Eq. 72}
$$

$$X = \frac{J(EI - HF) - K(BI - HC) + L(BF - CE)}{A(EI - HF) - D(BI - HC) + G(BF - CE)} \quad \text{Eq. 73}$$

$$Y = \frac{J(DI - GF) - K(AI - GC) + L(AF - CD)}{B(DI - GF) - E(AI - GC) + H(AF - CD)} \quad \text{Eq. 74}$$

$$Z = \frac{J(DH - EG) - K(AH - BG) + L(AE - DB)}{C(DH - EG) - F(AH - BG) + I(AE - DB)} \quad \text{Eq. 75}$$

The system of three linear equations given by Eq. 69 to Eq. 71 can be converted into the form as shown by Eq. 76 to Eq. 78 where the variables *A* through *L* are shown in Eq. 79 to Eq. 90 [8].

$$A\left(du^{k,l+1}\right) + B\left(dv^{k,l+1}\right) + C\left(dw^{k,l+1}\right) = J \quad \text{Eq. 76}$$

$$D\left(du^{k,l+1}\right) + E\left(dv^{k,l+1}\right) + F\left(dw^{k,l+1}\right) = K \quad \text{Eq. 77}$$

$$G\left(du^{k,l+1}\right) + H\left(dv^{k,l+1}\right) + I\left(dw^{k,l+1}\right) = L \quad \text{Eq. 78}$$

$$A = (\psi')_{Data}^{k,l+1}\left(I_x^k I_x^k + \gamma\left(I_{xx}^k I_{xx}^k + I_{xy}^k I_{xy}^k + I_{xz}^k I_{xz}^k\right)\right) \quad \text{Eq. 79}$$

$$B = (\psi')_{Data}^{k,l+1}\left(I_x^k I_y^k + \gamma\left(I_{xx}^k I_{xy}^k + I_{xy}^k I_{yy}^k + I_{xz}^k I_{yz}^k\right)\right) \quad \text{Eq. 80}$$

$$C = (\psi')_{Data}^{k,l+1}\left(I_x^k I_z^k + \gamma\left(I_{xx}^k I_{xz}^k + I_{xy}^k I_{yz}^k + I_{xz}^k I_{zz}^k\right)\right) \quad \text{Eq. 81}$$

$$D = (\psi')_{Data}^{k,l+1}\left(I_y^k I_x^k + \gamma\left(I_{yx}^k I_{xx}^k + I_{yy}^k I_{xy}^k + I_{yz}^k I_{xz}^k\right)\right) \quad \text{Eq. 82}$$

$$E = (\psi')_{Data}^{k,l+1}\left(I_y^k I_y^k + \gamma\left(I_{yx}^k I_{xy}^k + I_{yy}^k I_{yy}^k + I_{yz}^k I_{yz}^k\right)\right) \quad \text{Eq. 83}$$

$$F = (\psi')_{Data}^{k,l+1}\left(I_y^k I_z^k + \gamma\left(I_{yx}^k I_{xz}^k + I_{yy}^k I_{yz}^k + I_{yz}^k I_{zz}^k\right)\right) \quad \text{Eq. 84}$$

$$G = (\psi')_{Data}^{k,l+1}\left(I_z^k I_x^k + \gamma\left(I_{zx}^k I_{xx}^k + I_{yz}^k I_{xy}^k + I_{zz}^k I_{xz}^k\right)\right) \quad \text{Eq. 85}$$

$$H = (\psi')_{Data}^{k,l+1}\left(I_z^k I_y^k + \gamma\left(I_{zx}^k I_{xy}^k + I_{yz}^k I_{yy}^k + I_{zz}^k I_{yz}^k\right)\right) \quad \text{Eq. 86}$$

$$I = (\psi')_{Data}^{k,l+1}\left(I_z^k I_z^k + \gamma\left(I_{zx}^k I_{xz}^k + I_{yz}^k I_{yz}^k + I_{zz}^k I_{zz}^k\right)\right) \quad \text{Eq. 87}$$

$$\begin{aligned} J = \alpha\, (\psi')_{Smooth}^{k,l}\, \boldsymbol{Div}\left(\nabla\left(u^k + du^{k,l+1}\right)\right) \\ - (\psi')_{Data}^{k,l}\left(I_x^k I_t^k + \gamma\left(I_{xx}^k I_{xt}^k + I_{xy}^k I_{yt}^k + I_{xz}^k I_{zt}^k\right)\right) \end{aligned} \quad \text{Eq. 88}$$

$$K = \alpha\,(\psi')^{k,l}_{Smooth}\,\boldsymbol{Div}\left(\nabla\left(v^k + dv^{k,l+1}\right)\right) \\ -\ (\psi')^{k,l}_{Data}\left(I^k_y I^k_t + \gamma\left(I^k_{yx}I^k_{xt} + I^k_{yy}I^k_{yt} + I^k_{yz}I^k_{zt}\right)\right) \qquad \text{Eq. 89}$$

$$L = \alpha\,(\psi')^{k,l}_{Smooth}\boldsymbol{Div}\left(\nabla\left(w^k + dw^{k,l+1}\right)\right) \\ -\ (\psi')^{k,l}_{Data}\left(I^k_z I^k_t + \gamma\left(I^k_{zx}I^k_{xt} + I^k_{zy}I^k_{yt} + I^k_{zz}I^k_{zt}\right)\right) \qquad \text{Eq. 90}$$

Where;

$$\boldsymbol{Div}\left(\nabla\left(u^k + du^{k,l+1}\right)\right) = \left(u^k + du^{k,l+1}\right)_{xx} + \left(u^k + du^{k,l+1}\right)_{yy} + \left(u^k + du^{k,l+1}\right)_{zz} \\ = \left(u^k_{xx} + u^k_{yy} + u^k_{zz}\right) + \left(du^{k,l+1}_{xx} + du^{k,l+1}_{yy} + du^{k,l+1}_{zz}\right) \qquad \text{Eq. 91}$$

$$\boldsymbol{Div}\left(\nabla\left(v^k + dv^{k,l+1}\right)\right) = \left(v^k + dv^{k,l+1}\right)_{xx} + \left(v^k + dv^{k,l+1}\right)_{yy} + \left(v^k + dv^{k,l+1}\right)_{zz} \\ = \left(v^k_{xx} + v^k_{yy} + v^k_{zz}\right) + \left(dv^{k,l+1}_{xx} + dv^{k,l+1}_{yy} + dv^{k,l+1}_{zz}\right) \qquad \text{Eq. 92}$$

$$\boldsymbol{Div}\left(\nabla\left(w^k + dw^{k,l+1}\right)\right) = \left(w^k + dw^{k,l+1}\right)_{xx} + \left(w^k + dw^{k,l+1}\right)_{yy} + \left(w^k + dw^{k,l+1}\right)_{zz} \\ = \left(w^k_{xx} + w^k_{yy} + w^k_{zz}\right) + \left(dw^{k,l+1}_{xx} + dw^{k,l+1}_{yy} + dw^{k,l+1}_{zz}\right) \qquad \text{Eq. 93}$$

The approximation $X_{xx} + X_{yy} + X_{zz} \approx \bar{X} - X$ is used to simplify *J, K*, and *L* [23]. The same approximation of the Laplacian was also implemented by Horn-Schunck in their work [8], [18]. This results in the expressions for J, K, and L shown in Eq. 94 to Eq. 99 [8].

$$J = \alpha\,(\psi')^{k,l}_{Smooth}\,\left(\left(\overline{u^k} - u^k\right) + \left(\overline{du^{k,l+1}} - du^{k,l+1}\right)\right) - j \qquad \text{Eq. 94}$$

$$j = (\psi')^{k,l}_{Data}\left(I^k_x I^k_t + \gamma\left(I^k_{xx}I^k_{xt} + I^k_{xy}I^k_{yt} + I^k_{xz}I^k_{zt}\right)\right) \qquad \text{Eq. 95}$$

$$K = \alpha\,(\psi')^{k,l}_{Smooth}\,\left(\left(\overline{v^k} - v^k\right) + \left(\overline{dv^{k,l+1}} - dv^{k,l+1}\right)\right) - k \qquad \text{Eq. 96}$$

$$k = (\psi')^{k,l}_{Data}\left(I^k_y I^k_t + \gamma\left(I^k_{xy}I^k_{xt} + I^k_{yy}I^k_{yt} + I^k_{yz}I^k_{zt}\right)\right) \qquad \text{Eq. 97}$$

$$L = \alpha\,(\psi')^{k,l}_{Smooth}\,\left(\left(\overline{w^k} - w^k\right) + \left(\overline{dw^{k,l+1}} - dw^{k,l+1}\right)\right) - l \qquad \text{Eq. 98}$$

$$l = (\psi')^{k,l}_{Data}\left(I^k_z I^k_t + \gamma\left(I^k_{xz}I^k_{xt} + I^k_{yz}I^k_{yt} + I^k_{zz}I^k_{zt}\right)\right) \qquad \text{Eq. 99}$$

The entire expression in matrix form is shown in Eq. 100.

$$\begin{bmatrix} \left(A+\alpha\,(\psi')^{k,l}_{Smooth}\right) & B & C \\ D & \left(E+\alpha\,(\psi')^{k,l}_{Smooth}\right) & F \\ G & H & \left(I+\alpha\,(\psi')^{k,l}_{Smooth}\right) \end{bmatrix} \begin{bmatrix} du^{k,l+1} \\ dv^{k,l+1} \\ dw^{k,l+1} \end{bmatrix} = \begin{bmatrix} \alpha\,(\psi')^{k,l}_{Smooth}\left(\left(\overline{u^k}+\overline{du^{k,l+1}}-u^k\right)\right)-j \\ \alpha\,(\psi')^{k,l}_{Smooth}\left(\left(\overline{v^k}+\overline{dv^{k,l+1}}-v^k\right)\right)-k \\ \alpha\,(\psi')^{k,l}_{Smooth}\left(\left(\overline{w^k}+\overline{dw^{k,l+1}}-w^k\right)\right)-l \end{bmatrix} \quad \text{Eq. 100}$$

Cramer's rule is used to obtain the expressions for $du^{k,l+1}$, $dv^{k,l+1}$, $dw^{k,l+1}$ as shown in Eq. 101 to Eq. 103 using the determinate shown in Eq. 104 [8].

$$du^{k,l+1} = \frac{1}{det}\begin{bmatrix} \alpha\,(\psi')^{k,l}_{Smooth}\left(\left(\overline{u^k}+\overline{du^{k,l+1}}-u^k\right)\right)-j & B & C \\ \alpha\,(\psi')^{k,l}_{Smooth}\left(\left(\overline{v^k}+\overline{dv^{k,l+1}}-v^k\right)\right)-k & E & F \\ \alpha\,(\psi')^{k,l}_{Smooth}\left(\left(\overline{w^k}+\overline{dw^{k,l+1}}-w^k\right)\right)-l & H & I \end{bmatrix} \quad \text{Eq. 101}$$

$$dv^{k,l+1} = \frac{1}{det}\begin{bmatrix} A & \alpha\,(\psi')^{k,l}_{Smooth}\left(\left(\overline{u^k}+\overline{du^{k,l+1}}-u^k\right)\right)-j & C \\ D & \alpha\,(\psi')^{k,l}_{Smooth}\left(\left(\overline{v^k}+\overline{dv^{k,l+1}}-v^k\right)\right)-k & F \\ G & \alpha(\psi')^{k,l}_{Smooth}\left(\left(\overline{w^k}+\overline{dw^{k,l+1}}-w^k\right)\right)-l & I \end{bmatrix} \quad \text{Eq. 102}$$

$$dw^{k,l+1} = \frac{1}{det}\begin{bmatrix} A & B & \alpha\,(\psi')^{k,l}_{Smooth}\left(\left(\overline{u^k}+\overline{du^{k,l+1}}-u^k\right)\right)-j \\ D & E & \alpha\,(\psi')^{k,l}_{Smooth}\left(\left(\overline{v^k}+\overline{dv^{k,l+1}}-v^k\right)\right)-k \\ G & H & \alpha(\psi')^{k,l}_{Smooth}\left(\left(\overline{w^k}+\overline{dw^{k,l+1}}-w^k\right)\right)-l \end{bmatrix} \quad \text{Eq. 103}$$

Where;

$$\begin{aligned} det = {} & \left(A+\alpha\,(\psi')^{k,l}_{Smooth}\right)\left(EI+E\alpha\,(\psi')^{k,l}_{Smooth}+I\alpha\,(\psi')^{k,l}_{Smooth}+\alpha^2\left((\psi')^{k,l}_{Smooth}\right)^2 \right. \\ & \left. -HF\right)-D\left(BI+B\alpha\,(\psi')^{k,l}_{Smooth}-HC\right) \\ & +G\left(BF-CE-C\alpha\,(\psi')^{k,l}_{Smooth}\right) \end{aligned} \quad \text{Eq. 104}$$

Finally, using cofactor expansion, the simplified expressions for $du^{k,l+1}$, $dv^{k,l+1}$, $dw^{k,l+1}$ are shown in Eq. 105 to Eq. 107 [8].

$$du^{k,l+1} = \frac{1}{det}\Big(\alpha\,(\psi')^{k,l}_{Smooth}\,\Big(\big(\overline{u^k}+\overline{du^{k,l+1}}-u^k\big)\Big) - j\Big)\Big(\big(E+\alpha\,(\psi')^{k,l}_{Smooth}\big)\big(I+\alpha\,(\psi')^{k,l}_{Smooth}\big) - HF\Big) - \frac{1}{det}\Big(\alpha\,(\psi')^{k,l}_{Smooth}\,\Big(\big(\overline{v^k}+\overline{dv^{k,l+1}}-v^k\big)\Big) - k\Big)\Big(B\big(I+\alpha\,(\psi')^{k,l}_{Smooth}\big) - CH\Big) + \frac{1}{det}\Big(\alpha(\psi')^{k,l}_{Smooth}\,\Big(\big(\overline{w^k}+\overline{dw^{k,l+1}}-w^k\big)\Big) - l\Big)\Big(BF - C\big(E+\alpha\,(\psi')^{k,l}_{Smooth}\big)\Big) \qquad \text{Eq. 105}$$

$$dv^{k,l+1} = \frac{1}{det}\Big(\alpha\,(\psi')^{k,l}_{Smooth}\,\Big(\big(\overline{u^k}+\overline{du^{k,l+1}}-u^k\big)\Big) - j\Big)\Big(D\big(I+\alpha\,(\psi')^{k,l}_{Smooth}\big) - FG\Big) - \frac{1}{det}\Big(\alpha\,(\psi')^{k,l}_{Smooth}\,\Big(\big(\overline{v^k}+\overline{dv^{k,l+1}}-v^k\big)\Big) - k\Big)\Big(\big(A+\alpha\,(\psi')^{k,l}_{Smooth}\big)\big(I+\alpha\,(\psi')^{k,l}_{Smooth}\big) - CG\Big) + \frac{1}{det}\Big(\alpha(\psi')^{k,l}_{Smooth}\,\Big(\big(\overline{w^k}+\overline{dw^{k,l+1}}-w^k\big)\Big) - l\Big)\Big(F\big(A+\alpha\,(\psi')^{k,l}_{Smooth}\big) - CD\Big) \qquad \text{Eq. 106}$$

$$dw^{k,l+1} = \frac{1}{det}\Big(\alpha\,(\psi')^{k,l}_{Smooth}\,\Big(\big(\overline{u^k}+\overline{du^{k,l+1}}-u^k\big)\Big) - j\Big)\Big(DH - G\big(E+\alpha\,(\psi')^{k,l}_{Smooth}\big)\Big) - \frac{1}{det}\Big(\alpha\,(\psi')^{k,l}_{Smooth}\,\Big(\big(\overline{v^k}+\overline{dv^{k,l+1}}-v^k\big)\Big) - k\Big)\Big(\big(A+\alpha\,(\psi')^{k,l}_{Smooth}\big)H - BG\Big) + \frac{1}{det}\Big(\alpha(\psi')^{k,l}_{Smooth}\,\Big(\big(\overline{w^k}+\overline{dw^{k,l+1}}-w^k\big)\Big) - l\Big)\Big(\big(A+\alpha\,(\psi')^{k,l}_{Smooth}\big)\big(E+\alpha\,(\psi')^{k,l}_{Smooth}\big) - BD\Big) \qquad \text{Eq. 107}$$

The unknowns, $du^{k,l+1}$, $dv^{k,l+1}$, $dw^{k,l+1}$, can be found through an iterative process. All three unknowns are initialized to zero, then re-evaluated using the previous guess to compute the values of; $\overline{du^{k,l+1}}$, $\overline{dv^{k,l+1}}$, and $\overline{du^{k,l+1}}$ using the average of the $7 \times 7 \times 7$ neighbouring pixels [8].

2.2.1.3 Error Measurement of Optical Flow

Different metrics have been established in order to determine the effectiveness of an optical flow algorithm. Three of the more common methods are as follows: endpoint error, percent error of the magnitude of displacement, and an angular measure of error [21]. Endpoint error considers the distance between the endpoints of the calculated nodal displacement vectors, subscript c, and expected nodal displacement vectors or ground truth, subscript e [21]. The endpoint error is shown in Eq. 108. The percent error method calculates the difference between the magnitude of displacement between the ground truth displacement vector, subscript *e* for expected, and the calculated displacement vector, subscript *c*, obtained through the optical flow algorithm, as shown in Eq. 109 [21]. The angular error is shown in Eq. 110, where $\vec{\mathbf{v}}_c$ is the calculated displacement, computed using the optical flow algorithm, and $\vec{\mathbf{v}}_e$ is the expected displacement, or ground truth [24]. Both displacement terms are calculated using Eq. 111. This method of measuring error does not account for the magnitude of the vectors which is useful in problems that are only concerned with the trajectory of an object, not its speed [21].

$$Endpoint\ Error = \sqrt{(u_c - u_e)^2 + (v_c - v_e)^2} \qquad \text{Eq. 108}$$

$$Percent\ Error = \frac{\left|\sqrt{{u_c}^2 + {v_c}^2} - \sqrt{{u_e}^2 + {v_e}^2}\right|}{\sqrt{{u_e}^2 + {v_e}^2}} \qquad \text{Eq. 109}$$

$$\psi_E = arccos(\overrightarrow{\mathbf{v}_c} \cdot \overrightarrow{\mathbf{v}_e}) \qquad \text{Eq. 110}$$

$$\vec{\mathbf{v}} = \frac{1}{\sqrt{u^2 + v^2 + 1}}(u, v, 1)^T \qquad \text{Eq. 111}$$

2.2.1.4 Use of Optical Flow with Finite Element Analysis

Satriano et al. proposed using optical flow techniques to develop aortic wall strain maps in order to inform surgical decisions for abdominal aortic aneurysms (AAA) [4]. Their work served as an inspiration for the approach suggested herein. Using patient specific indicators of AAA could improve a doctor's ability to treat the patient compared to the alternative criteria of maximum diameter and growth rate [4]. In this study, cine-MRI scans taken with a General Electric 1.5 T imager were analyzed. Images at 20 points throughout the cardiac cycle were used [4]. These images were 256x256 pixels and 6-mm slice thickness, resulting in 1.4 mm in plane resolution, and 6 mm between plane resolution [4]. The images were segmented and resampled to obtain a 2 mm isotropic resolution which was exported as an STL triangular mesh [4]. Using the 3D extension of the Horn-Schunck optical flow technique, the displacement field of image n to image n+1 was computed [4]. Green-Lagrange strain was computed from the displacement field [4]. To validate the approach, a finite element analysis was completed on an aneurysm to determine a strain field; this was used as a ground truth [4]. The mesh from this analysis was then converted into synthetic DICOM image stacks [4]. These images were of varying resolution, 0.5 mm, 1 mm, and 1.5 mm isotropic resolution, and 1.5 mm in plane resolution with 6 mm between plane resolution [4]. The optical flow procedure was completed on the synthetic DICOM images and the resulting strain maps were compared to the ground truth from the finite element model results [4]. Spearman correlation coefficients of 0.81, 0.82, 0.84, and 0.76 were found for the varying resolutions respectively [4]. All of these resulted in a p- value less than 0.01 indicating statistical significance [4]. The effect of noise was also evaluated. Rician noise with a signal to noise ratio 9 (SNR) between 14 and 40 db was added to the 1.5 mm isometric synthetic images and assessed using a Spearman correlation coefficient [4]. It was found that when the SNR were 14.26 dB, 18.16 dB, 23.12 dB, 30.18 dB, and 42.30 dB, the correlation coefficients to the noise free model were 0.79, 0.90, 0.90, 0.93, and 0.95 respectively [4]. It was concluded that although there were challenges that needed to be addressed with respect to using real images, the use of optical flow in analysing wall deformation could be useful in analysing patient-specific AAAs [4].

2.2.2 Demons

Thirion introduced the idea of using diffusion models for image-to-image matching based on the concept of Maxwell's demons [6]. Maxwell's demons were originally a thought experiment in the 19th century as a solution to the Gibb's paradox in thermodynamics [6]. Maxwell's theory involved a gas mixture with two types of particles, A and B, which are separated by a semipermeable membrane as shown in Figure 9 [6]. On the membrane are a set of 'demons' who are able to distinguish particle A from particle B, allowing only type A particles to move to side A, and only type B particles to move to side B [6]. The result of the sorting is a decrease in entropy, which is contrary to the second law of thermodynamics [6]. The paradox is explained by the entropy generated by the demons to recognize the two types of particles resulting in a net increase in the total entropy of the system [6].

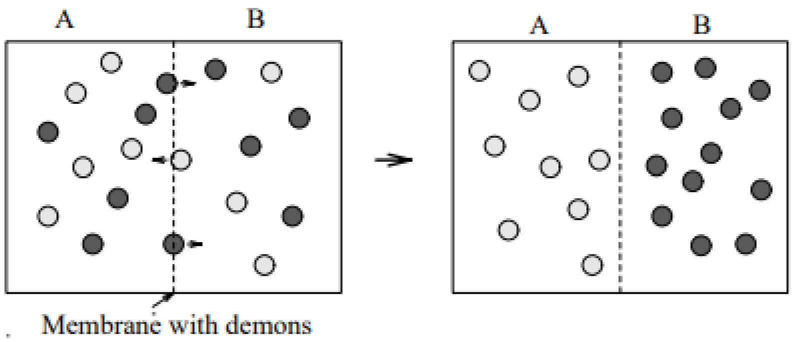


Figure 9: Maxwell's demons and a mixed gas [6]. See text for details.

The concept of demons and the diffusion process was applied to image processing as a non-parametric non-rigid type of image registration [6]. Given two images, one is set as a model image, M, and one is set as a scene image, S [6]. The model image is to be deformed in order to match the scene image [6]. Within the scene image, the contour of an object, O, is assumed to be a membrane [6]. At any given point on the membrane, a perpendicular vector can be defined [6]. The model image, M, is assumed to be a deformable grid whose vertices are considered particles and can be identified as 'inside particles' or 'outside particles' [6]. The 'demons' are defined as effectors that sit on the membrane and apply forces to push M into O if the particle is an 'inside particle' or away from O if the particle is an 'outside particle' [6]. The forces that the 'demons' apply are computed by optical flow. Eq. 112 shows the calculation; $\vec{\boldsymbol{u}}$ represents the displacement field, m represents the brightness intensity of image M, and s represents the brightness intensity of image S [6]. It should be noted that this only takes into account the brightness intensity constancy assumption of optical flow.

$$\vec{\boldsymbol{u}} = \frac{(m-s)\vec{\nabla}s}{\left(\vec{\nabla}s\right)^2} \qquad \text{Eq. 112}$$

The definitions of ‘inside particle’ and ‘outside particle’ are based on brightness intensities within the contour of an object and outside of it [6]. Figure 10 shows a deformable grid M being pushed by ‘demons’ into the set scene’s static contour, O [6].

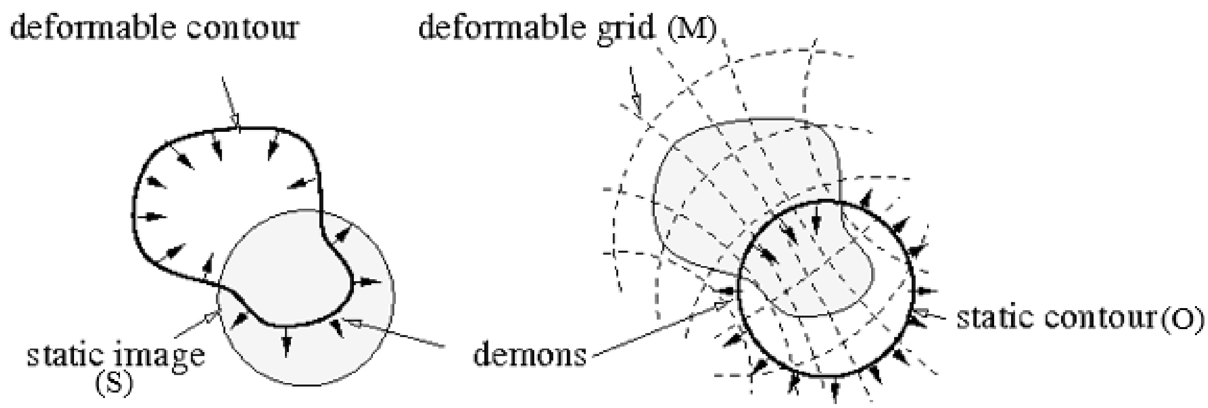


Figure 10: Diffusing model, used to preform image-to-image matching (modified from [6])

2.2.2.1 Implementation of Demons

The general process of computing a displacement field from the demons algorithm based on the diffusion model is an iterative process. First, the set of demons, D_S is extracted from the set scene, S [6]. For each demon, the following information should be known; its spatial position, P, in S, its demon direction (typically gradient $\vec{\nabla} s(P)$), the current displacement of P between S and M, and the brightness intensity of P in S [6]. Once the demons are defined, the iterative process begins. For each demon, the elementary demon force is computed [6]. This force is dependent on both the demon direction and the polarity of the demon point P in image M [6]. The transform, T, of the next step is based on the demon forces of the current step and the set of allowable deformations, τ [6].

There are many possible variants of the general form of the algorithm, four in particular to be considered. First, which points are selected as demons: D_S can be the entire image where one demon is assigned per voxel, D_S can be only points of the contour, or D_S can be points located by edge detection methods [6]. Second, what types of deformations are allowable to form the transform, T: rigid, affine, spline, or free form [6]. Third, what type of interpolation method is used when the position of a demon on image M is not an integer: linear, spline, etc [6]. Lastly, the formula to calculate the elementary demon force: constant magnitude, gradient, or optical flow [6].

It was found that one combination of these factors was most successful in determining displacement fields between medical images [6]. First, for demon assignment, a demon should be assigned to every non-zero brightness intensity voxel [6]. Second, for allowable deformations, free form deformation should be used followed by a Gaussian smoothing filter to determine the transformation of every iteration [6]. The Gaussian filter controls the smoothness of deformation between neighbouring voxels [6]. Third, for interpolation method, trilinear interpolation was used to assign the brightness intensity of demon point P in transformed image M [6]. Lastly, for determining the elementary demon forces, the optical flow equation was the most effective [6].

Similar to the Brox optical flow methods discussed in Section 2.2.1.2, a multi scale approach was used to increase the robustness of the results from the demon algorithm [6]. A scale factor of 0.5 was applied. Computationally, for 3D images, it was found to be more efficient as only 1/8 of the voxels from the next finer level had to be analysed [6].

One of the limitations of Thirion's algorithm is it does not enforce diffeomorphic transformations [25]. Diffeomorphic transformations are transformations in which the grid does not fold over itself. Diffeomorphic transformations are invertible. Non-diffeomorphic transformations are not physically possible. Vercauteren et al. built on the demon's method to enforce diffeomorphic transformations [25]. Instead of optimizing a global energy equation over the complete space of non-parametric transformations, it is only optimized over a space of diffeomorphisms [25].

2.3 MOTIVATION

The purpose of this thesis is to work towards a system to effectively and efficiently validate biological finite element models using 4D medical images. The aortic valve was chosen as the biological model to test on as any solution that could manage the complexity of the valve's motion would likely work for simpler biological models. The motion estimation method options were narrowed down to only direct methods because of the benefits of dense displacement fields. Optical flow was chosen as a possible motion estimation technique for its robustness, and its previous use in conjunction with finite element methods when analysing abdominal aortic aneurisms [4]. The Demon's method was also chosen as a possible motion estimation technique because it is the method built into MATLAB for 3D image motion estimation, and because it is, in theory, better able to capture the motion of rigid transformations. The future goal of this work is to create a code package that would read, analyse, and compare 4D medical images to a corresponding dynamic finite element model. If the displacements of the nodes matched, then the stresses and strains determined by the finite element model could be considered validated. The inputs to this package would include: a DICOM image stack, and the initial nodal position coordinates from a corresponding finite element model. The output of this package would be the new locations of each of the node at any point of time captured by the DICOM image stack. The new location of each node would be compared to the corresponding location of the node determined by the finite element model at the same point in time. Agreement between the two sets of nodal locations, and the capability of node-to-node comparisons would act as a validation tool more robust than any current method of validation for biological finite element models.

3 METHODS

3.1 OVERVIEW OF THE PROCESS

As a proof of concept, the objective outlined above was approached using ideal synthetic medical images. Ideal synthetic medical images were made using an existing finite element model created for an aortic valve. The finite element model was exported as STL shape files at 20 time-steps evenly spaced throughout the cardiac cycle. These shape files were then converted into DICOM files that could be processed as 3D images. The DICOM files represent ideal medical images, because, they contain zero noise, and have an isotropic resolution of 0.5 mm. The resolution was chosen to match current CT scan technology. Typical aortic valve leaflets are 0.5 mm to 3 mm in thickness. It is possible that some valve leaflets on the low end of the thickness could cause issues as they would only be one voxel thick; however, it is unavoidable until the imaging technology improves. Since the DICOM files were created from a finite element model, the nodal information of the finite element model represented the ground truth. The DICOM files were analyzed using three different motion estimation algorithms: two optical flow algorithms and one demons algorithm. The accuracy of the three methods of motion estimation were judged by the comparison of the nodal displacements from the algorithms and the ground truth nodal displacements using several error metrics. The following sections will detail components of the method: how the DICOM files were prepared, the implementation of the 3D Horn-Schunk optical flow method, the implementation of the 3D Brox optical flow method, the implementation of the demons image registration method, the process of assigning nodal displacements based on the optical flow displacement fields, and the metrics for evaluation of how closely the nodal locations determined by the algorithm relate to the ground truth. Lastly, the effect of noise will be briefly examined by the addition of Gaussian noise at varying levels to the synthetic DICOM images.

3.2 PREPROCESSING OF DATA FROM THE GROUND TRUTH FINITE ELEMENT MODEL

3.2.1 Preparing the Synthetic DICOM Files

The creation of synthetic medical images was necessary to establish a ground truth for the displacement of individual nodes of an aortic valve throughout a cardiac cycle to compare to the results of the motion detection techniques. The mesh of the unpressurized aortic valve's finite element model for the conduit of the valve, and the leaflet of the valve is shown in Figure 11. The STL file of the aortic valve exported from LS-Dyna (LSTC, Livermore, CA) at one time point of the cardiac cycle is shown in Figure 12. Twenty STL files from an existing aortic valve's finite element model were exported from LS-Dyna (LSTC, Livermore, CA.

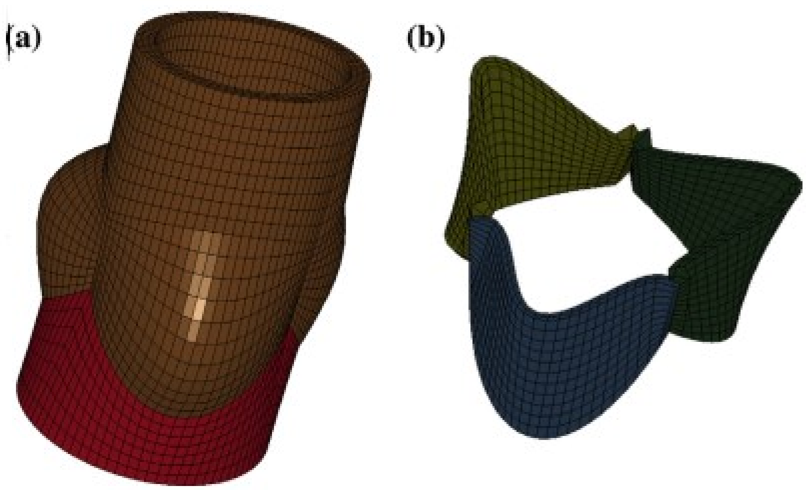


Figure 11: Unpressurized finite element model showing mesh for (a) aortic valve conduit and (b) aortic valve leaflets

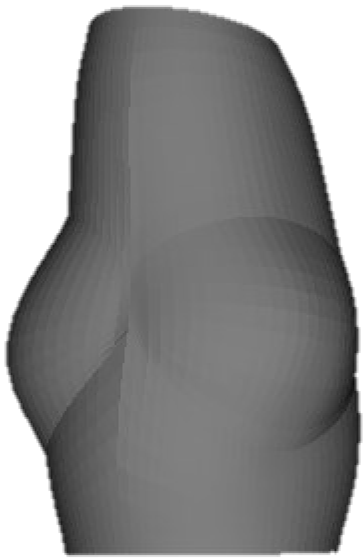


Figure 12: STL file of the existing aortic valve exported from LS-Dyna (LSTC, Livermore, CA)

These 20 images were evenly spaced across a cardiac cycle. 20 images across a cardiac cycle matches the capability of 4D CT scans that will be used for future research. The nodal locations for each of the 13,359 nodes at each of the 20 time steps were recorded and were considered the ground truth for the analysis. To transform the STL model into DICOM files, 20 STL files were imported into 3DSlicer 4.10.2. as models. The models were converted to segmentation nodes. A specific volume, the “Master Volume” was created using the “crop volume” module. Using one specific volume to segment all the images allowed all the images to have the same size of voxels, the same number of voxels, and the same origin location. The Master Volume was sized to encompass the entirety of the valve at all 20 time steps with at least three voxels worth of excess space on each side. Excess space allowed the image analysis algorithms that can be sensitive or inaccurate around image boundaries to run without issues. The voxel size defined in the “Master Volume” was chosen to be 0.5 mm isotropic. This size was chosen as it matches the capability of 4D CT scans that will be used for further research. The origin was defined at 25 mm in all directions in reference to the origin of the finite element model. The total dimensions of the image were set to 120x91x89 voxels; rows, columns, slices respectively. In the module “Segment Editor”, the segment nodes were assigned to the “Master Volume”, which created a set of segments of the stated dimensions and spacing. These segments were exported using the function "export visible segments to binary label map". Finally, DICOM files were created from the binary label map by the function "export to DICOM". Figure 13 shows a front view cross section of the STL file (red) and the DICOM file (green) for time step 1. It can be seen that the conversion from STL to DICOM did not change the overall shape of the valve. Figure 14 shows a zoomed in section of the front view cross section of the STL file (red) and the DICOM file (green) for time step 1. In this figure, one can appreciate that when the STL file is converted into voxels, information that was part of the STL can be lost, or information that was not a part of the STL can be included.

Figure 13: Conversion of time step 1 from STL file (red) to DICOM file (green)

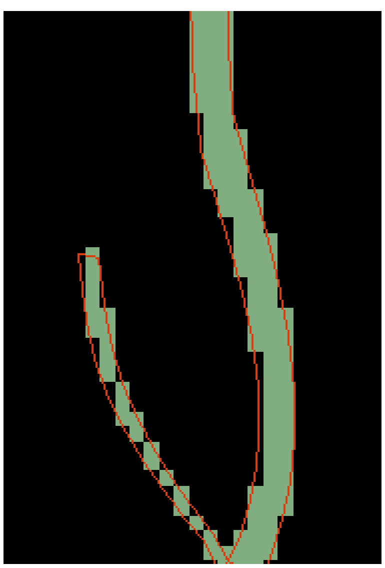


Figure 14: Conversion of time step 1 from STL (red) to DICOM file (green) zoomed in

These DICOM images were imported as structure arrays, and converted into double-precision arrays. The brightness intensity of the grey scale arrays varied between zero and one. These arrays were used as the input images for the motion estimation algorithms. The front and side cross section views for time steps 1 through 20 are shown in Figure 15 and Figure 16 respectively. The images were run through a Gaussian smoothing filter before they were processed in order to reduce the difference between the STL and DICOM files, as well as to prepare them to be analyzed for optical flow which requires a range of grey-scale voxel values.

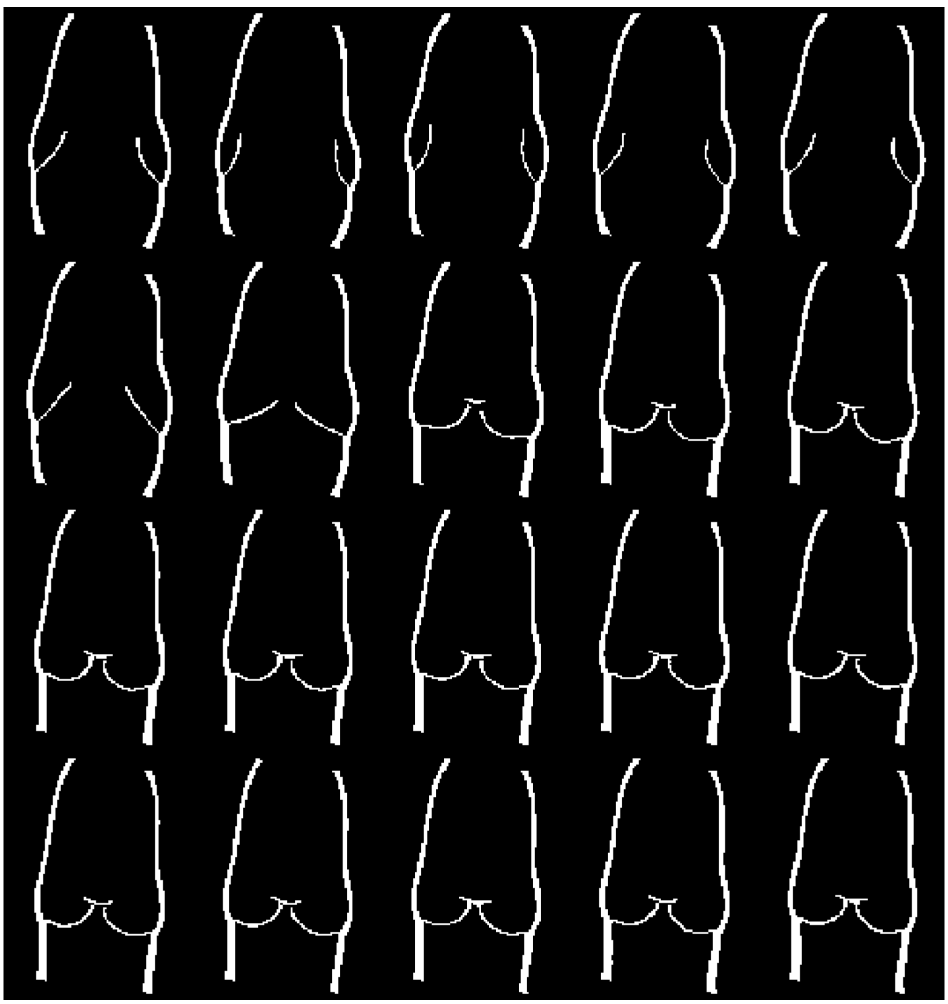


Figure 15: Front view cross section of DICOM images for time steps 1-20

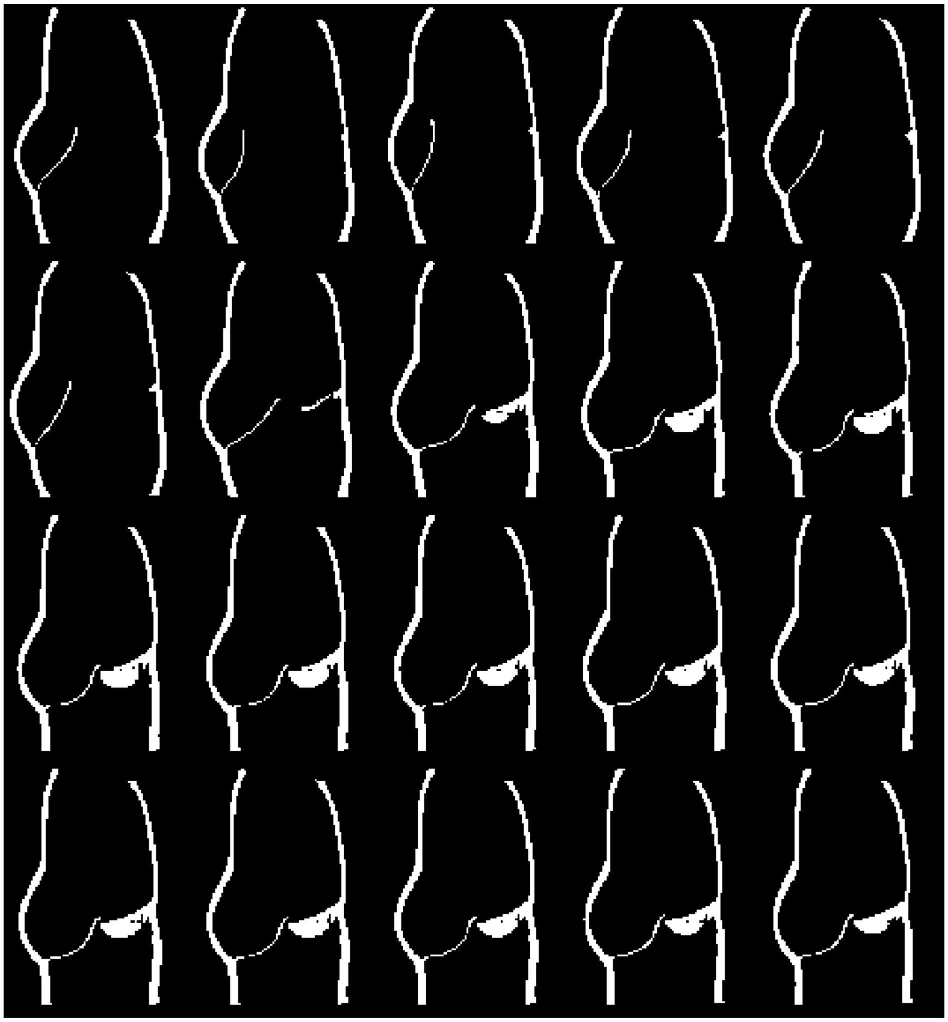


Figure 16: Side view cross section of DICOM images for time steps 1-20

3.2.2 Creating Lists of Leaflet Nodes and Aortic Nodes

The three leaflets of the aortic valve attach to the conduit of the aortic valve at the aortic annulus and the commissure points on the sinotubular junction. The leaflets open and close within the valve, and the sides of the aorta (the conduit of the valve) pulsate with pressure variations during the cardiac cycle. The cardiac cycle is made up of two phases, systole and diastole. The detailed breakdown of the events that occur during the cardiac cycle are shown by the Wiggers diagram in Figure 17. In diastole, the heart relaxes and fills with blood, and in systole the blood is expelled by contraction of the heart. When the mitral valve closes at the beginning of systole, the ventricular pressure starts to build up. When the ventricular pressure becomes higher than the aortic pressure, the aortic valve opens. When the aortic valve is open, ejection occurs until the aortic pressure becomes higher than the ventricular pressure and the aortic valve closes and remains closed until the next cycle. Figure 18 illustrates the shape of the aortic valve and how it relates to the ventricular and aortic pressure. Throughout the cardiac cycle, it is expected that the leaflets undergo larger displacements than the aorta. Due to the vastly different expected displacements, a list of the nodes that make up the leaflets of the valve and a list of the nodes that make up the aorta will be useful in the analysis of the different motion estimation methods. This was done using the software ParaView 5.7.0-RC1. The timestep of the valve that will have undergone the greatest displacement from the first time step was the 20th time step, at end diastole. A VTK file (Visualization Toolkit file) containing the ground truth nodal displacements between the first and the last time step was written using the function **WriteVTKFiles.m**. The pseudo code for **WriteVTKFiles.m** is shown in Section 8.1.1. Within ParaView, the VTK file shows the original locations of the nodes, and the magnitude of the displacements of the nodes. The original locations of the nodes are visualized by a point cloud in the shape of the valve. The point cloud is overlayed with the magnitude of the displacements of each node, which are visualized by a color gradient and corresponding legend. The ParaView visualization of the ground truth nodal displacement between timestep 1 and timestep 20 is shown in Figure 19. Within ParaView data can be extracted from the VTK file. The nodes were sorted based on their magnitude of displacement. It was found that when the threshold was set at a magnitude of displacement greater than 3.5 mm, only the nodes from the leaflets were selected. This subset is shown highlighted in pink in Figure 20. These nodes were added to the list of leaflet nodes. Any nodes that had a magnitude less than 3.5 mm were added to the list of aortic nodes.

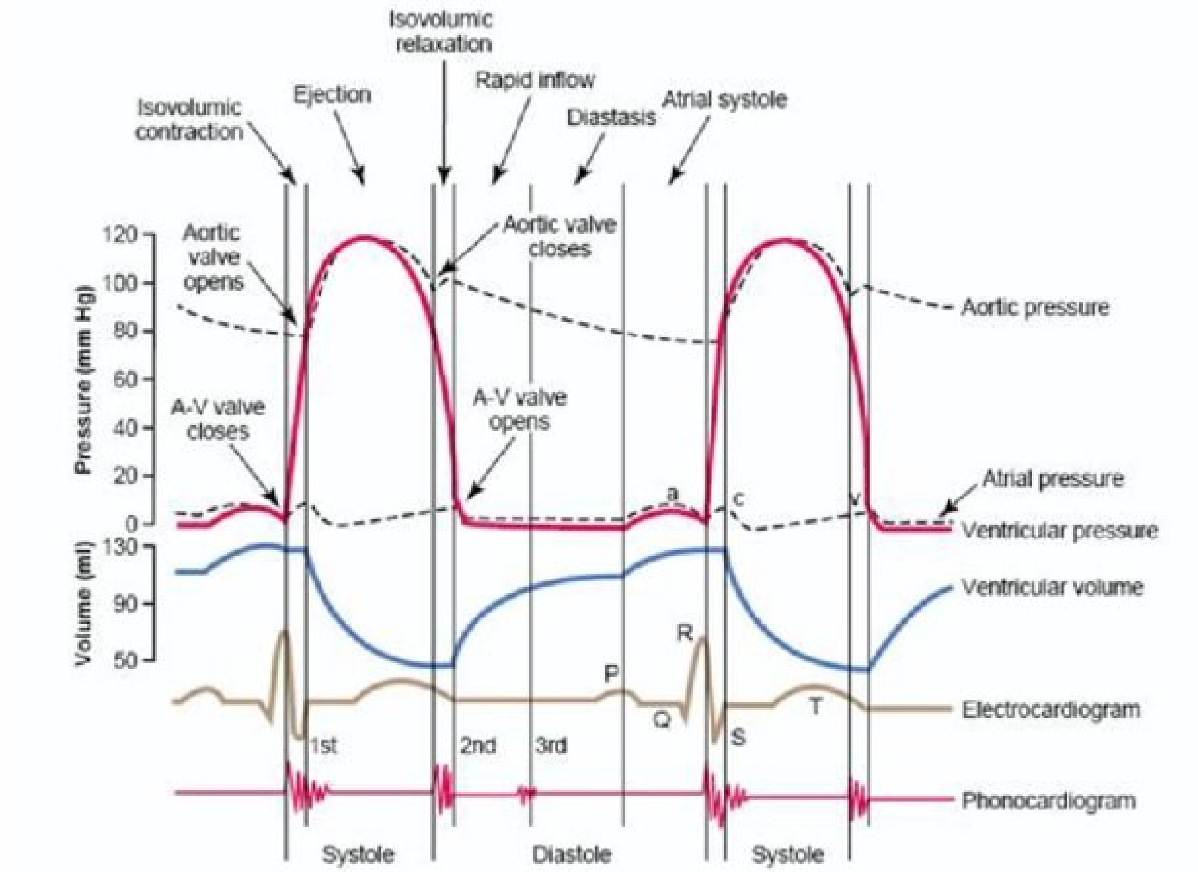


Figure 17: Wiggers diagram showing how the aortic valve behaves throughout the cardiac cycle [26]

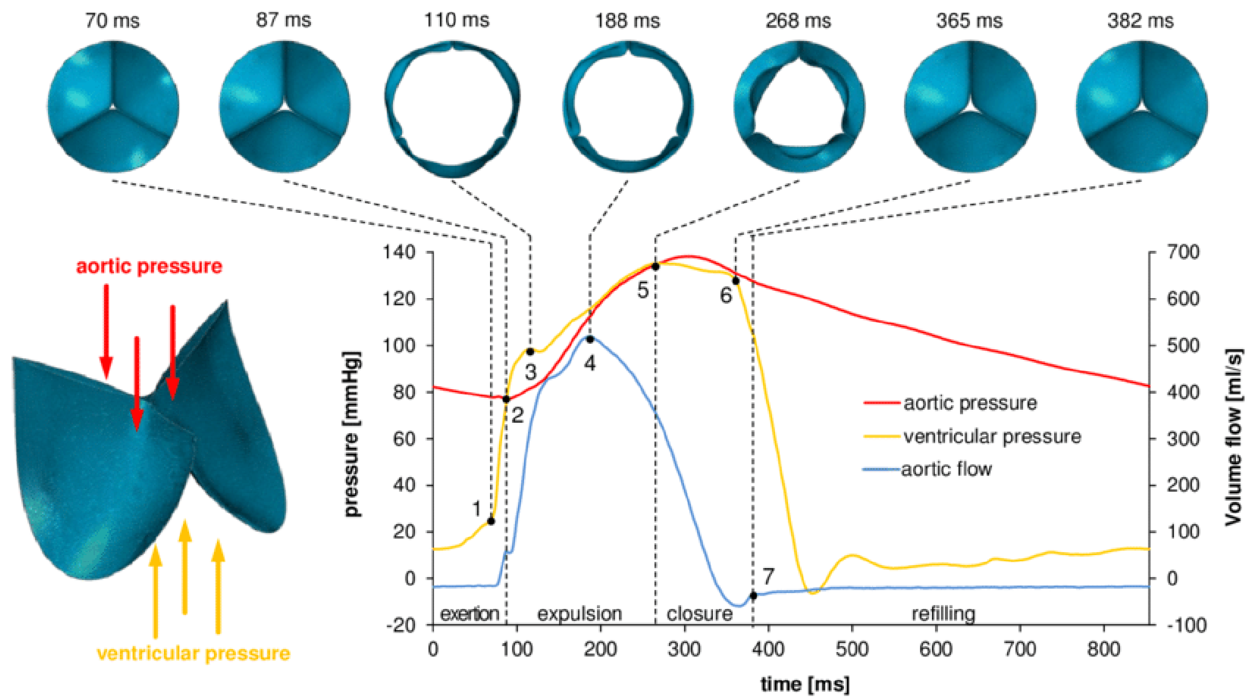


Figure 18: Shape of aortic valve across cardiac cycle based on the aortic pressure and the ventricular pressure [27]

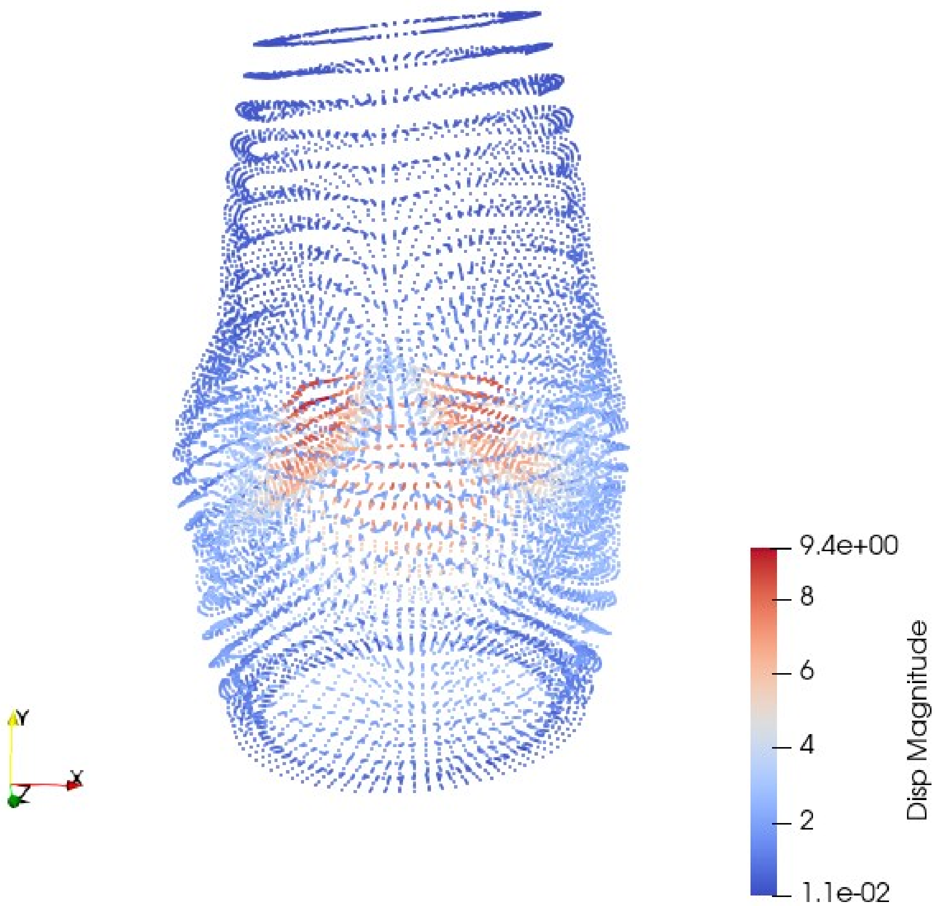


Figure 19: ParaView visualization of the ground truth displacement magnitude (mm) of each nodes between timestep 1 and timestep 20

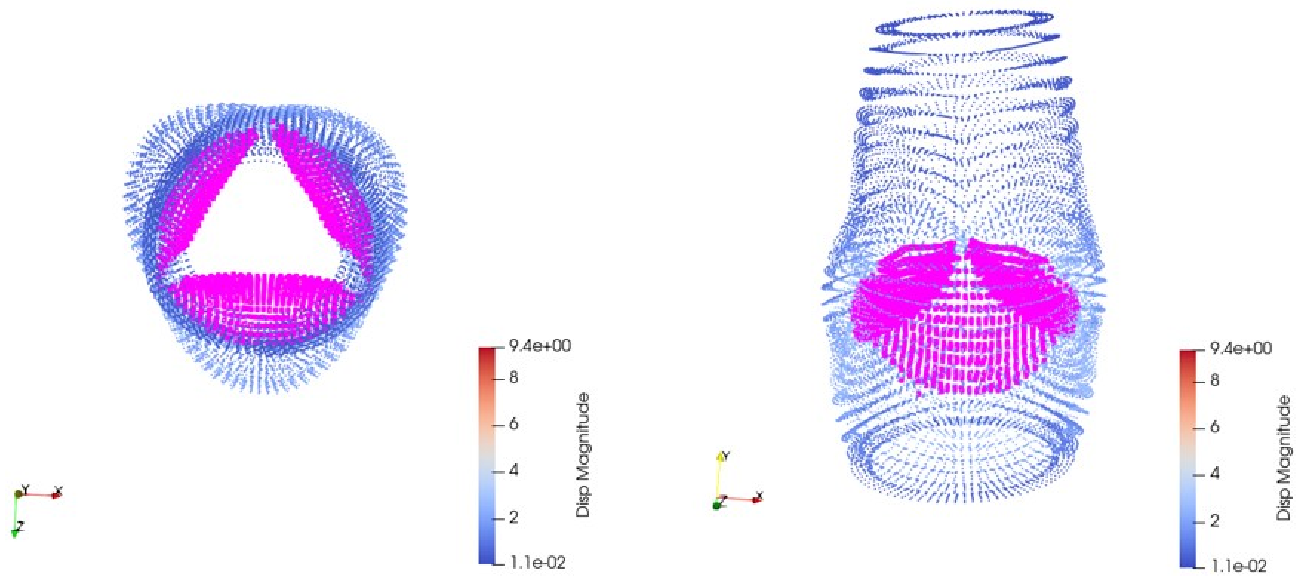


Figure 20: ParaView visualization of leaflet node subset (pink nodes)

3.3 Horn-Schunck Optical Flow

The pseudo code of the implementation of 3D Horn-Schunck optical flow within the validation method is given in Section 8.1.2. The base function of the 3D Horn-Schunck optical flow was found on MathWorks File Exchange as an open-source code [5]. The function takes in the inputs of: two sequential *images*, an *alpha value*, number of *iterations*, and initial guesses for *u*, *v*, and *w*. The function outputs a voxel displacement field, *u*, *v*, and *w.* The function implements 3D Horn-Schunck optical flow as described in Section 2.2.1.1. The expression used by the function to find the displacement field, *u*, *v*, and *w,* are given by Eq. 113 to Eq. 115. These expressions are solved through an iterative method. In these equations *u*, *v*, and *w* represent the displacement field of the voxels, *n* denotes the iteration, $\frac{\partial I}{\partial x}, \frac{\partial I}{\partial y}, \frac{\partial I}{\partial z}$ represent the spatial derivatives in the x, y, and z directions respectively, $\frac{\partial I}{\partial t}$ represents the temporal derivative, and α is the variable parameter in the energy function.

$$u^{n+1} = \bar{u}^n - \frac{\left(\frac{\partial I}{\partial x}\right)\left(\left(\frac{\partial I}{\partial x}\right)\bar{u}^n + \left(\frac{\partial I}{\partial y}\right)\bar{v}^n + \left(\frac{\partial I}{\partial z}\right)\bar{w}^n + \left(\frac{\partial I}{\partial t}\right)\right)}{\left(\left(\frac{\partial I}{\partial x}\right)^2 + \left(\frac{\partial I}{\partial y}\right)^2 + \left(\frac{\partial I}{\partial z}\right)^2 + \alpha^2\right)} \quad \text{Eq. 113}$$

$$v^{n+1} = \bar{v}^n - \frac{\left(\frac{\partial I}{\partial y}\right)\left(\left(\frac{\partial I}{\partial x}\right)\bar{u}^n + \left(\frac{\partial I}{\partial y}\right)\bar{v}^n + \left(\frac{\partial I}{\partial z}\right)\bar{w}^n + \left(\frac{\partial I}{\partial t}\right)\right)}{\left(\left(\frac{\partial I}{\partial x}\right)^2 + \left(\frac{\partial I}{\partial y}\right)^2 + \left(\frac{\partial I}{\partial z}\right)^2 + \alpha^2\right)} \quad \text{Eq. 114}$$

$$w^{n+1} = \bar{w}^n - \frac{\left(\frac{\partial I}{\partial z}\right)\left(\left(\frac{\partial I}{\partial x}\right)\bar{u}^n + \left(\frac{\partial I}{\partial y}\right)\bar{v}^n + \left(\frac{\partial I}{\partial z}\right)\bar{w}^n + \left(\frac{\partial I}{\partial t}\right)\right)}{\left(\left(\frac{\partial I}{\partial x}\right)^2 + \left(\frac{\partial I}{\partial y}\right)^2 + \left(\frac{\partial I}{\partial z}\right)^2 + \alpha^2\right)} \quad \text{Eq. 115}$$

3.3.1 Average Displacements

The function calls for $\bar{u}^n$, $\bar{v}^n$, and $\bar{w}^n$ which represent the average displacement of the voxels at the current iteration. The average displacement of the voxels at the current iteration is approximated by the convolution of the *u*, *v*, and *w* result from the previous iteration with an averaging mask. The averaging mask used in the function is a 3X3X3 kernel shown in Eq. 116.

$$\begin{aligned} Kernel(:,:,1) &= \begin{bmatrix} 0 & 0 & 0 \\ 0 & \frac{1}{6} & 0 \\ 0 & 0 & 0 \end{bmatrix} \\ Kernel(:,:,2) &= \begin{bmatrix} 0 & \frac{1}{6} & 0 \\ \frac{1}{6} & 0 & \frac{1}{6} \\ 0 & \frac{1}{6} & 0 \end{bmatrix} \\ Kernel(:,:,3) &= \begin{bmatrix} 0 & 0 & 0 \\ 0 & \frac{1}{6} & 0 \\ 0 & 0 & 0 \end{bmatrix} \end{aligned} \quad \text{Eq. 116}$$

3.3.2 Spatial and Temporal Differentiation

A discrete derivative is used as an approximation of the derivative of an image. The discrete derivative finds change in brightness intensity across time, for the temporal derivative, $\frac{\partial I}{\partial t}$, and across the direction of interest, for the spatial derivatives, $\frac{\partial I}{\partial x}, \frac{\partial I}{\partial y}, \frac{\partial I}{\partial z}$. The spatial derivatives were found using forward differences which was implemented by Horn-Schunck in their work [18]. For each direction, x, y, and z, the two images were convolved with 2X2X2 directional kernel. The directional derivative was said to be the average of the result from the convolution of image 1 and the convolution of image 2 with the directional kernel. The directional kernels used in the convolution were; dx(:,:,1)=[-0.25, 0.25; -0.25, 0.25] and dx(:,:,2)=[-0.25, 0.25; -0.25, 0.25] for the x-direction, dy(:,:,1)=[-0.25, -0.25; 0.25, 0.25] and dy(:,:,2)=[-0.25, -0.25; 0.25, 0.25] for the y-direction, and dz(:,:,1)=[-0.25, -0.25; -0.25, -0.25] and dz(:,:,2)=[0.25, 0.25; 0.25, 0.25] for the z-direction. The temporal derivative was also found using forward differences. The two images were convolved with a 2X2X2 kernel with all the indices 0.25. The temporal derivative was said to be half the difference between the results of the convolution.

3.3.3 Alpha Value Selection Process

The alpha value in Horn-Schunck optical flow is used to define the trade off between the two constraints, brightness constancy, and velocity smoothness [18]. Low alpha values increase the emphasis of the brightness constancy assumption, while high alpha values increase the emphasis of the smoothness constraint [18]. There are drawbacks to both low and high alpha values; low alpha values can be more susceptible to noise, while high alpha values can have difficulty capturing large magnitudes of displacement [18].

Initially, the alpha value was originally set to 200 as used by Satriano et al. in their work using Horn-Schunck optical flow on abdominal aortic aneurysms [4]. This was found to be too high for aortic valves as the displacement of the valve leaflets throughout the cardiac cycle has a much larger magnitude than the displacement of an abdominal aortic aneurysm throughout the cardiac cycle. Therefore, an optimization test was completed on the model to find the alpha value that would minimize the difference error between the Horn-Schunk optical flow nodal displacements and ground truth nodal displacements as shown in Figure 21. It was found that an alpha value of 0.9 was optimal for this model.

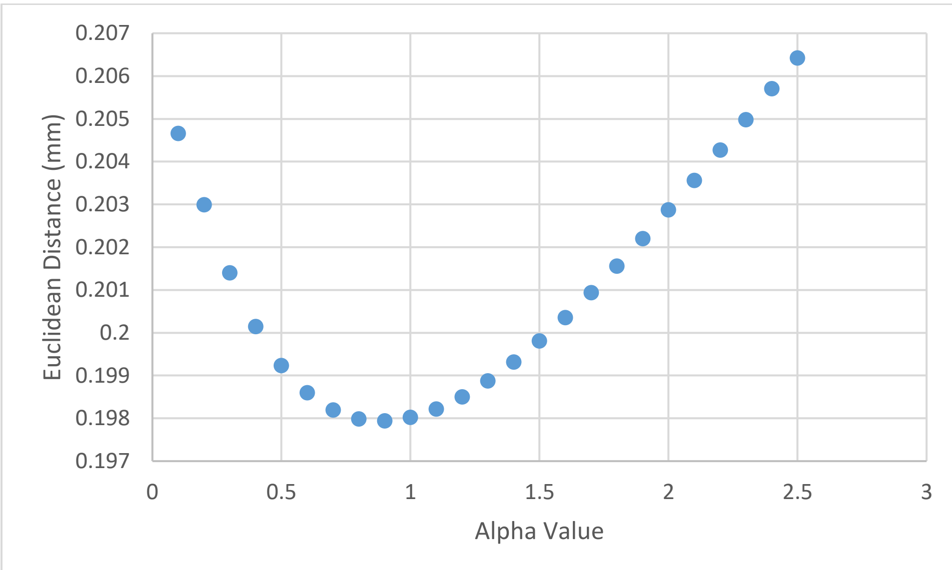


Figure 21: Alpha value error minimization - average difference of magnitude between optical flow and ground truth nodal displacements

Initially, the algorithm was tested for each time step of the image sequence, 1 to 2, then 2 to 3, until 19 to 20. In these cases, an alpha value of 1 produced adequate displacement field results from the Horn-Schunck optical flow algorithm in comparison to the ground truth displacement field. Unfortunately, this method was not feasible for the application of real 4D CT scan images as the inputs would require the starting nodal locations for each time step. The expected inputs to the validation method would be a time step sequence of images where only the first image is segmented to create the finite element model. The nodal locations would only be known for the first timestep; therefore, every displacement field computed needs to be in reference to the first timestep. The algorithm was run for the cases where the first image was set as the first time step of the 20-image cardiac cycle sequence, and the second image was iterated from time steps 2 to 20 using an alpha value of 1. It was found that the alpha value of 1 accurately captured the displacement field for the first couple of images in the sequence. In these images, the valve leaflets, and their associated nodes, did not undergo large displacements. However, an alpha value of 1 was unable to capture the displacement field of the leaflets later in the image sequence where the valve leaflets, and their associated nodes, underwent larger magnitudes of displacement while moving to the closed position. This finding is illustrated by Figures 22 and 23. Figure 22 shows three cross sections of the aortic valve; the first image of the cardiac cycle on the left side, the second image of the cardiac cycle on the right side, and the warped image, time step 2 to time step 1, in the center. The warped image is the second image warped using the displacement field determined by the Horn-Schunck optical flow algorithm. The middle image resembles the first image closely which suggests that the alpha setting for the optical flow algorithm is effective in this case. Figure 23 shows three cross sections of the aortic valve; the first image of the cardiac cycle on the left side, the seventh image of the cardiac cycle on the right side, and the warped image in the center. The warped image is the seventh image warped using the displacement field determined by Horn-Schunck optical flow. The center image does not resemble the left image. The annulus shape is similar, but the leaflets more closely resemble the image on the right which suggests that the alpha setting for the optical flow algorithm is not effective in this case. The leaflets makeup only a small percentage of the total nodes in the aorta, so even though the alpha value of 1 was optimal based on the difference, it does not accurately represent the leaflet displacement.

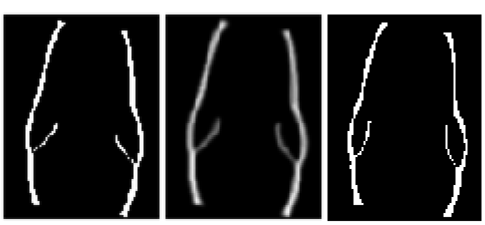


Figure 22: Image warping from time step 2 to time step 1 using α=1: time step 1 (left), warped image 2 to 1 (middle), time step 2 (right)

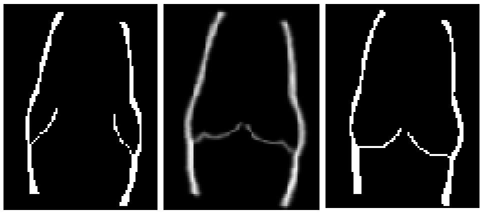


Figure 23: Image warping from time step 7 to time step 1 using α=1: time step 1 (left), warped image 7 to 1 (middle), time step 7 (right)

The alpha value was slowly decreased until it was visually able to reflect the motion of the valve leaflets. Figure 24 shows the method using an alpha value of 0.25. The leaflets on warped image (center) resemble the first image (left). The downside to using a lower alpha value is that the difference has a larger tail as shown in the histograms is Figure 24. Even though visually the warped image using the lower alpha value looks more accurate based on the position of the leaflets, numerically, over all the nodes, the error is higher. No single alpha value will adequately evaluate the nodal displacement field using 3D Horn-Schunck optical flow model.

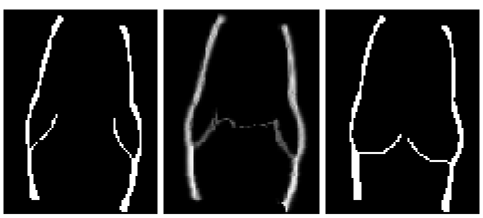


Figure 24: Image warping from time step 7 to time step 1 using α=0.25: time step 1 (left), warped image 7 to 1 (middle), time step 7 (right)

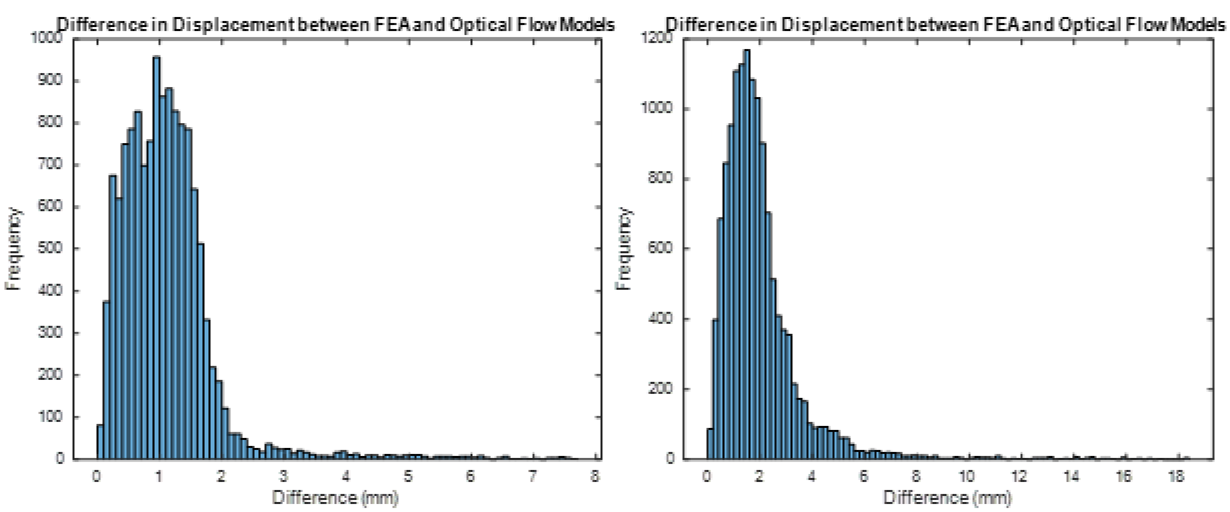


Figure 25: Histograms of difference between ground truth and Horn-Schunck optical flow nodal displacement fields using an alpha value of 1 (left) and 0.25 (right)

3.3.3.1 Selection of Alpha Value Dependent on Position

We observed that no single alpha value was sufficient to evaluate the nodal displacement field across the entire cardiac cycle using 3D Horn-Schunck optical flow. Lower alpha values were able to sufficiently capture the motion of the leaflets as the valve closed; however, the overall error increased. Higher alpha values were able to maintain reasonable error levels for the majority of the nodes; however, they were unable to capture the motion of the valve leaflets. Therefore, we decided to investigate different alpha values for different parts of the valve. The two components of the aortic valve considered were the leaflets and the aorta. The leaflets undergo higher magnitudes of displacement than the aorta throughout the cardiac cycle. Less weight on the velocity smoothness constraint, i.e. a lower alpha value, can be expected to provide a more accurate displacement field for the leaflets. Higher weight on the velocity smoothness constraint, i.e. a higher alpha value, can be expected to yield a more accurate displacement field for the aorta.

An automatic selection of the alpha values for the valve leaflets and for the valve aorta was embedded into the algorithm. The process of choosing which alpha values was based on the comparison of the center cross sections of the warped 3D image and the center cross sections of the first image using the different options for alpha values, namely: 0.2, 0.4, 0.6, 0.8, and 1, as determined from trial and error. The cross sections were two slices from the 3D images; the first, halfway through the valve from the front view, and the second, halfway through the valve from the side view. The cross sections of image 1 visualized as part of the volume is shown in Figure 26. The optical flow algorithm was run for all alpha values of interest and warped images corresponding to each alpha value were created. Each warped image was compared to the first image by taking the absolute value of the difference between the brightness intensity values at each voxel. An example of a resulting image is shown in Figure 27. The sum of the absolute values of all the differences for each voxel, the sum of differences, was calculated and added up for both front view and side view images. Whichever alpha value resulted in the lowest sum of brightness intensities of the difference image was chosen as the best alpha values for the aorta. The same process was used to determine the alpha value best suited for the leaflets except that the rows containing the leaflets were considered. The voxel displacement fields from both selected alpha values were converted into nodal displacement fields and the nodes were assigned displacements based on weather they belonged in the leaflet node list or aorta node list as described in Section 3.2.2.

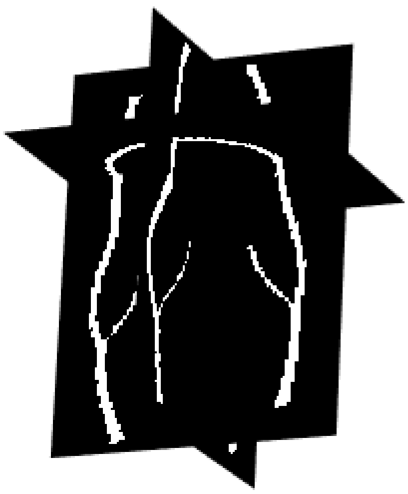


Figure 26: Cross section slices of 3D image

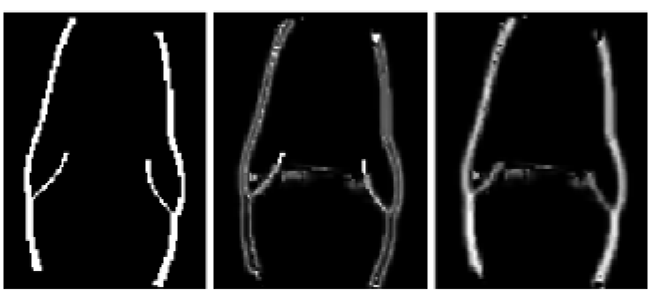


Figure 27: Image 1 (left), difference between brightness intensities of warped image and image 1 (center), warped image (right)

3.3.4 Pyramid Scheme

The purpose of implementing a pyramid scheme within the optical flow algorithm was to enable the algorithm to capture larger movements between frames. The classic Horn-Schunck optical flow approach is limited to voxel displacement magnitudes of less than one voxel, otherwise the algorithm tends to get trapped in a local minimum as opposed to the global minimum. By downsizing the image to the point where the size of a voxel is greater than the maximum expected displacement, optical flow is able to better capture the larger magnitudes of voxel displacement. This coarse solution can then be used as initialization for solving refined versions of the problem as the image scales. The application of the pyramid scheme involves three steps: resizing the images, projecting velocities between pyramid levels, and warping images.

Images were resized based on two variables: scale factor, and number of pyramid levels. As suggested by Brox et al., instead of the classic scale factor of 0.5 used in Gaussian pyramids, an arbitrary scale factor was introduced allowing for smoother transitions between pyramid levels [9]. In this case, the scale factor was set to 0.95. The number of levels was determined in order to satisfy the constraint that the magnitude of a voxel's displacement cannot exceed the size of the voxel; this is given by Eq. 117. The voxel size was set as 0.5 mm based on current medical imaging capabilities. The maximum nodal displacement of the aortic valve model, 9.4 mm, was found using the difference between the nodal locations from the ground truth finite element files. This was the maximum magnitude of displacement from the contributions of the three directions, x, y, and z. Based on these parameters, the number of pyramid levels was set to 60.

$$\frac{Voxel\ Size\ (mm)}{Scale\ Constant^{Number\ of\ Levels}} > Maximum\ Nodal\ Displacement \qquad \text{Eq. 117}$$

A flow diagram of how the pyramid scheme was implemented to move between levels from coarse images to fine images is shown in Figure 28. The size of the text is proportional to the size of the arrays for the images and displacement fields. At each level of the pyramid the images, Image 1 and Image 2, were smoothed and scaled from the top-level image. The images were smoothed using a 3D Gaussian smoothing filter. The Gaussian smoothing kernel has a standard deviation specified by sigma which is set to $\frac{1}{Scale\ Factor}$. The images are resized using to the scale of $Scale\ Constant^{Pyramid\ Level}$. Cross-sections from the top 10 levels of the pyramid at time step 1 is shown in Figure 29.

The coarsest level of the pyramid had the smallest images with voxels representing the largest area. The finest level of the pyramid were the original images. These were the largest images, and their voxels represented 0.5 cubic mm boxes. Starting from the coarsest level, the displacement field between the two resized images was computed using the Horn-Schunck optical flow algorithm. This algorithm outputted the displacement field in the unit of voxels. To account for the change in the size of the voxels between pyramid levels, the displacement field from the coarser pyramid level was scaled by a factor of $\frac{1}{Scale\ Factor}$ and was then resized to the next less coarse level.

The newly sized displacement field was used as an input to warp Image 2 of the next level. This warped image moved the voxels of Image 2 according to the displacement field to create a new image that appeared more similar to Image 1. The purpose of warping Image 2 to appear closer to Image 1 was for the maximum magnitude of the displacement field to always remain smaller than the size of the voxel at the current pyramid level. Larger adjustments were made on the coarser images while smaller adjustments were made on the finer images, the process working in a similar way to microscope adjustment knobs. A new displacement field between the resized Image 1 and warped Image 2 was computed using the Horn-Schunck optical flow algorithm. The new displacement field was scaled by a factor of $\frac{1}{Scale\ Factor}$ and resized to be used as an input to warp Image 2 at the next level of the pyramid. This process was continued until the finest pyramid level was reached at which point the final displacement field should represent the movement of the voxels between the original two images.

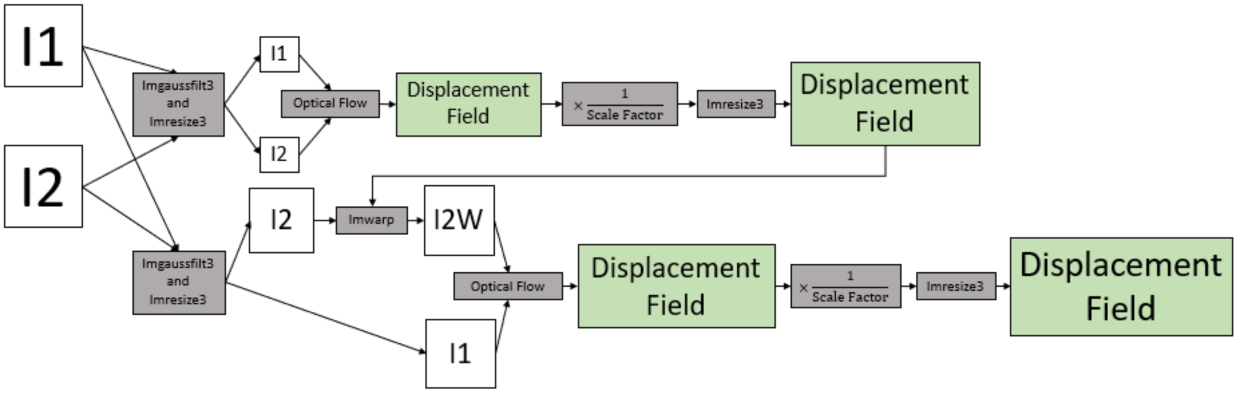


Figure 28: Flow chart of the implementation of pyramid scheme between coarsest and second level

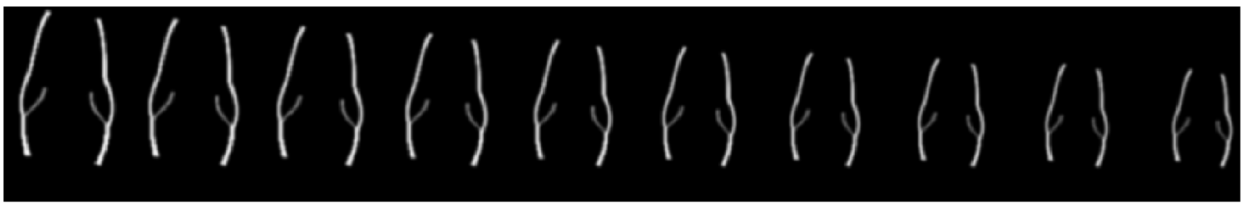


Figure 29: First 10 layers of pyramid for cross section of image at time step 1

3.4 BROX OPTICAL FLOW

The pseudo code of the implementation of 3D Brox optical flow within the validation method is given in Section 8.1.3. The base function of 3D Brox optical flow was developed for this work. The function takes the inputs of: two *images*, an *alpha value*, a *gamma value*, number of *outer iterations*, number of *inner iterations,* and initial guesses for *u, v, and w*. The function outputs a voxel displacement field, *u, v, and w.* The function implements 3D Brox optical flow as described by Section 2.2.1.2. Brox optical flow aims for a more accurate calculation of the displacement field by using an additional constraint, gradient constancy, compared to optical flow. The expressions used by the function to determine the displacement field, *u, v, and w,* are given by Eq. 104 to Eq. 107. These expressions are solved using two levels of iterations as described in Section 2.2.1.2. The inner iteration is denoted by the superscript l, and the outer iteration is denoted by the superscript k.

3.4.1 Spatial and Temporal Differentiation

Unlike a continuous function, an image is made up of a seemingly random array of brightness intensity values. Therefore, to take a derivative of an image, a discrete derivative is used as an approximation of the derivative. The discrete derivative finds changes in brightness intensity across time, for the temporal derivative, or across the direction of interest, for the spatial derivative.

3.4.1.1 Spatial Differentiation

The spatial derivatives were computed by the convolution of the image I, with a Gaussian derivative kernel. The kernel is calculated using Eq. 118. In this equation sigma represents the standard deviation and is defaulted to 1. The kernel is also cropped to exclude values less than 1e-6 to save computation time. The resulting kernel has a size of 1 X n X 1. The outputted kernel's values are used for all three spatial derivatives; however, the orientation is changed for the derivatives in the y, and z-directions. When taking derivative in the y-direction the kernel should be transposed to have dimensions of n X 1 X 1, and when taking the derivative in the z-direction the kernel should by reoriented to have dimensions of 1 X 1 X n. The same kernel is used for computing the second order derivatives. For second order derivatives the kernel is convolved with the outputs of the first order derivatives, I_{xy} is created by convolving I_x with the kernel in the y direction. It is assumed that the second order derivatives are symmetric, $I_{xy} = I_{yx}$. This assumption is based on the original work of Brox et al. [9]. In image processing the spatial derivatives around the boundaries of an image array can be inaccurate. For this reason, when creating the synthetic DICOM files extra blank space was left on each side to reduce any boundary effects. MATLAB defaults to reflecting data at the boundaries of an array, so leaving empty space around the valve in the synthetic DICOM files ensured that the reflected data would also be blank space and not create inaccuracies.

$$Gaussian\ derivative = -\frac{e^{\left(\frac{-x^2}{2\sigma^2}\right)}}{\sqrt{(\pi 2\sigma^2)}}\frac{x}{\sigma^2} \qquad \text{Eq. 118}$$

3.4.1.2 Temporal Differentiation

The temporal derivatives of the image were taken using a 2-point finite difference scheme. The brightness intensity values of each voxel of the second image were simply subtracted from the brightness intensity values of each voxel of the first image as shown in Eq. 119. The same method was applied to compute the spatio-temporal derivatives as shown in Eq. 120 to Eq. 122; the 2-point scheme was used between the first order spatial derivatives at time t, and time t+1. In general, the 2-point difference scheme is a weak method for finding temporal derivatives; however, in this case, because it was imbedded within the pyramid scheme, it was sufficient. At each level of the pyramid scheme, the derivatives were recalculated for the new *u, v,* and *w,* by warping the second image towards the first image. By the top-level *u, v,* and *w* should approach zero as the approximations get more accurate.

$$I_T = I(x+u, y+v, z+w, z+w, t+1) - I(x,y,z,t) \qquad \text{Eq. 119}$$

$$I_{xt} = I_x(x+u, y+v, z+w, t+1) - I_x(x,y,z,t) \qquad \text{Eq. 120}$$

$$I_{yt} = I_y(x+u, y+v, z+w, t+1) - I_y(x,y,z,t) \qquad \text{Eq. 121}$$

$$I_{zt} = I_z(x+u, y+v, z+w, t+1) - I_z(x,y,z,t) \qquad \text{Eq. 122}$$

3.4.2 Pyramid Scheme

The purpose of implementing a pyramid scheme within the optical flow algorithm was to enable the algorithm to capture larger movements between frames. The classic optical flow approach is limited to voxel displacement magnitudes less than one voxel, otherwise the algorithm tends to get trapped in a local minimum as opposed to the global minimum. By downsizing the image to the point where the size of a voxel was greater than the expected displacement, optical flow was able to better capture the larger voxel displacement magnitudes. This coarse solution could then be used as initialization for solving refined versions of the problem as the image scaled. The pyramid scheme applied to Brox optical was the same as the pyramid scheme applied to Horn-Schunck optical flow shown in Section 3.3.4. 60 pyramid levels were used with a scale constant of 0.95.

3.4.3 Convergence of Iterations

Within the pyramid scheme outer and inner iteration loops exist to determine the displacement field at the given pyramid level. To determine the number of iterations needed, the convergence of the results must be examined. The inner iteration, *l*, is used to fine tune $du^{k,l+1}$, $dv^{k,l+1}$, and $dw^{k,l+1}$. The loop breaks when the maximum entry in the array of the magnitude of difference, $\sqrt{(du^{k,l+1} - du^{k,l})^2 + (dv^{k,l+1} - dv^{k,l})^2 + (dw^{k,l+1} - dw^{k,l})^2}$, is less than the set tolerance of 0.00005 voxel units. An array of this criterion is outputted at every pyramid level for every inner and outer iteration. If the value at the highest outer iteration, highest inner iteration, and the top pyramid level is equal to zero, the number of iterations is sufficient for convergence. It was found that 3 outer iterations and 1500 inner iterations was optimal for the aortic valve model.

3.4.4 Selection of Alpha and Gamma Values

The alpha value in Brox optical flow, like Horn-Schunck optical flow, is used to define the trade off between the two constraints, brightness constancy, and velocity smoothness [9]. Low alpha values increase the emphasis of the brightness constancy assumption, while high alpha values increase the emphasis of the smoothness constraint [9]. The gamma value in Brox optical flow, not included in the Horn-Schunck model, is used to define the trade off between the constraints of brightness constancy and gradient constancy [9]. When gamma is set to zero, the Brox model essentially becomes the Horn Schunk model. Both the alpha and gamma values must be tuned to optimise the results of the Brox optical flow algorithm. First, the gamma value was set to zero while the alpha value was optimized, then the alpha value was set and the gamma value was optimized. A process was embedded in the algorithm to determine these parameters. The process for selecting the alpha values was identical to the method explained in Section 3.3.3.1. The options for the alpha value were: 0.02, 0.04, 0.06, 0.08, and 0.1. Once the alpha values were selected, the gamma values were optimized. The alpha value was set to the leaflet setting and a range of gamma values were evaluated. The options for the gamma value were: 0.1, 0.5, 1 and 2. This range was chosen through trial and error. Ideally, all values between 0.1 and 2 would be included, however, that program would use too much memory. The optical flow algorithm was run for all gamma values of interest and warped images corresponding to each gamma value were created. Each warped image was compared to the first image by taking the absolute value of the difference between the brightness intensity values at each voxel. The sum of the absolute values of all the differences for each voxel, the sum of differences, was calculated and added up for both front view and side view images. Whichever gamma value resulted in the lowest sum of differences was chosen as the best gamma values for the leaflet of the valve. The whole process was repeated for the aorta.

3.5 DEMONS

3.5.1 MATLAB Built-In Function - imregdemons

MATLAB has a built-in function to estimate a displacement field between two images in 2D or 3D. The function **imregdemons** applies the demons algorithm as described in Section 2.2.2. The function takes the following inputs: moving image, fixed image, number of iterations per pyramid level, accumulated field smoothing, and pyramid levels [7]. The moving image is what Thirion described as the model image in his work, and the fixed image is what he described as the scene image [6]. The number of iterations drives how many times the elementary demon forces are recalculated and smoothed per pyramid level [7]. The number of iterations must be a positive integer. The function does not apply any convergence criterion; therefore, the number of iterations is the only driving factor on how long the function runs. The accumulated smoothing field gives the option for setting the standard deviation of the Gaussian smoothing applied at each iteration [7]. The default standard deviation is set to 1 [7]. The pyramid levels input gives the option for setting how many levels of resolution are stepped through in the multilevel refinement process [7]. The number of pyramid levels possible is constrained by the smallest dimension of the image. Between each pyramid level the scale factor is set to 0.5; this is a set value within the MATLAB program and cannot be changed. The function gives the following outputs: a displacement field, and an aligned image. The displacement field for a 3D image is an array of size m-by-n-by-p-by-3, where m, n, and p represent the number of rows, columns, and slices of an image respectfully [7]. The range of the fourth dimension of the array accommodates the displacement in the x, y and z directions for each voxel of the image [7]. The aligned image is a new image made out of the moving image warped with the determined displacement field [7]. The process of warping involves moving each voxel of the moving image based on the displacements in the x, y, and z direction given by the displacement field.

Within the context of this work, it was necessary to test the built in MATLAB function for tracking the motion of the set of synthetic DICOM images of an aortic valve. Similar to the application of the optical flow algorithms, the function needed to be able to represent the displacement field between the first DICOM image and any subsequent image between timesteps 2 and 20. DICOM 1, the image at the first timestep, was set as the fixed image. The other DICOM images, 2 through 20, were each set as the moving image. The accumulated smoothing filter was left as its default, 1. The number of pyramid levels was adjusted between 1 and the maximum for the size of the image, 6. It was found that the determined motion of the valve's leaflets was inaccurate using these inputs. The leaflets of the resultant warped image were positioned more similarly to the image they were being warped from than the image of the first time step. The resultant warped images are shown in Figure 30.

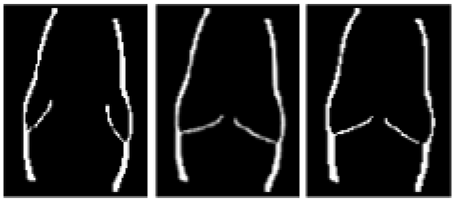


Figure 30: Image warping from time step 7 to time step 1 using demons method with the built in pyramid scheme: time step 1 (left), warped image 7 to 1 (middle), time step 7 (right)

3.5.2 Use of Modified Pyramid Scheme with imregdemons

It was theorized that the reason the results of the initial demons test were inaccurate was because the scale factor of the pyramid scheme was set at 0.5. Similar to the effect of small scale factors when using the optical flow methods, the low scale factor caused too large a difference between the size of images at adjacent pyramid levels to obtain smooth flow projections between the levels. If the scale factor was variable and could be larger, the inaccuracies might be minimized. To achieve this, a pyramid scheme was built around the existing function that allowed for the scale constant to be a variable. The pseudo code of this implementation is given in Section 8.1.4. The MATLAB function imregdemons runs inside the pyramid scheme as shown in Figure 31. The two starting images are smoothed and resized. The function imgaussfilt3 uses a Gaussian smoothing kernel which has a standard deviation specified by sigma which is set to $\frac{1}{Scale\ Factor}$. The images are resized to the size of $Scale\ Constant^{Pyramid\ Level}$. The new images are run through the function **i**mregdemons and a resulting displacement field is outputted. The displacement field is then scaled by a factor of $\frac{1}{Scale\ Factor}$ and resized to be applied at the next pyramid level. In order for the function imregdemons to work with the modified pyramid scheme, the input number of pyramid levels given to imregdemons is set to 1, removing the pyramid scheme contributions within the function. The images inputted into imregdemons are determined by the pyramid scheme code which warps images with the displacement field determined by imregdemons as it moves up through the finer levels.

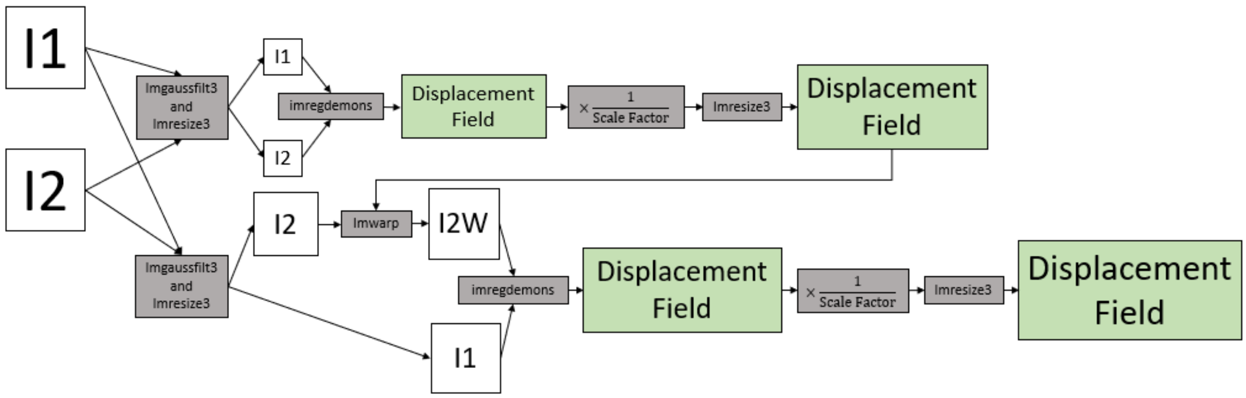


Figure 31: Pyramid scheme built around demons method function

3.5.3 Application of imregdemons with Modified Pyramid Scheme

In application, the algorithm ran with 10 pyramid levels, and the images were decreased in size by a scale factor of 0.95 between each level. Unlike the optical flow algorithms, the demon's function was able to estimate motion when the magnitude of displacement of a voxel was greater than the size of the voxel. For this reason, the number of pyramid levels was not constrained by the size of the voxels. It was found that increasing the number of pyramid levels beyond 10 resulted in less accurate displacement fields. At each pyramid level, imregdemons was run to find the displacement field for the next level. The input for *fixed image*, regardless of the pyramid level, was the image at the first timestep resized to the current pyramid level size. The input for *moving image* depended on the pyramid level. For the coarsest pyramid level, the input for *moving image* was set to the image at the time step of interest resized to the coarsest pyramid level size. For every other pyramid level, the input for moving image was set to the image at the time step of interest warped using the displacement field determined from the previous pyramid levels and then resized to the current pyramid level size. The number of iterations within the function imregdemons was set to 500 for all the pyramid levels. Convergence of the displacement field was tested by checking the results when the number of iterations per pyramid level varied between 300 and 1000. It was found that adding more than 500 iterations did not change the resulting deformation field significantly; however, it increased the computing time. The standard deviation for the Gaussian smoothing was set to the default, 1. When the standard deviation of the Gaussian smoothing was set lower, the displacement field was too inconsistent, and when the standard deviation of the Gaussian smoothing field was set higher, the displacement field could not capture the motion of the leaflets. The process was completed to obtain the voxel displacement fields between the image at time step 1 and the images at time steps 2-20. Figure 32 shows the resultant warped valve cross-section (center), from warping the valve at time step 20 (right) using the determined displacement field to look like the valve at time step 1 (left).

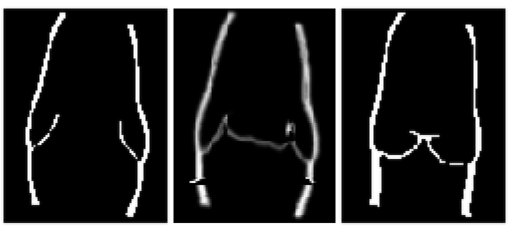


Figure 32: Image warping from time step 20 to time step 1 using demons method with a variable scale constant pyramid scheme: time step 1 (left), warped image 20 to 1 (middle), time step 20 (right)

3.6 Assigning Nodal Displacements from Optical Flow or Demons Displacement Field Outputs

The output from either optical flow or demons algorithm is a voxel displacement field which contains the information about the displacement of each individual voxel that makes up the 3D image. Once the displacement field of the voxels has been determined, the data was used to determine the nodal displacements. The position of every node at time step 1 was known. Every node was assigned to a voxel with a non-zero brightness intensity. Voxels with a non-zero brightness intensity make up the valve model, as opposed to the voxels with zero-value brightness intensity which represent the space around the valve model. In real medical images, which contain other biological entities besides the valve, a higher threshold on the grey value scale can be applied to determine which voxels make up the valve. Depending on the resolution of the image, some nodes from the surface of a finite element model will inevitably not overlap the voxels with non-zero brightness intensity values. This is due to the pixilation that occurs when the STL file is converted into a DICOM image. As the resolution of the DICOM image gets smaller, more nodes will be located in a voxel with a zero-value brightness intensity. As shown in Figure 33, all the nodes located in non-zero brightness intensity voxels were within 3 voxel dimensions of the aortic valve. The following procedure was designed to assign nodes to non-zero brightness intensity voxels.

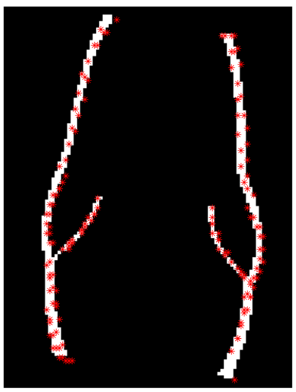


Figure 33: Nodes at ground truth coordinate locations overlapping DICOM image at center slice

If the node location overlaps with a voxel containing image information, a non-zero brightness intensity voxel, then that node is assigned the displacement of that voxel as shown in Figure 34.

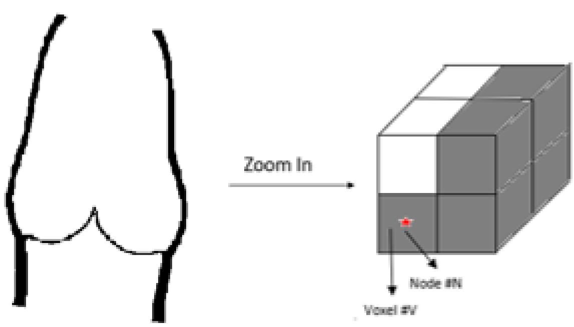


Figure 34: Assignment of node to voxel given voxel has non-zero brightness intensity

If the node overlaps a voxel which has a zero-value brightness intensity, a search box dimensioned 5x5x5 is defined around the voxel containing the node as shown in Figure 35. The voxels in this box with non-zero brightness intensity values are identified as prospective voxels. The Euclidean distance from the node of interest to the center of each of the prospective voxels is calculated. The displacement of the node of interest is assigned based on the displacement of whichever non-zero brightness intensity voxel is closest as shown in Figure 36.

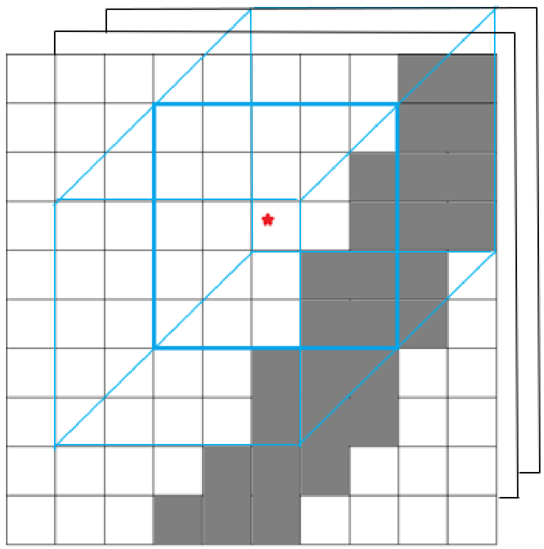


Figure 35: 5x5x5 search box for nearest voxel around node

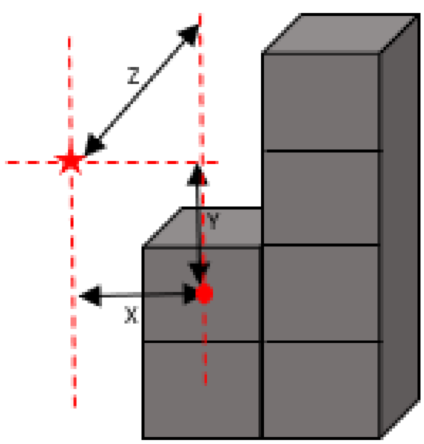
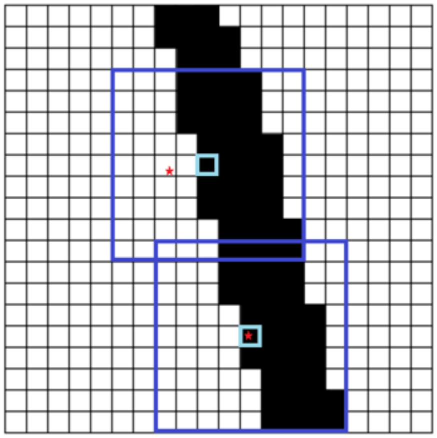


Figure 36: Assignment of node to nearest non-zero brightness intensity voxel

The displacement field of each voxel was based on a 9X9X9 voxel box average from the optical flow or demons results as shown in Figure 37. Using a box average filter was proven to reduce the higher outlier displacement errors that can be a result of using lower alpha values as the weight between brightness constancy and the smoothness constraint.



1. Nodes (red star) are assigned to their nearest voxel (light blue box)
2. Voxels are assigned displacement based on 9x9x9 box average of optical flow results (dark blue box)
3. Node assigned displacement based on new voxel displacement

Figure 37: Use of box average of voxel displacement field for nodal displacement assignments

Figure 38 shows the original location of nodes based on their coordinates (red), and the assigned locations of the nodes on their corresponding voxels (blue). The nodes that were originally on information containing voxels are overlapped by the blue nodes, only the nodes whose locations were changed can be seen in red. It can be seen that the assigned locations provided good coverage of the space occupied by the valve model.

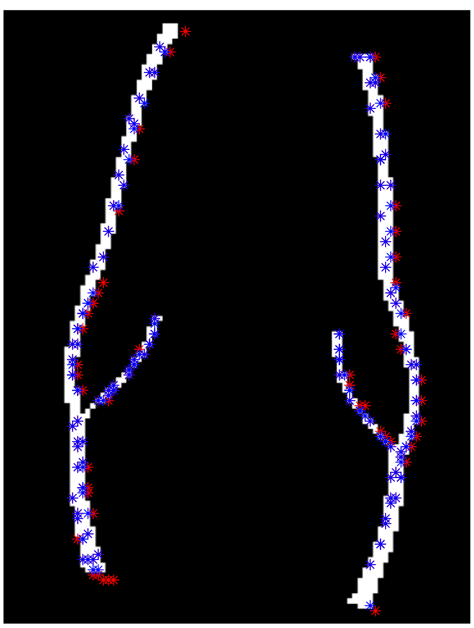


Figure 38: Original location of node coordinates superimposed on a slice of DICOM Image 1 (red), final location of nodes assigned to voxels on a slice DICOM Image 1 (blue)

3.6.1.1 *Optimization of Box Size for Averaging Displacement Results*

To reduce the effect of any outlier error within the optical flow displacement, a box averaging filter was applied to the optical flow displacement field after each of the alpha values was computed. A box filter averages the results of the voxels surrounding the voxel of interest using equal weights for each voxel. The size of the box filter was determined by an optimization process. Two metrics were used in the optimization process. First, the maximum difference, and second, the number of voxels with difference greater than 2 mm. Figure 39 shows the plot of these metrics.

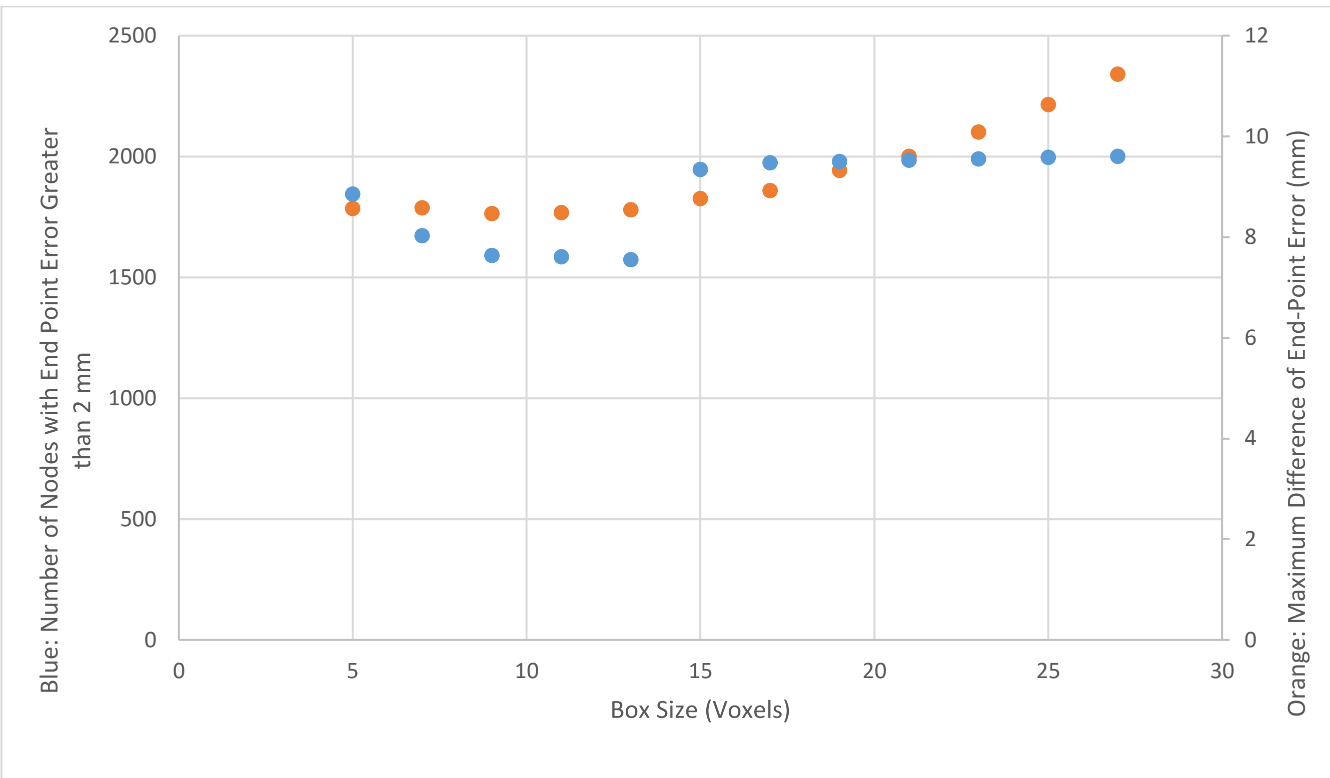


Figure 39: Minimization of error for size of box average applied to the voxel displacement field: number of nodes with an end point error greater than 2mm (blue), maximum difference of the endpoint error (orange)

3.7 Metrics for Comparison of Motion Estimation Nodal Displacements to the Ground Truth

Four metrics were used for determining agreement between the results of the motion estimation nodal displacement methods and the ground truth. The first metric was the magnitude of the difference vector between the ground truth nodal displacements and the calculated nodal displacements for each node as shown in Eq. 123, also called end-point error. The variables u, v, and w represent the displacements in the x, y, and z directions respectively. The subscript e stands for expected; these are the ground truth values. The subscript c stands for calculated; these are the values determined using the different motion estimation methods. The same variable label convention was applied to all the metrics. The second metric was the absolute error of the magnitude of the nodal displacements, shown in Eq. 124. The third metric was the relative percent difference between the magnitude of ground truth nodal displacements, and the calculated nodal displacements, shown in Eq. 125. Finally, the fourth metric was the angular error shown in Eq. 126. The displacement terms of the angular error equation, $\mathbf{v_c}$ and $\mathbf{v_e}$, are shown in Eq. 127. This equation was extended to 3D from Eq. 111. While the absolute error and relative percent difference metrics are based on the magnitude of the vectors, the angular error only considers trajectory. The magnitude of difference error metric considers both magnitude and trajectory.

$$Magnitude\ of\ Difference\ Error = \sqrt{(u_c - u_e)^2 + (v_c - v_e)^2 + (w_c - w_e)^2} \quad \text{Eq. 123}$$

$$Absolute\ Error = \left|\sqrt{{u_c}^2 + {v_c}^2 + {w_c}^2} - \sqrt{{u_e}^2 + {v_e}^2 + {w_e}^2}\right| \quad \text{Eq. 124}$$

$$Relative\ Percent\ Difference = \frac{\left|\sqrt{{u_c}^2 + {v_c}^2 + {w_c}^2} - \sqrt{{u_e}^2 + {v_e}^2 + {w_e}^2}\right|}{\frac{1}{2}\left(\sqrt{{u_c}^2 + {v_c}^2 + {w_c}^2} + \sqrt{{u_e}^2 + {v_e}^2 + {w_e}^2}\right)} \quad \text{Eq. 125}$$

$$Angular\ Error = arccos(\vec{\mathbf{v}_c} \cdot \vec{\mathbf{v}_e}) \quad \text{Eq. 126}$$

$$\vec{\mathbf{v}} = \frac{1}{\sqrt{u^2 + v^2 + w^2 + 1}}(u, v, w, 1)^T \quad \text{Eq. 127}$$

3.8 Images with Noise

Once the most effective method of motion detection was identified based on its performance on the ideal images, the effects of noise in the images were determined. In CT imaging, the typical type of noise distribution is Gaussian noise [28]. The noise in medical images is measured by a signal to noise ratio. The signal to noise ratio (SNR) of a CT scan of the ascending aorta was found to vary between 20 dB and 40 dB [29].

Noise was added to the ideal images with varying levels of SNR of 10 dB, 20 dB, 30 dB, and 40 dB. The extension of the range of SNR values to 10 dB was selected as a safe guard if the SNR was underestimated because of the otherwise perfect images. The built in MATLAB function **awgn** was be used to add noise with a Gaussian distribution to the images. Awgn stands for "Add white Gaussian noise". The function took inputs of: a 2-dimensional gray scale image, and a SNR. To add noise to a 3-dimensional image, noise at the given SNR was added to each slice of the image individually. Figure 40 shows the front cross section of the image at time step 1 with SNRs of 10dB, 20dB, 30dB, and 40dB. After noise was added to the images, the process of determining the nodal displacements was run on the new noisy images. The nodal displacements of the noisy images were compared to the nodal displacements of the noise-free images to find the level of correlation between the two.

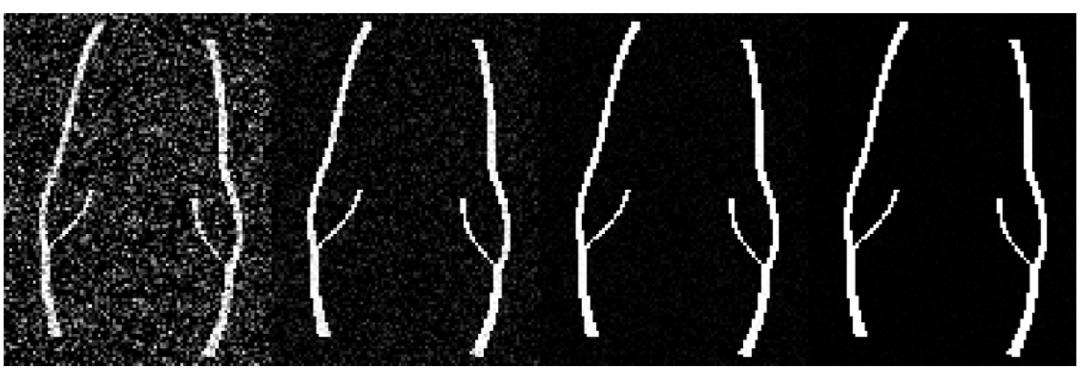


Figure 40: Cross-section view of time step 1 with different SNR levels; 10 dB, 20 dB, 30 dB, and 40 dB from left to right respectively

4 RESULTS

4.1 COMPARISON OF MOTION ESTIMATION METHODS FOR VALIDATION OF FINITE ELEMENT MODELS

The following section will present the results of the three motion estimation methods used within the process of determining the nodal displacement field of the synthetic DICOM stack over 20 timesteps. Section 4.1.1 presents the results of the parameter selection for both the Horn-Schunck optical flow method, and the Brox optical flow method. Section 4.1.2 presents the average results of the three motion estimation methods across all the time steps. It also presents a comparison of the results of the group of aorta nodes to the group of leaflet nodes for each motion estimation method. Section 4.1.3 presents the nodal results of the three motion estimation methods within the process at time steps 2, 7, and 20. These timesteps represent the range of the most challenging motion of the aortic valve over a cardiac cycle.

4.1.1 Optimal Parameters Selected for Optical Flow Motion Estimation Methods

The validation method was tested between the image at time step 1 and the images at every other timestep, 2-20, using the three motion estimation methods. For the optical flow motion estimation methods, Horn-Schunck and Brox, the parameters were tuned within the process. Two alpha values were selected during the process for Horn-Schunck optical flow: one for the valve leaflets and one for the aorta. The results of the alpha values selected for Horn-Schunck optical flow are shown in Table 1. For Brox optical flow an alpha and a gamma value were selected for the valve leaflets and an alpha and a gamma value were selected for the aorta. The results of the alpha and gamma values selected for Brox optical flow are shown in

Table 2.

Table 1: Horn-Schunck Optical Flow Alpha Value Selection

Timestep	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Leaflet	0.4	0.2	1	0.4	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2
Aorta	0.4	0.4	0.4	0.2	0.2	0.2	0.4	0.6	0.4	0.4	0.6	0.6	0.6	0.4	0.4	0.4	0.4	0.4	0.6

Table 2: Brox Optical Flow Alpha Value and Gamma Value Selection

	Timestep	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Alpha Value Selection	Leaflet	0.04	0.04	0.06	0.06	0.06	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04
	Aorta	0.08	0.1	0.1	0.08	0.1	0.08	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06
Gamma Value Selection	Leaflet	2	2	2	2	2	0.1	0.5	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1
	Aorta	0.1	2	2	0.5	2	2	2	2	1	1	1	1	1	2	2	2	1	2	2

4.1.2 Performance of Motion Estimation Methods

Figure 41 shows the median and spread of the nodal displacement magnitudes for the ground truth (top), and synthetic DICOM images using the different motion estimation methods for all 20 time steps. Table 3 shows the Spearman correlation coefficient between the absolute nodal displacements determined by the three motion estimation methods and the ground truth at each timestep. The motion estimation methods with the highest correlation are shown in bold. Figure 42 shows a graphical view of the correlation between the nodal displacements determined by the three motion estimation methods and the ground truth at time steps 2, 7, and 20. The estimated magnitude of displacement of each node is plotted against the ground truth magnitude of displacement of the same node. The red line shows the ideal trend, where the estimated displacement is equal to the ground truth displacement.

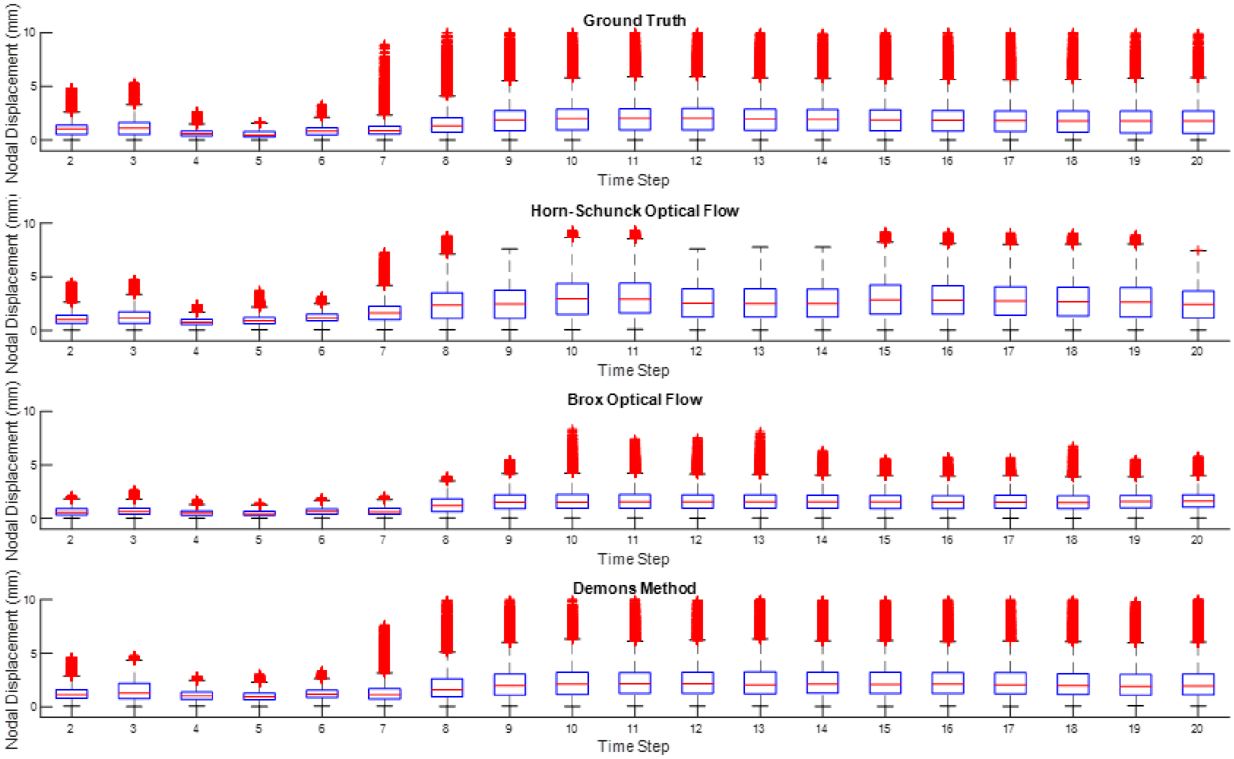


Figure 41:Boxplots showing the mean (red line), interquartile range (blue box), and outliers (red crosses) of the absolute nodal displacements from the ground truth nodal displacements, Horn-Schunck optical flow nodal displacements, Brox optical flow nodal displacements, and demons method nodal displacements, top to bottom respectively

Table 3: Spearman Correlation Coefficient between Motion Estimation Nodal Displacement and the Ground Truth Nodal Displacement

	Horn-Schunck Optical Flow		Brox Optical Flow		Demons Method	
Time Step	Correlation to Ground Truth	*P* value	Correlation to Ground Truth	*P* value	Correlation to Ground Truth	*P* value
2	**0.88**	$P < 0.05$	0.73	$P < 0.05$	0.69	$P < 0.05$
3	**0.90**	$P < 0.05$	0.73	$P < 0.05$	0.79	$P < 0.05$
4	**0.73**	$P < 0.05$	0.71	$P < 0.05$	0.43	$P < 0.05$
5	0.51	$P < 0.05$	**0.71**	$P < 0.05$	0.38	$P < 0.05$
6	0.59	$P < 0.05$	**0.80**	$P < 0.05$	0.58	$P < 0.05$
7	0.77	$P < 0.05$	0.68	$P < 0.05$	**0.87**	$P < 0.05$
8	**0.93**	$P < 0.05$	0.59	$P < 0.05$	0.90	$P < 0.05$
9	**0.96**	$P < 0.05$	0.51	$P < 0.05$	0.91	$P < 0.05$
10	**0.94**	$P < 0.05$	0.50	$P < 0.05$	0.90	$P < 0.05$
11	**0.95**	$P < 0.05$	0.48	$P < 0.05$	0.90	$P < 0.05$
12	**0.96**	$P < 0.05$	0.46	$P < 0.05$	0.90	$P < 0.05$
13	**0.96**	$P < 0.05$	0.46	$P < 0.05$	0.91	$P < 0.05$
14	**0.96**	$P < 0.05$	0.45	$P < 0.05$	0.90	$P < 0.05$
15	**0.94**	$P < 0.05$	0.42	$P < 0.05$	0.90	$P < 0.05$
16	**0.94**	$P < 0.05$	0.44	$P < 0.05$	0.90	$P < 0.05$
17	**0.94**	$P < 0.05$	0.49	$P < 0.05$	0.91	$P < 0.05$
18	**0.94**	$P < 0.05$	0.53	$P < 0.05$	0.92	$P < 0.05$
19	**0.94**	$P < 0.05$	0.57	$P < 0.05$	**0.94**	$P < 0.05$
20	**0.96**	$P < 0.05$	0.60	$P < 0.05$	0.93	$P < 0.05$

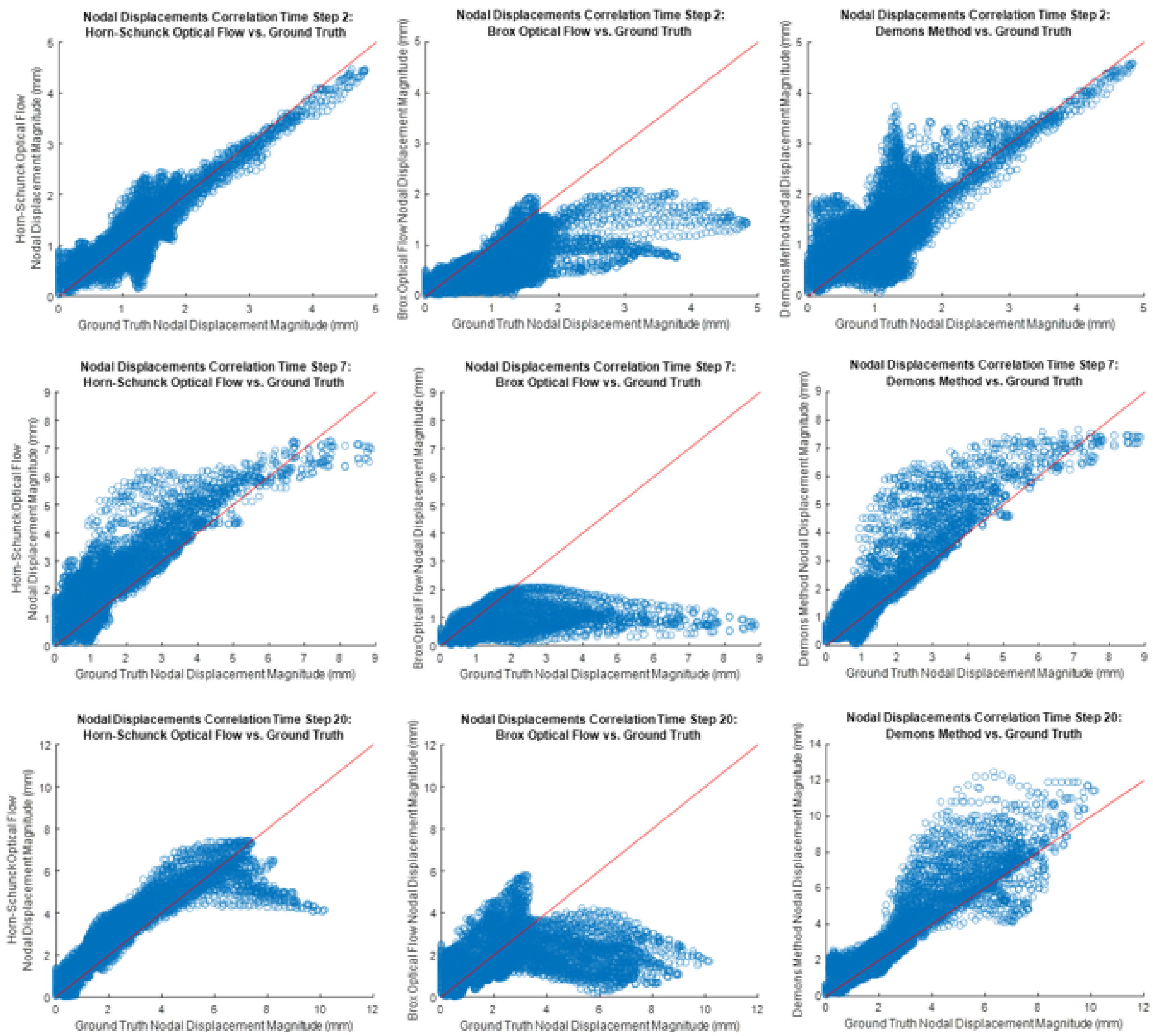


Figure 42: Correlation of Nodal Displacement Magnitudes - Motion Estimation Methods (Horn-Schunck Optical Flow, Brox Optical Flow, and demons method) vs. Ground Truth for Time Steps 2, 7, and 20

4.1.2.1 Difference Error

The magnitude of difference error given by Eq. 123 was calculated for each of the 13,359 nodes for each of the 19 different time steps. This metric considers both magnitude and trajectory of the estimated motion vectors. It shows the magnitude of the vector that is the difference between the ground truth nodal displacements and the nodal displacements found using the motion estimation techniques. The mean and standard deviation of the results for the three different methods of motion estimation: Horn-Schunck optical flow, Brox optical flow, and demons method are shown in Table 6 in Section 8.2.1. The nodes were subdivided into two categories based on their locations within the valve: aorta nodes, and leaflet nodes. The lowest mean error for each time step and for each subdivision of nodes are shown in bold. Figure 43 shows the mean and standard deviation of the magnitude of the difference vector for each time step of the valve using each method of motion estimation. The motion estimation method that produced the smallest mean difference error was Horn-Schunck optical flow for time steps 1 to 3, 9 to 14, and 20, Brox optical flow for time steps 5 and 6, and demons method for time steps 7 to 8, and 15 to 19. The standard deviations determined by Horn-Schunck optical flow were consistently smaller. The results of the subdivided valve categories: aorta nodes and leaflet nodes, are shown in Figures 44 and 45 respectively. The trend for the aorta nodes follows the overall trend. The trend of the leaflet nodes shows that the Horn-Schunck optical flow method produces the smallest mean difference error for the leaflet nodes for all time steps except time step 6.

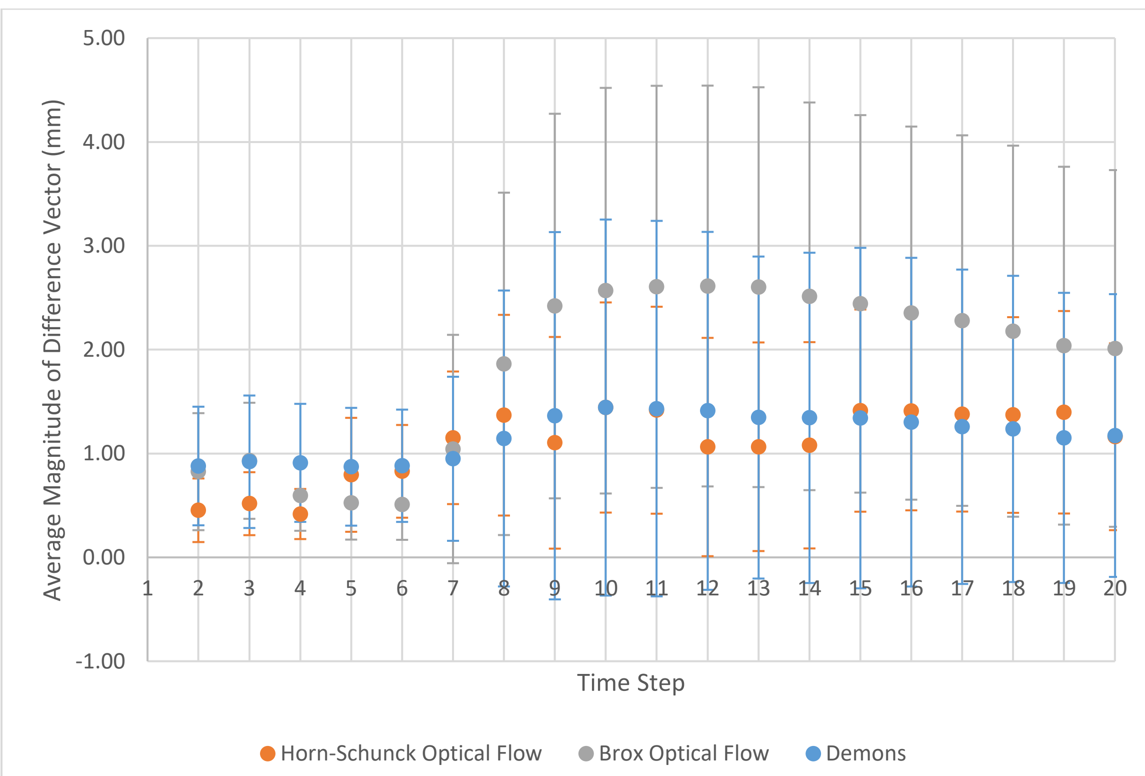


Figure 43: Average magnitude of difference vector between the ground truth nodal displacement and the calculated nodal displacement using the three motion estimation methods: Horn-Schunck optical flow, Brox optical flow, and demons method

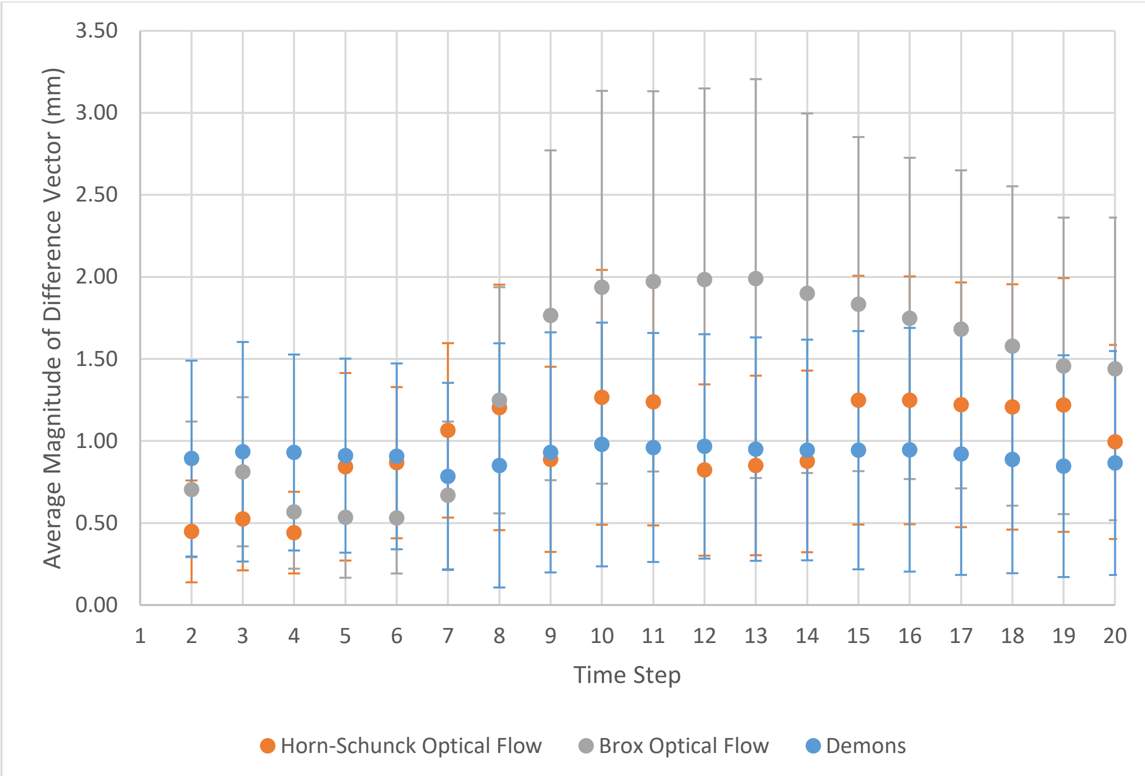


Figure 44: Average magnitude of difference vector for the aorta nodes only between the ground truth nodal displacement and the calculated nodal displacement using the three motion estimation methods: Horn-Schunck optical flow, Brox optical flow, and demons method

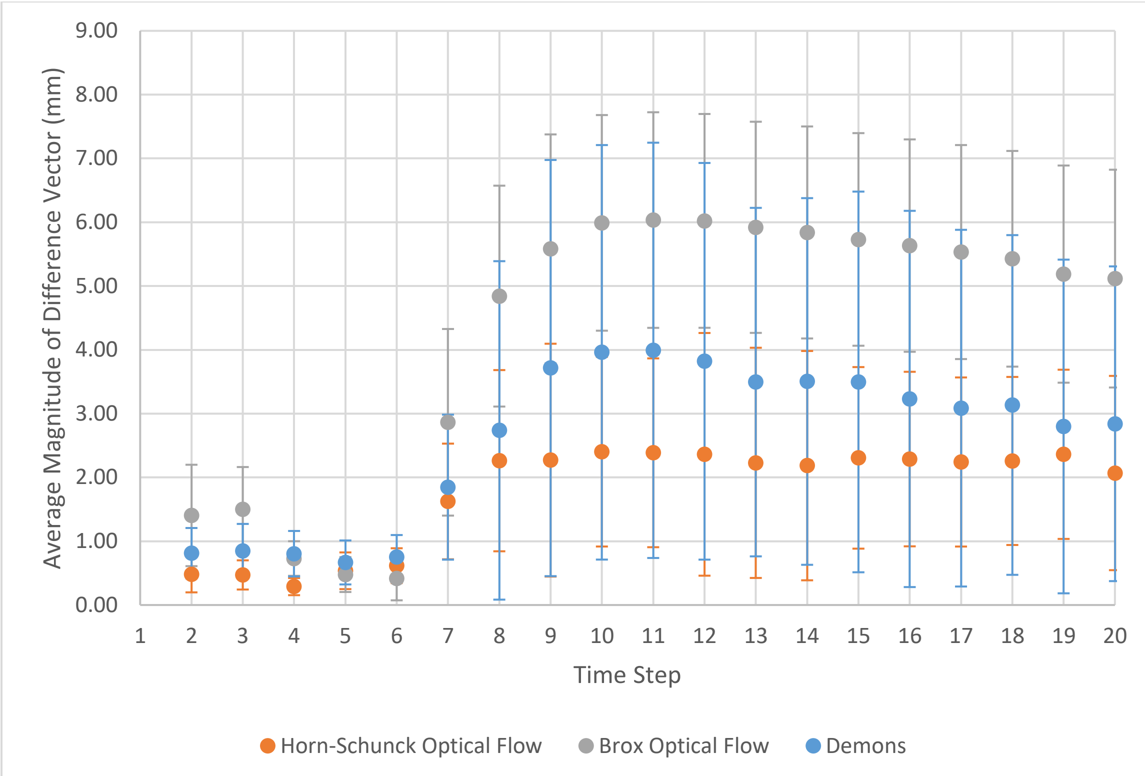


Figure 45: Average magnitude of difference vector for the leaflet nodes only between the ground truth nodal displacement and the calculated nodal displacement using the three motion estimation methods: Horn-Schunck optical flow, Brox optical flow, and demons method

4.1.2.2 Absolute Error

The absolute error given by Eq. 124 was calculated for each of the 13,359 nodes for each of the 19 different time steps. This metric considers only the magnitude of the estimated motion vectors. It shows the absolute error between the magnitude of the ground truth nodal displacements and the magnitude of the nodal displacements found using the motion estimation techniques. The mean and standard deviation of the results for the three different methods of motion estimation: Horn-Schunck optical flow, Brox optical flow, and demons method are shown in Table 7 in Section 8.2.2. The nodes were subdivided into two categories based on their location within the valve; aorta nodes, and leaflet nodes. The lowest mean error for each time step and for each subdivision of nodes are shown in bold. Figure 46 shows the mean and standard deviation of the absolute error for each time step of the valve using each method of motion estimation. The motion estimation method that produced the smallest mean absolute error was Horn-Schunck optical flow for time steps 1 to 3, Brox optical flow for time steps 4 and 5, and demons method for time steps 7 to 20. The standard deviations determined by Brox optical flow were larger than both demons method, and Horn-Schunck optical flow. The results of the subdivided node categories: aorta nodes and leaflet nodes, are shown in Figures 47 and 48 respectively. The trend for the aorta nodes follows the overall trend. The trend of the leaflet nodes is different from the overall trend in that the motion estimation method that produced the smallest mean absolute error for the leaflet nodes was Horn-Schunck optical flow for all the time steps except 5 and 6.

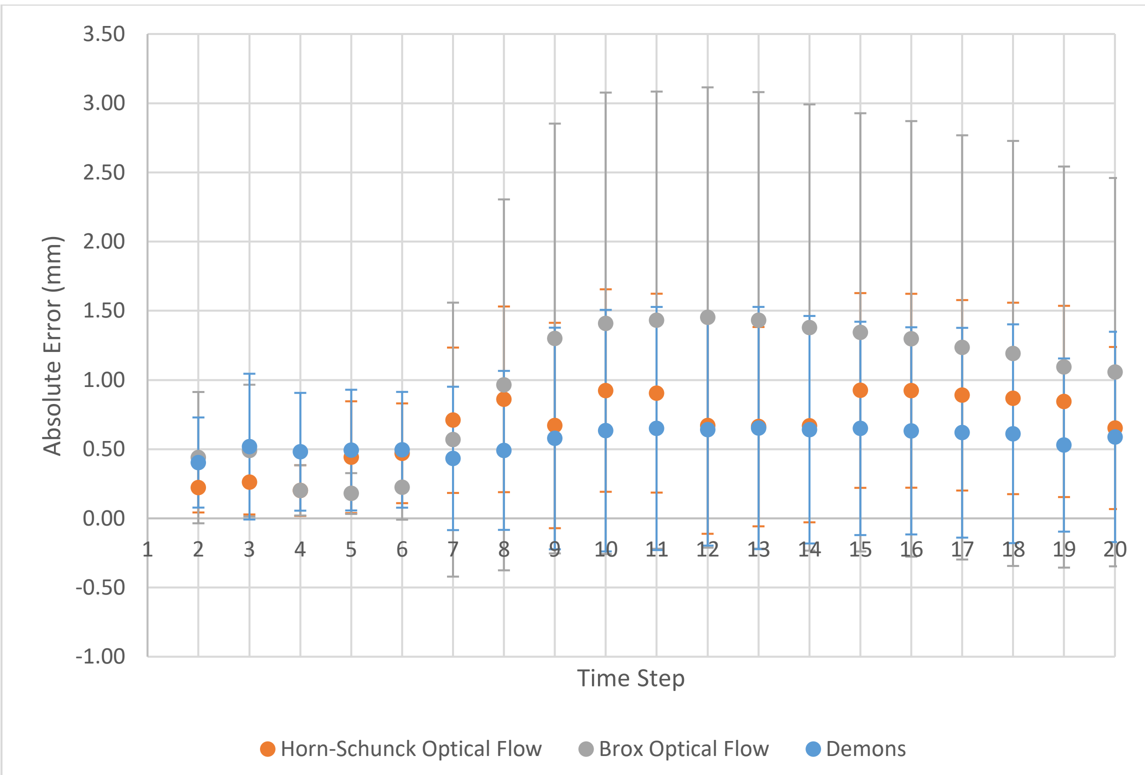


Figure 46: Average absolute error between the ground truth nodal displacement and the calculated nodal displacement using the three motion estimation methods: Horn-Schunck optical flow, Brox optical flow, and demons method

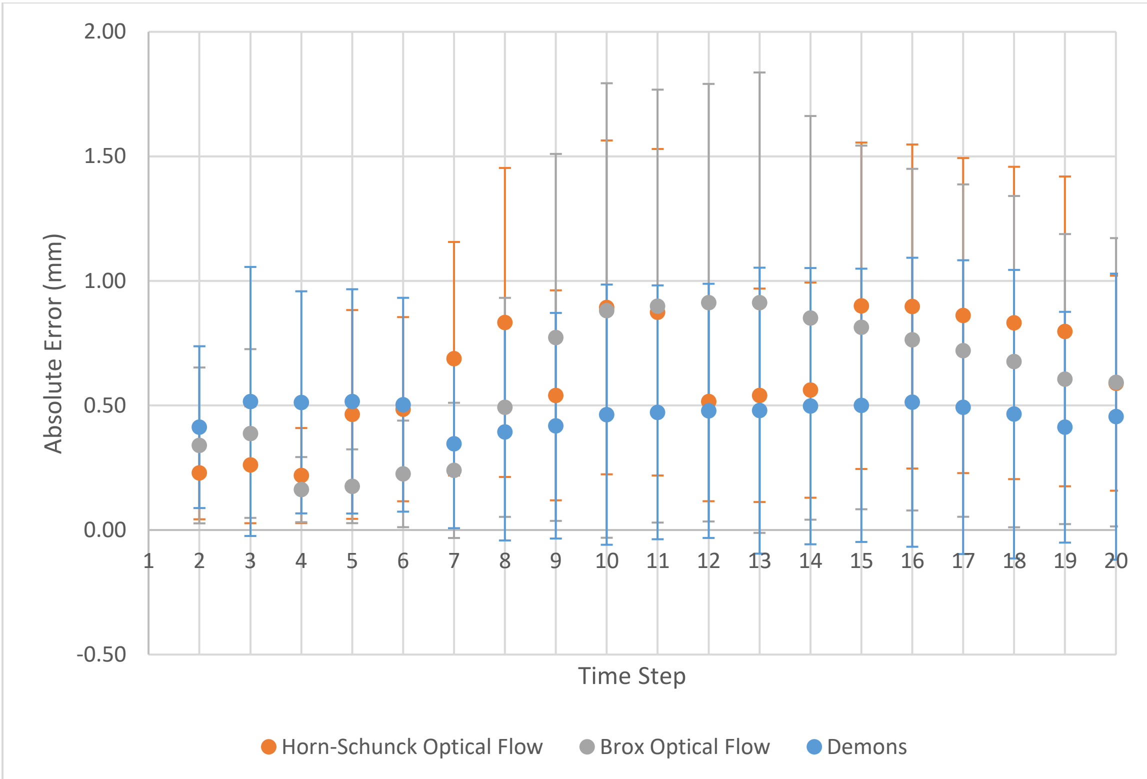


Figure 47: Average absolute error for the aorta nodes between the ground truth nodal displacement and the calculated nodal displacement using the three motion estimation methods: Horn-Schunck optical flow, Brox optical flow, and demons method

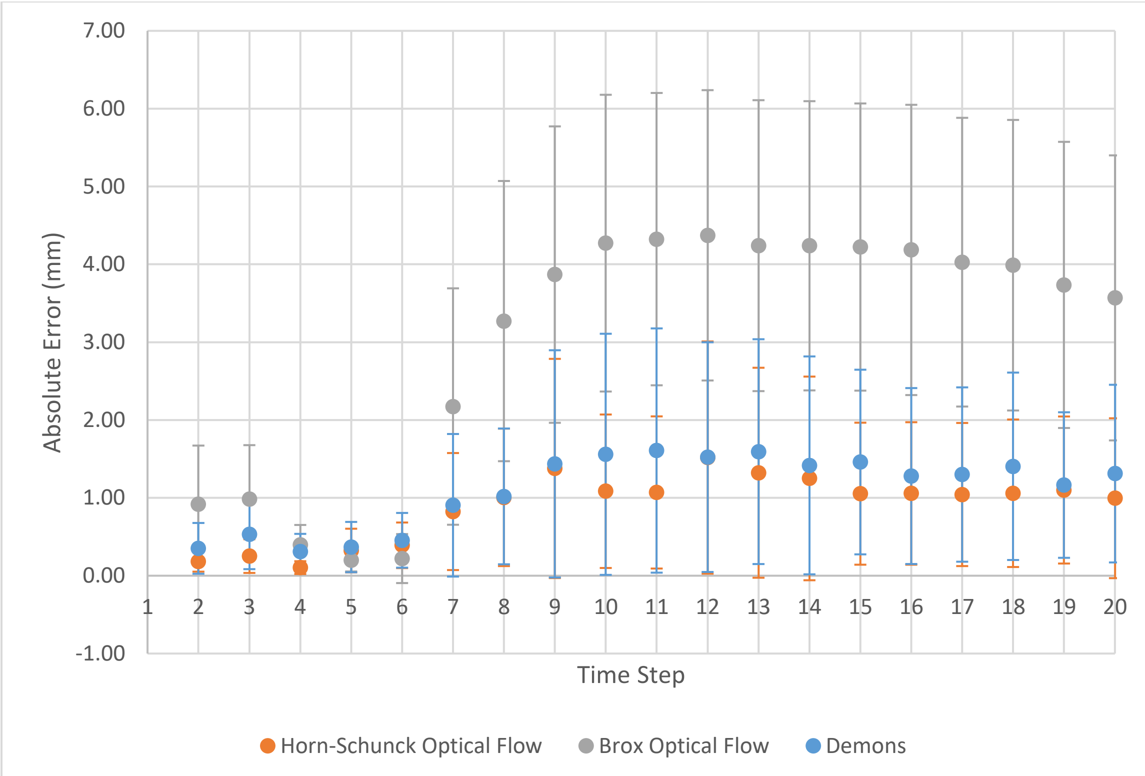


Figure 48: Average absolute error for the leaflet nodes between the ground truth nodal displacement and the calculated nodal displacement using the three motion estimation methods: Horn-Schunck optical flow, Brox optical flow, and demons method

4.1.2.3 Relative Percent Difference

The relative percent difference given by Eq. 125 was calculated for each of the 13,359 nodes for each of the 19 different time steps. This metric considers only the magnitude of the estimated motion vectors. It shows the relative percent difference between the magnitude of the ground truth nodal displacements and the magnitude of the nodal displacements found using the motion estimation techniques. The mean and standard deviation of the results for the three different methods of motion estimation: Horn-Schunck optical flow, Brox optical flow, and demons method are shown in Table 8 in Section 8.2.3. The nodes were subdivided into two categories based on their location within the valve; aorta nodes, and leaflet nodes. The lowest mean error for each time step and for each subdivision of nodes are shown in bold. Figure 49 shows the mean and standard deviation of the relative percent difference for each time step of the valve using each method of motion estimation. The motion estimation method that produced the smallest mean relative percent difference was Horn-Schunck optical flow for time steps 1 to 3 and 12, Brox optical flow for time steps 5 and 6, and demons method for time steps 7 to 11, and 13 to 20. The results of the subdivided node categories: aorta nodes and leaflet nodes, are shown in Figures 50 and 51 respectively. The trend for the aorta nodes follows the overall trend. The trend of the leaflet nodes is different from the overall trend in that the motion estimation method that produced the smallest mean relative percent difference for the leaflet nodes was Horn-Schunck optical flow for time steps; 1 to 3, 7, 9 to 20.

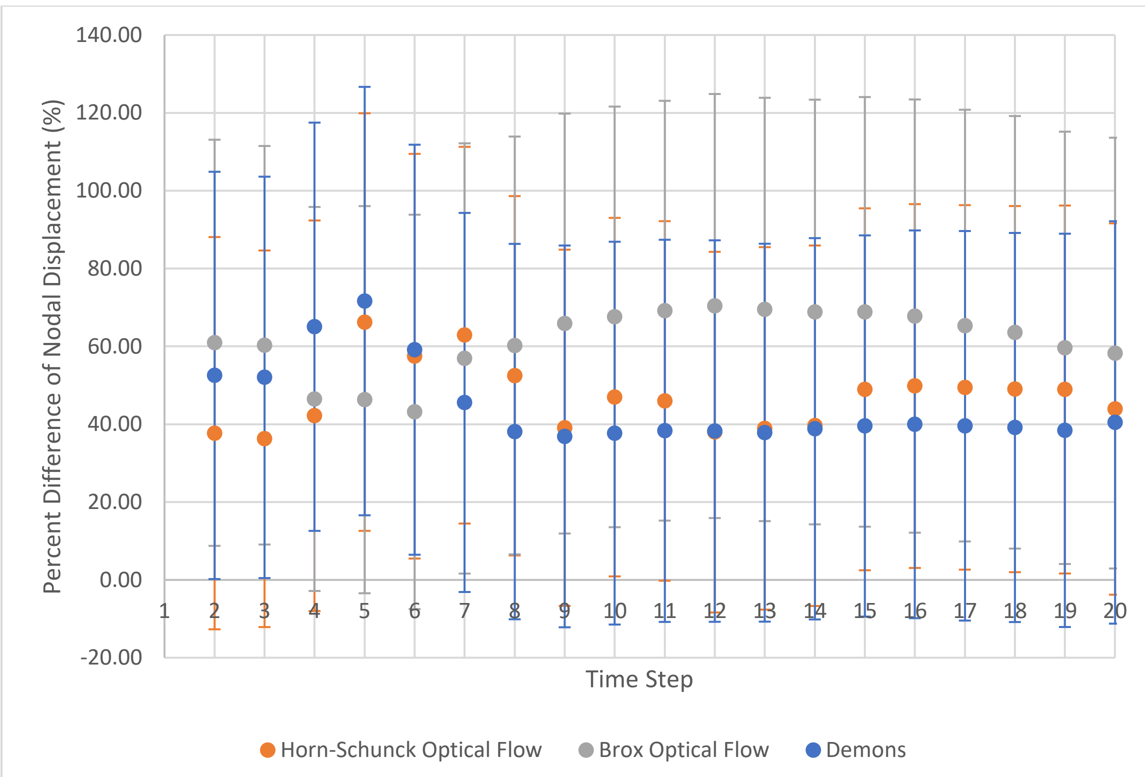


Figure 49: Average relative percent difference between the ground truth nodal displacement and the calculated nodal displacement using the three motion estimation methods: Horn-Schunck optical flow, Brox optical flow, and demons method

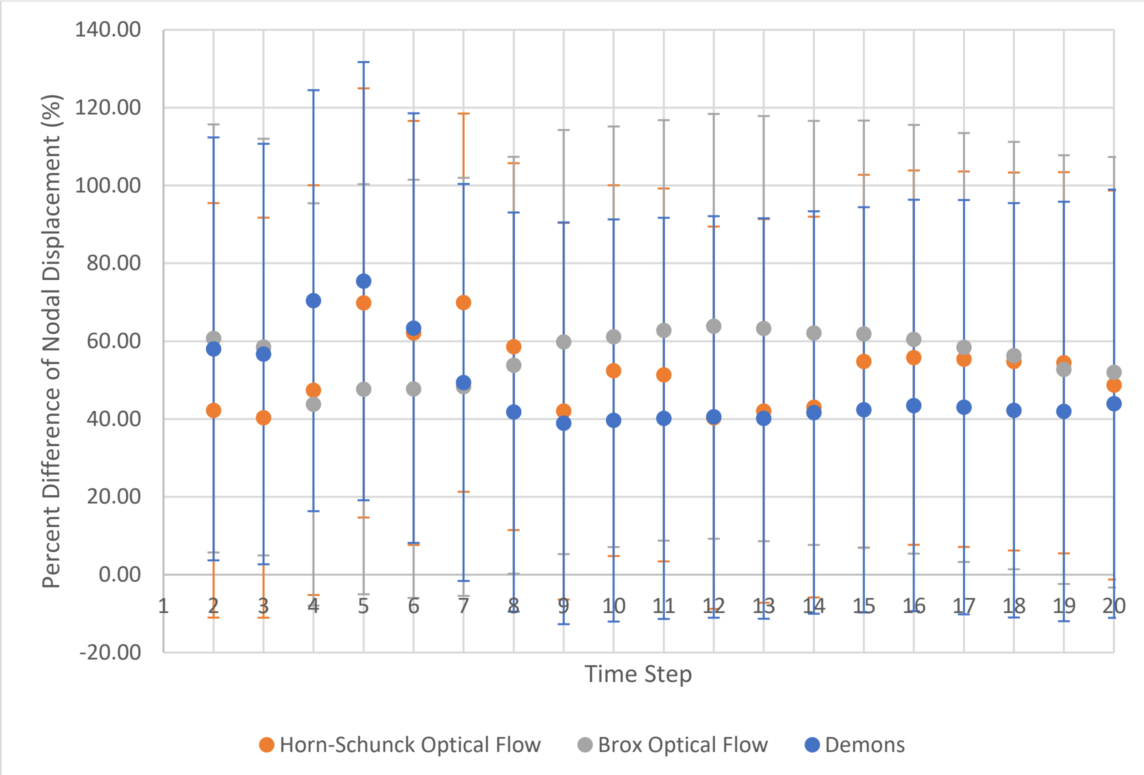


Figure 50: Average relative percent difference for the aorta nodes between the ground truth nodal displacement and the calculated nodal displacement using the three motion estimation methods: Horn-Schunck optical flow, Brox optical flow, and demons method

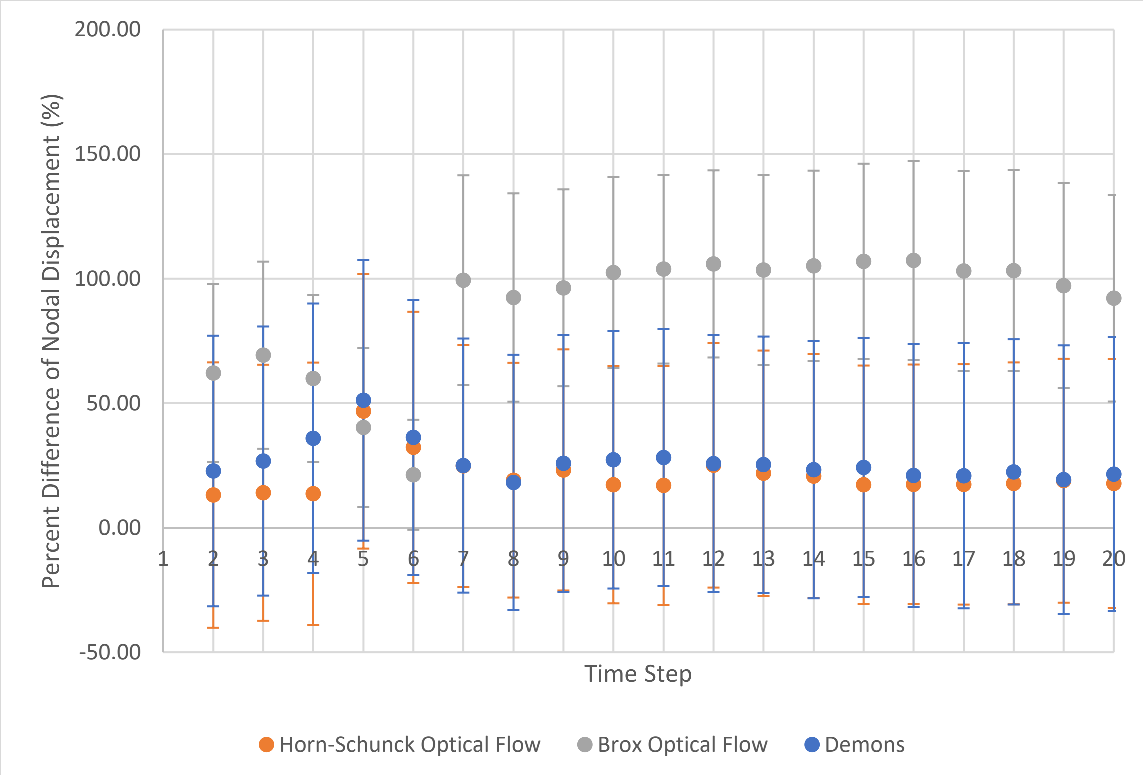


Figure 51: Average relative percent difference for the leaflet nodes between the ground truth nodal displacement and the calculated nodal displacement using the three motion estimation methods: Horn-Schunck optical flow, Brox optical flow, and demons method

4.1.2.4 Angular Error

The angular error given by Eq. 126 was calculated for each of the 13,359 nodes for each of the 19 different time steps. This metric considers only the trajectory of the estimated motion vectors. It shows the angular error between the ground truth nodal displacements and the nodal displacements found using the motion estimation techniques. The mean and standard deviation of the results for the three different methods of motion estimation: Horn-Schunck optical flow, Brox optical flow, and demons method are shown in Table 9 in Section 8.2.4. The nodes were subdivided into two categories based on their location within the valve; aorta nodes, and leaflet nodes. The lowest mean error for each time step and for each subdivision of nodes are shown in bold. Figure 52 shows the mean and standard deviation of the angular error for each time step of the valve using each method of motion estimation. The motion estimation method that produced the smallest mean angular error was Horn-Schunck optical flow for time steps 1 to 3, 9 to 17, and 20, Brox optical flow for time steps 5 and 6, and demons method for time steps 7 to 8, and 18 to 19. The results of the subdivided node categories: aorta nodes and leaflet nodes, are shown in Figures 53 and 54 respectively. The trend for the aorta nodes follows the overall trend. The trend of the leaflet nodes is different from the overall trend in that the motion estimation method that produced the smallest mean angular error for the leaflet nodes was Horn-Schunck optical flow for all the time steps except one.

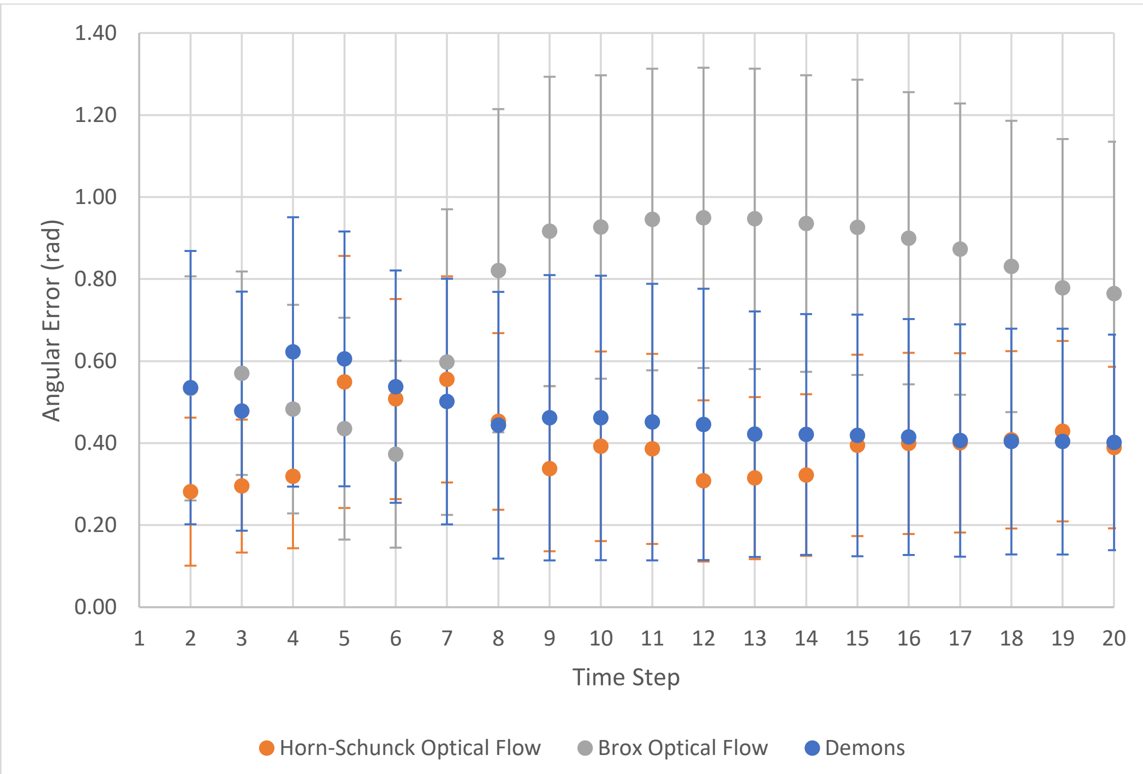


Figure 52: Average angular error between the ground truth nodal displacement and the calculated nodal displacement using the three motion estimation methods: Horn-Schunck optical flow, Brox optical flow, and demons method

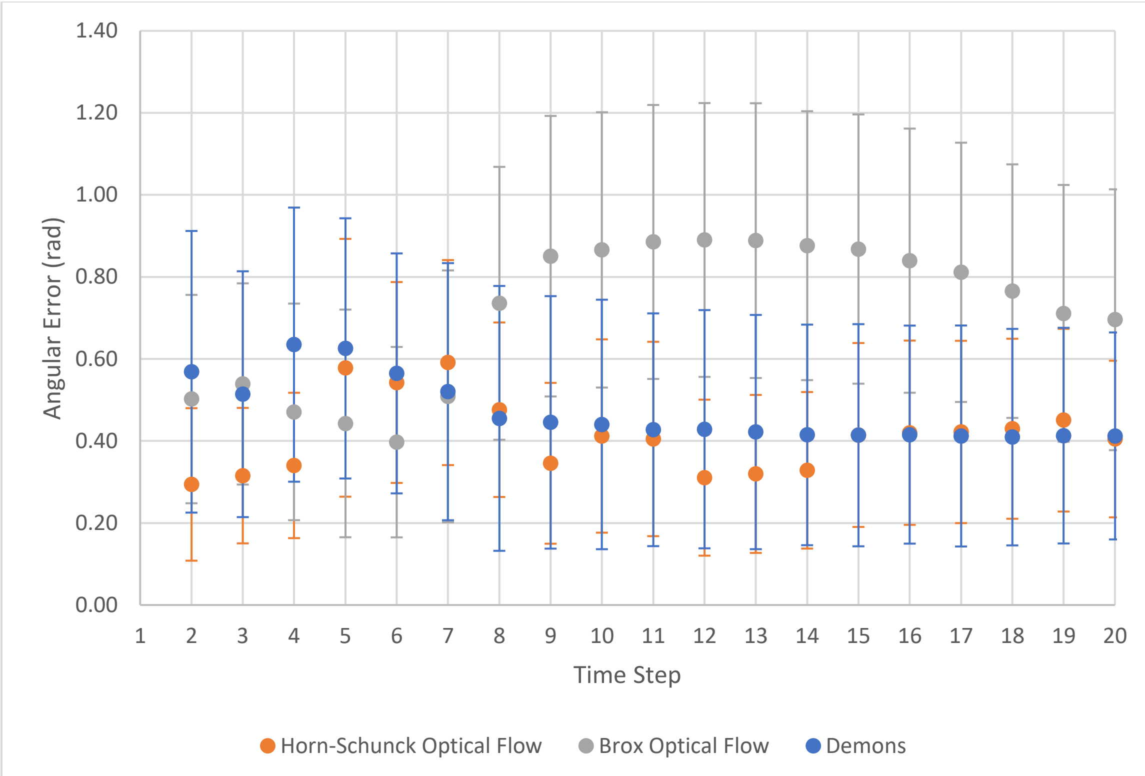


Figure 53: Average angular error for the aorta nodes between the ground truth nodal displacement and the calculated nodal displacement using the three motion estimation methods: Horn-Schunck optical flow, Brox optical flow, and demons method

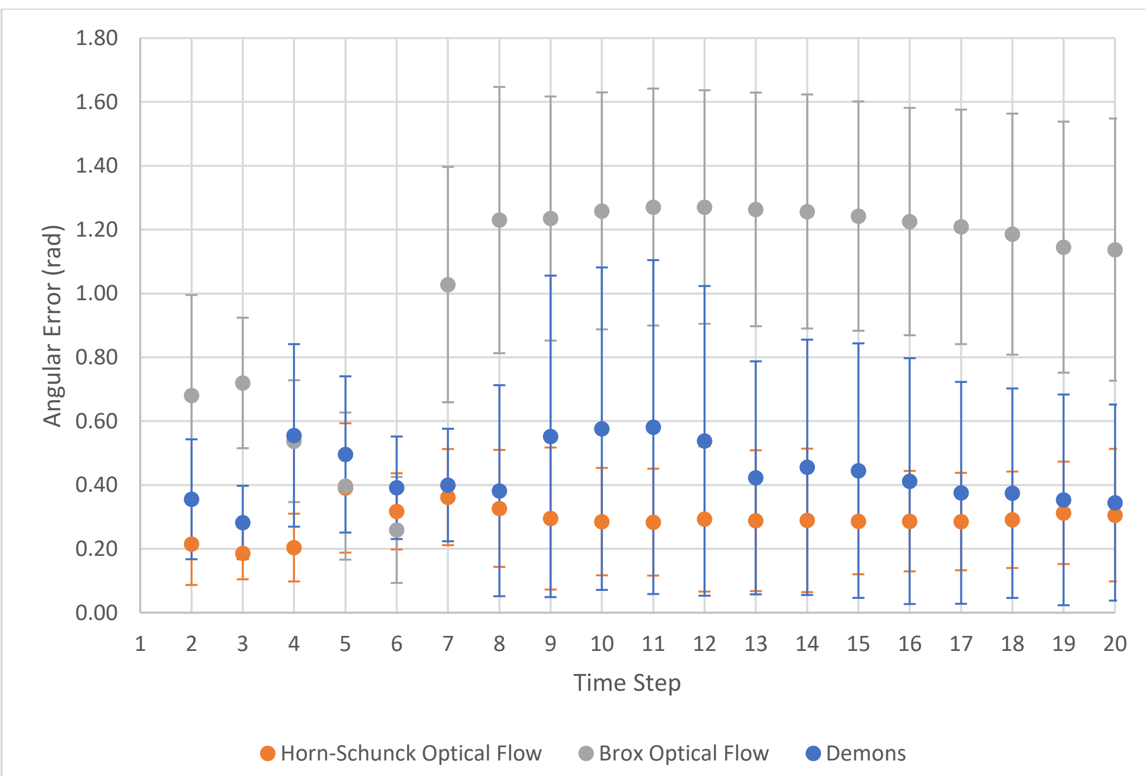


Figure 54: Average angular error for the leaflet nodes between the ground truth nodal displacement and the calculated nodal displacement using the three motion estimation methods: Horn-Schunck optical flow, Brox optical flow, and demons method

4.1.3 Performance of Motion Estimation Methods between Specific Steps

The results of the motion estimation between timestep 1 and timesteps 2, 7, and 20 are shown in detail to the nodal level below. These timesteps were selected as they represent the spectrum of the motion the valve undergoes through the cardiac cycle. The motion between timestep 2 and timestep 5 occurred while the valve was still open. The magnitudes of nodal displacement were small relative to when the valve closes. The motion at time steps 6 and 7 occurred when the valve was in the midst of closing. The magnitudes of the nodal displacements were greater than in the open valve, but less than in the closed valve. The motion at timesteps 10 through 20 occurred at the later stages of the valve closing. The magnitude of nodal displacements between the first timestep and these timesteps were the greatest.

4.1.3.1 Time step 2

The results of the process using the three different motion estimation methods between the images at timestep 1 and timestep 2 are shown in Figures 55 to 60. These images show the detailed results of how the individual nodes of the aortic valve behave with respect to the metrics defined in Eq. 123 to Eq. 126. Figure 55 shows a qualitative and illustrative representation of how well each motion estimation algorithm preformed on the leaflets of the valve between the image at timestep 1 and the image at timestep 2 using the selected parameters for the leaflets from Table 1 and

Table 2 . The montage of nine images on the left are the front cross sections, and the montage of nine images on the right are the side cross sections. Within the montages, the three images on the left-hand side are the cross sections at time step 1, and the three images on the right-hand side are the cross sections at time step 2. The images in the center are the warped imaged using the three different methods of motion estimation: Horn-Schunck optical flow, Brox optical flow, and demons method, from the top row to the bottom row, respectively. The warped images apply the displacement field determined by each of the motion estimation algorithms to the image at timestep 2. Ideally, if the motion estimation is perfect, the warped image should look identical to the image at timestep 1. It can

be seen that the displacement fields generated from both the demons method, and Horn-Schunck optical flow effectively capture the motion of the leaflets of the valve, while the displacement field generated by Brox optical flow does not. This is supported by the results of Figure 56. Figure 56 shows the histograms of the difference error between the nodal displacements determined by the motion estimation methods and the ground truth nodal displacements given by Eq. 123. This metric considers both the magnitude and trajectory of the nodal displacements. The three histograms from left to right each represent the results from a different method of motion estimation: Horn-Schunck optical flow, Brox optical flow, and demons method, respectively. The Horn-Schunck optical flow has a maximum error of 1.8 mm, the Brox optical flow has a maximum error of 3.5 mm, and the demons method had a maximum error of 3 mm. Figure 57 shows the locations of the 100 most problematic nodes on the aortic valve for the three motion estimation methods; Horn-Schunck optical flow, Brox optical flow, and demons method, from left to right respectively. The highlighted nodes from the Horn-Schunck optical flow method show nodes with a difference in magnitude of absolute error greater than 1.5 mm. The highlighted nodes from the Brox optical flow method show nodes with a difference in magnitude of absolute error greater than 3.3 mm. The highlighted nodes from the demons method show nodes with a difference in magnitude of absolute error greater than 2.7 mm. The other metrics for agreeability between the determined nodal displacements and the ground truth nodal displacements are shown in Figures 58 to 60. Figure 58 shows the histograms of the absolute error between the magnitude of the nodal displacements determined by the motion estimation methods and the ground truth nodal displacement given by Eq. 124. This metric considers only the magnitude of the nodal displacements. Figure 59 shows the histograms of the relative percent difference between the magnitude of the nodal displacements determined by the motion estimation methods and the ground truth nodal displacement given by Eq. 125. This metric also only considers the magnitude of the nodal displacements. Figure 60 shows the histograms of the angular error between nodal displacements determined by motion estimation methods and the ground truth nodal displacements given by Eq. 126. This metric only considers the trajectory of the nodal displacements.

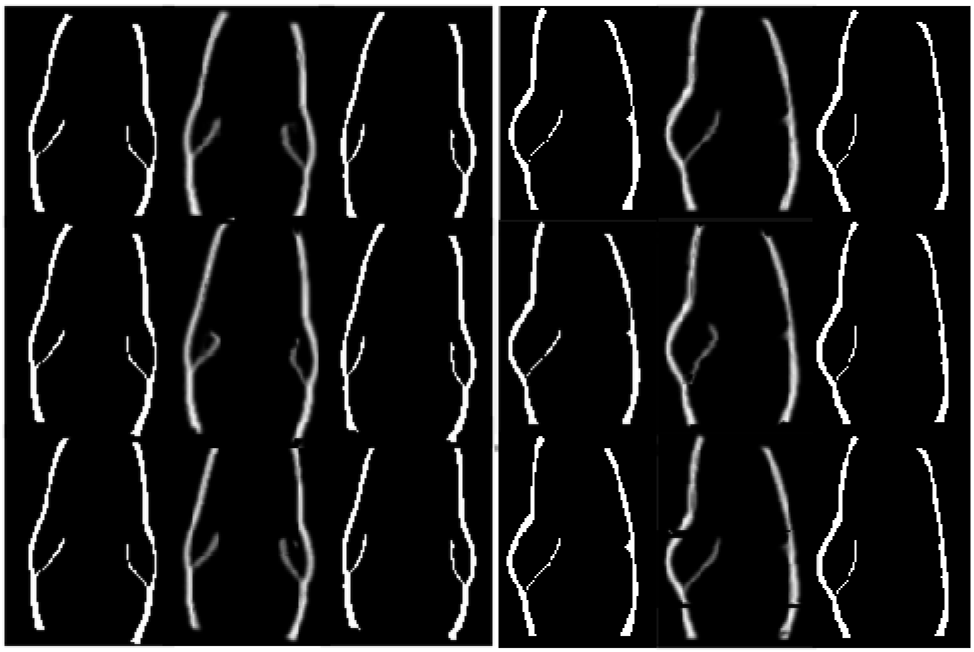


Figure 55: Front view (left) and side view (right) of images warped from timestep 2 image to timestep 1 image using three motion estimation methods: left - timestep 1 image, center – warped image (2 to 1), right – timestep 2 image, top – Horn-Schunck optical flow, center – Brox optical flow, bottom – demons method

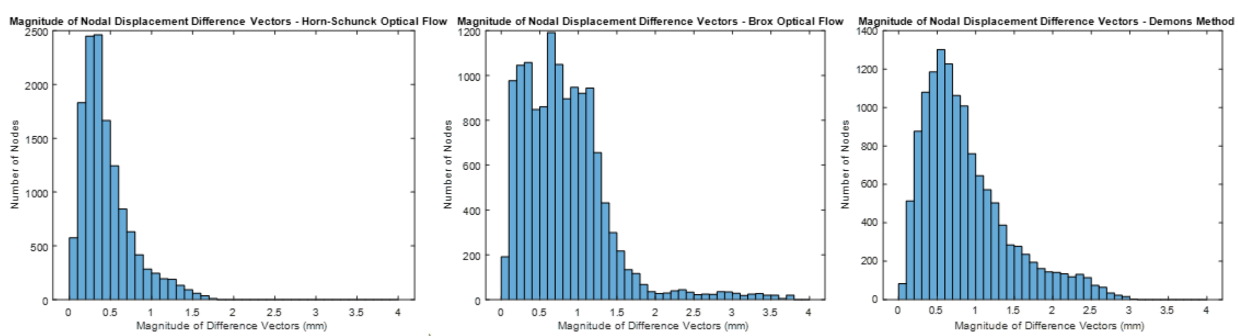


Figure 56: Difference error of nodal displacements between time steps 1 and 2 of three motion estimation methods – Horn-Schunck optical flow (left), Brox optical flow (middle), demons method (right)

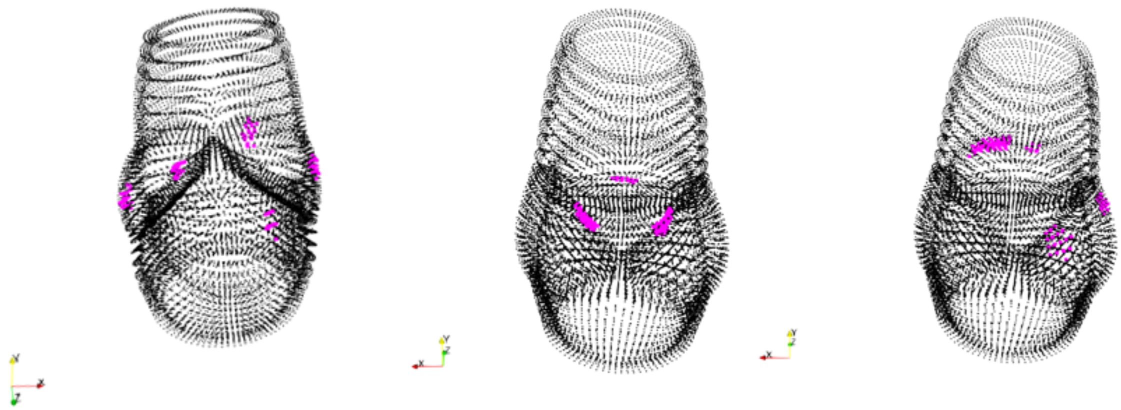


Figure 57: Locations of 100 nodes with the highest absolute error of nodal displacement magnitudes between time steps 1 and 2 of three motion estimation methods - Horn-Schunck optical flow (left), Brox optical flow (middle), demons method (right)

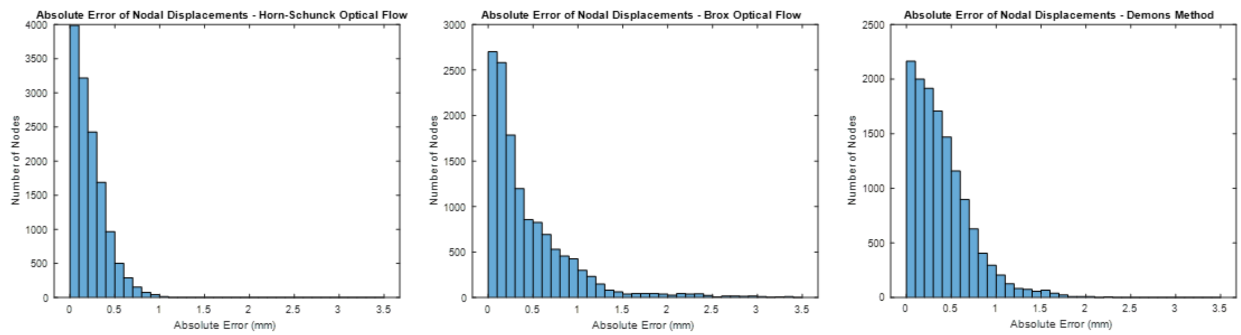


Figure 58: Absolute error of nodal displacement magnitudes between time steps 1 and 2 of three motion estimation methods - Horn-Schunck optical flow (left), Brox optical flow (middle), demons method (right)

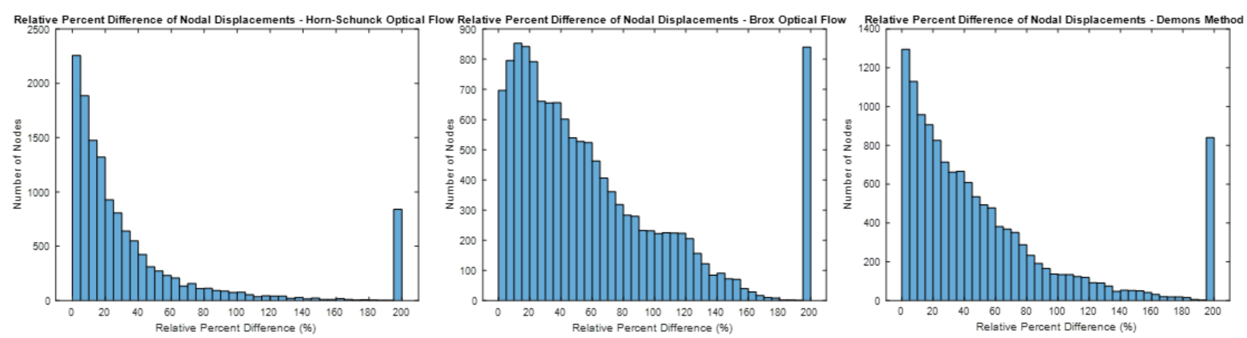


Figure 59: Relative percent difference of nodal displacement magnitudes between time steps 1 and 2 of three motion estimation methods - Horn-Schunck optical flow (left), Brox optical flow (middle), demons method (right)

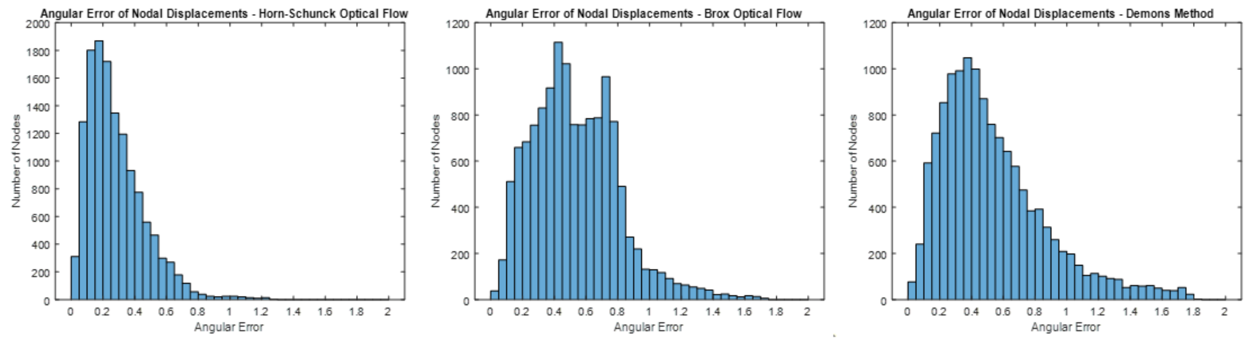


Figure 60: Angular error (radians) of nodal displacements between time steps 1 and 2 of three motion estimation methods - Horn-Schunck optical flow (left), Brox optical flow (middle), demons method (right)

4.1.3.2 Time step 7

The results of the process between the images at timestep 1 and timestep 7 are shown in Figures 61 to 66. Figure 61 shows a qualitative and illustrative representation of how well each motion estimation algorithm preformed on the leaflets of the valve between the image at timestep 1 and the image at timestep 7 using the selected parameters for the leaflets from Table 1 and

Table 2. The layout is the same as in Figure 55. It can be seen that only the Horn-Schunck motion estimation method is able to capture the motion of the valve leaflets. Figure 62 shows the histograms of the difference error between the nodal displacements determined by the motion estimation methods and the ground truth nodal displacements given by Eq. 123. This metric considers both the magnitude and trajectory of the nodal displacements. The three histograms from left to right each represent the results from different methods of motion estimation: Horn-Schunck optical flow, Brox optical flow, and demons method, respectively. The Horn-Schunck optical flow has a maximum error of 4.5 mm, the Brox optical flow has a maximum error of 9 mm, and the demons method has a maximum error of 5 mm. Figure 63 shows the locations of the 100 most problematic nodes on the aortic valve for the three motion estimation methods; Horn-Schunck optical flow, Brox optical flow, and demons method, from left to right respectively. The highlighted nodes from the Horn-Schunck optical flow method show nodes with a difference in magnitude of absolute error greater than 3.4 mm. The highlighted nodes from the Brox optical flow method show nodes with a difference in magnitude of absolute error greater than 6.0

mm. The highlighted nodes from the demons method show nodes with a difference in magnitude of absolute error greater than 4.3 mm. The other metrics for agreeability between the determined nodal displacements and the ground truth nodal displacements are shown in Figures 64 to 66. Figure 64 shows the histograms of the absolute error between the magnitude of the nodal displacements determined by the motion estimation methods and the ground truth nodal displacement given by Eq. 124. This metric considers only the magnitude of the nodal displacements. Figure 65 shows the histograms of the relative percent difference between the magnitude of the nodal displacements determined by the motion estimation methods and the ground truth nodal displacement given by Eq. 125. This metric also only considers the magnitude of the nodal displacements. Figure 66 shows the histograms of the angular error between nodal displacements determined by motion estimation methods and the ground truth nodal displacements given by Eq. 126. This metric only considers the trajectory of the nodal displacements.

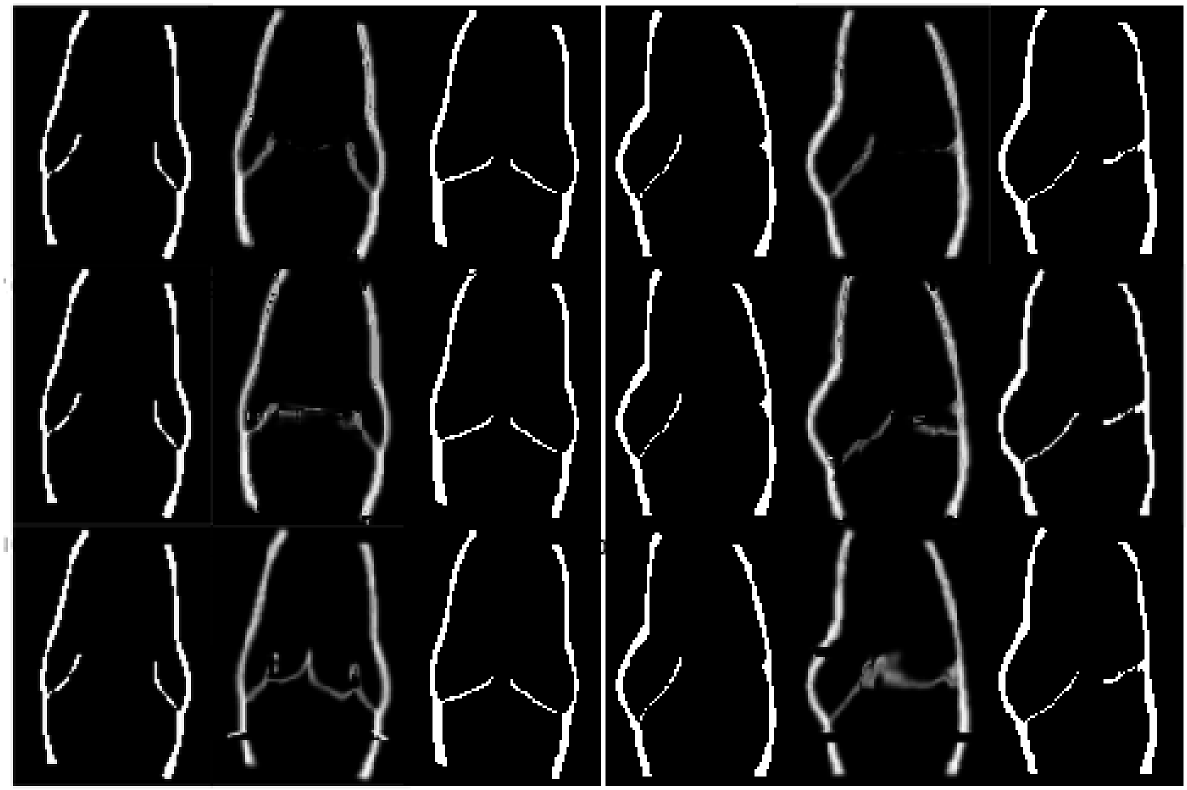


Figure 61: Front and side views of images warped from timestep 7 image to timestep 1 image using three motion estimation methods: left - timestep 1 image, center – warped image (7 to 1), right – timestep 7 image, top – Horn-Schunck optical flow, center – Brox optical flow, bottom – demons method

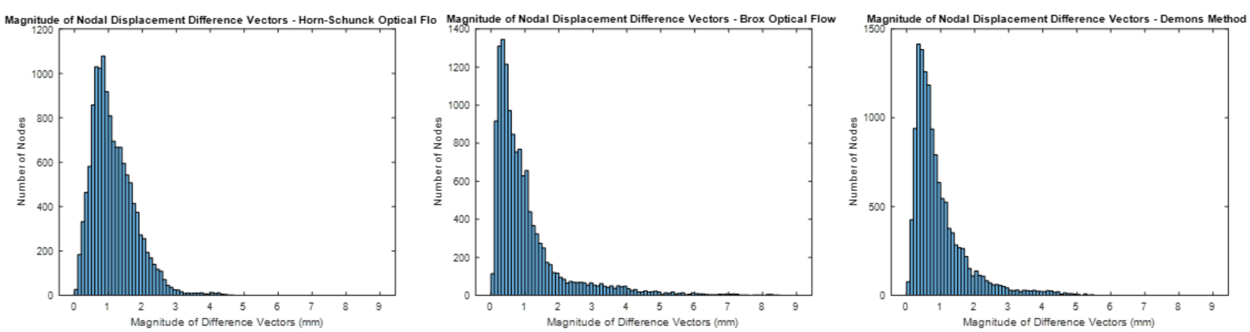


Figure 62: Difference error of nodal displacements between time steps 1 and 7 of three motion estimation methods - Horn-Schunck optical flow (left), Brox optical flow (middle), demons method (right)

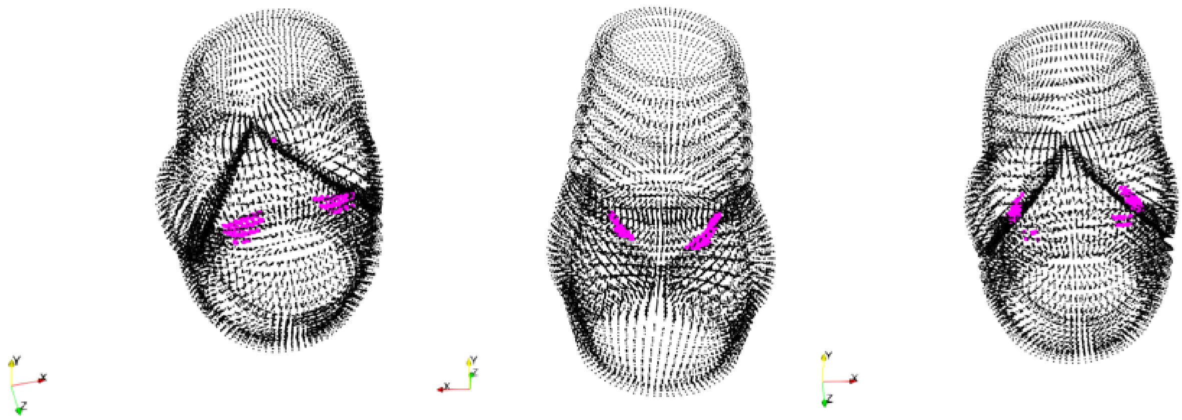


Figure 63: Locations of 100 nodes with the highest absolute error of nodal displacement magnitudes between time steps 1 and 7 of three motion estimation methods - Horn-Schunck optical flow (left), Brox optical flow (middle), demons method (right)

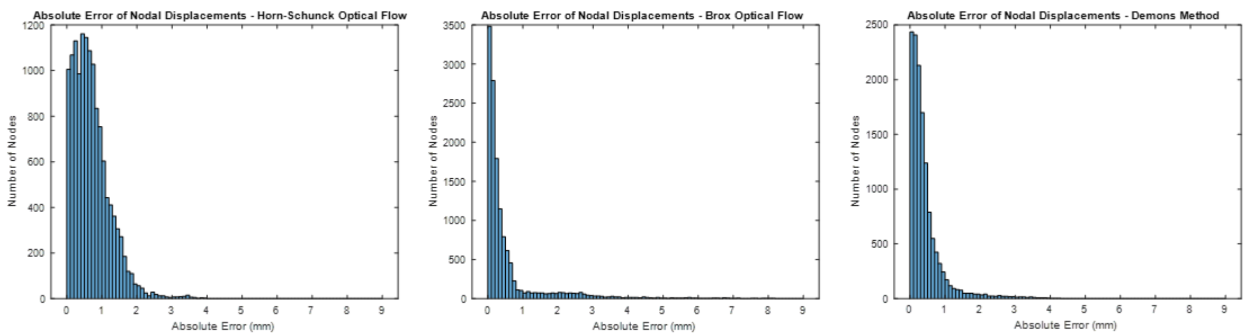


Figure 64: Absolute error of nodal displacement magnitudes between time steps 1 and 7 of three motion estimation methods - Horn-Schunck optical flow (left), Brox optical flow (middle), demons method (right)

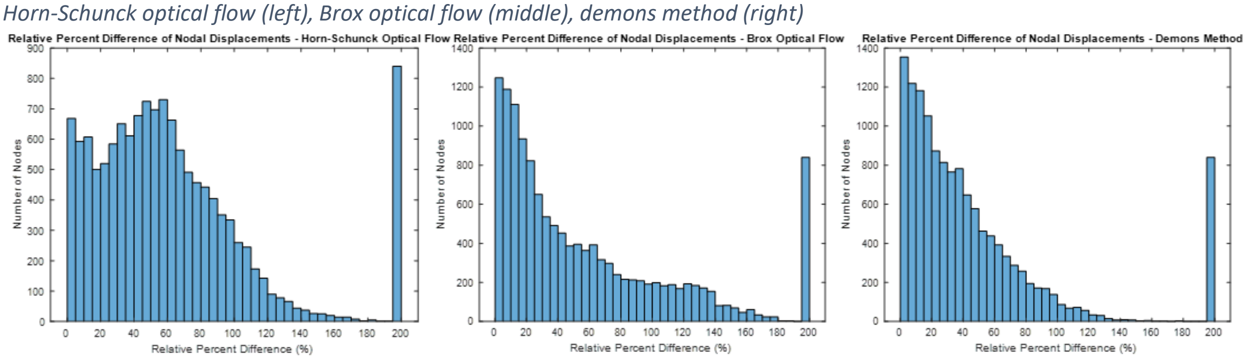


Figure 65: Relative percent difference of nodal displacement magnitudes between time steps 1 and 7 of three motion estimation methods - Horn-Schunck optical flow (left), Brox optical flow (middle), demons method (right)

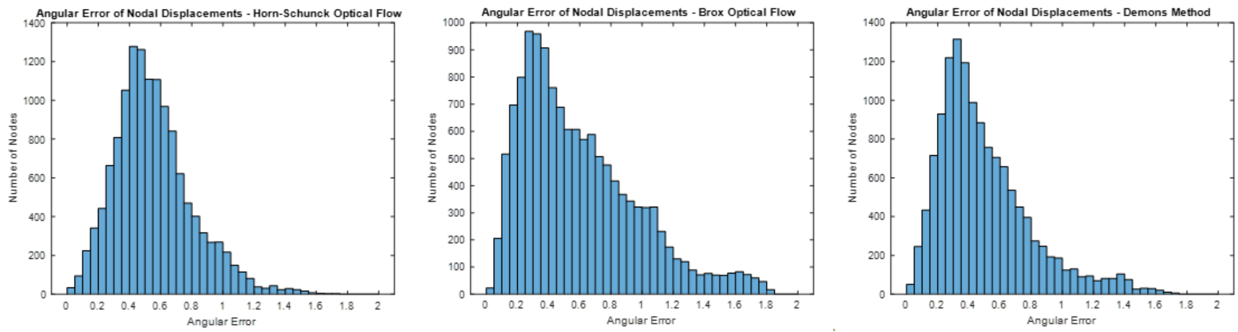


Figure 66: Angular error of nodal displacements between time steps 1 and 7 of three motion estimation methods - Horn-Schunck optical flow (left), Brox optical flow (middle), demons method (right)

4.1.3.3 Time Step 20

The results of the process between the images at timestep 1 and timestep 20 are shown in Figures 67 to 72. Figure 67 shows a qualitative and illustrative representation of how well each motion estimation algorithm preformed on the leaflets of the valve between the image at timestep 1 and the image at timestep 20 using the selected parameters for the leaflets from Table 1 and

Table 2. The layout is the same as in Figure 55. It can be seen that the only the Horn-Schunck motion estimation method is able to capture the motion of the valve leaflets. Figure 68 shows the histograms of the difference error between the nodal displacements determined by the motion estimation algorithms and the ground truth nodal displacements given by Eq. 123. The three histograms from left to right each represent the results from different methods of motion estimation: Horn-Schunck optical flow, Brox optical flow, and demons method respectively. The Horn-Schunck optical flow has a maximum error of 9 mm, the Brox optical flow has a maximum error of 9 mm, and the demons method has a maximum error of 12 mm. Figure 69 shows the locations of the 100 most problematic nodes on the aortic valve for the three motion estimation methods; Horn-Schunck optical flow, Brox optical flow, and demons method, from left to right respectively. The highlighted nodes from the Horn-Schunck optical flow method show nodes with a difference in magnitude of absolute error greater than 5.7 mm. The highlighted nodes from the Brox optical flow method show nodes with a difference in magnitude of absolute error greater than 7.9 mm. The highlighted nodes from the demons method show nodes with a difference in

magnitude of absolute error greater than 9.2 mm. The other metrics for agreeability between the determined nodal displacements and the ground truth nodal displacements are shown in Figures 70 to 72. Figure 70 shows the histograms of the absolute error between the magnitude of the nodal displacements determined by the motion estimation methods and the ground truth nodal displacement given by Eq. 124. This metric considers only the magnitude of the nodal displacements. Figure 71 shows the histograms of the relative percent difference between the magnitude of the nodal displacements determined by the motion estimation methods and the ground truth nodal displacement given by Eq. 125. This metric also only considers the magnitude of the nodal displacements. Figure 72 shows the histograms of the angular error between nodal displacements determined by motion estimation methods and the ground truth nodal displacements given by Eq. 126. This metric only considers the trajectory of the nodal displacements.

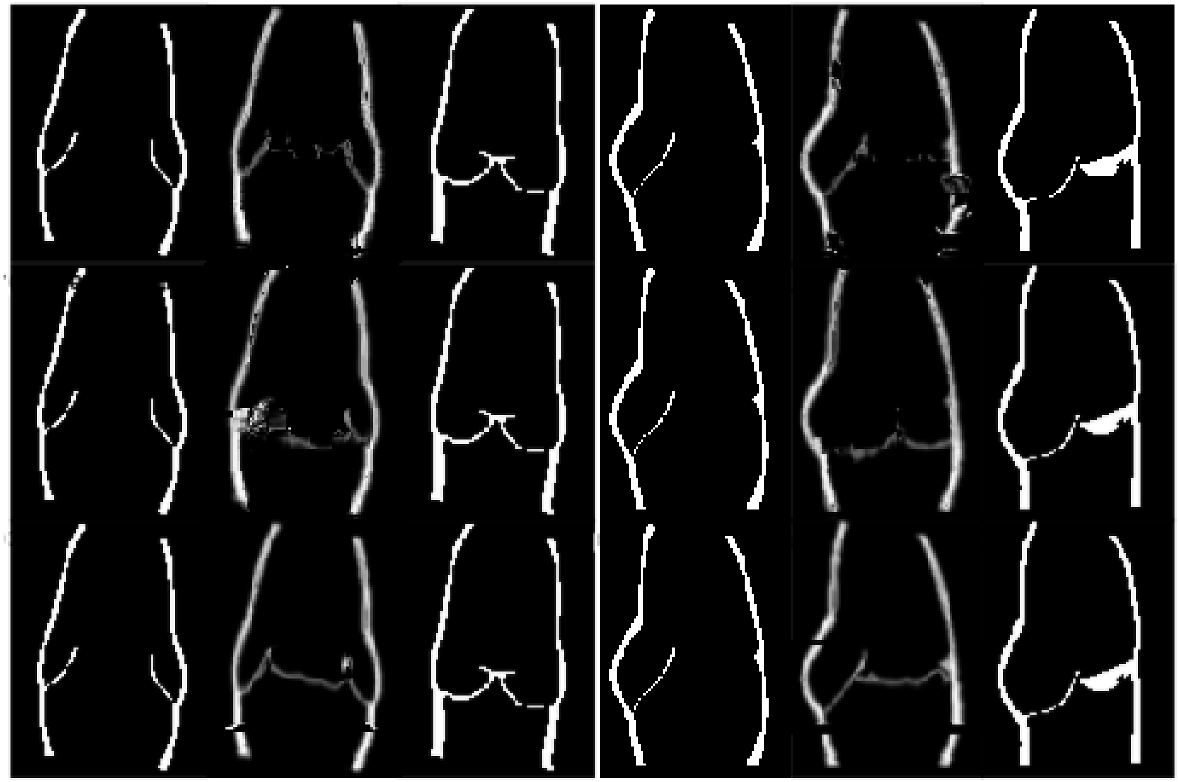


Figure 67: Front and side views of images warped from timestep 20 image to timestep 1 image using three motion estimation methods: left - timestep 1 image, center – warped image (20 to 1), right – timestep 20 image, top – Horn-Schunck optical flow, center – Brox optical flow, bottom – demons method

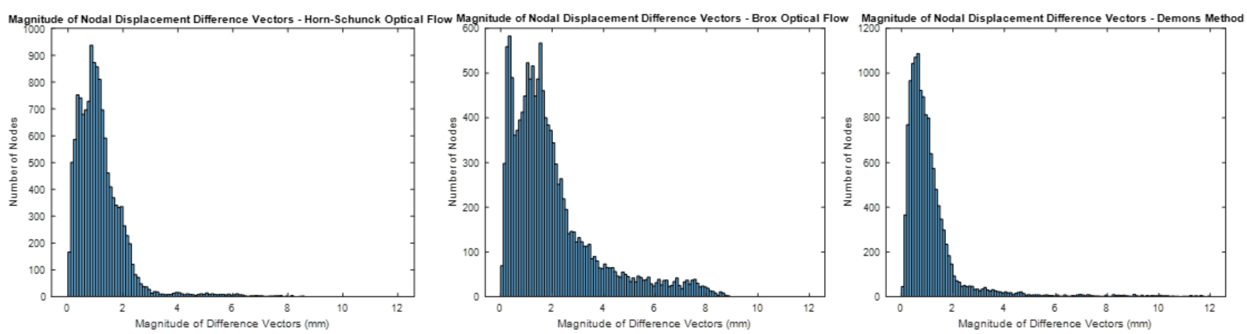


Figure 68: Difference error of nodal displacements between time steps 1 and 20 of three motion estimation methods - Horn-Schunck optical flow (left), Brox optical flow (middle), demons method (right)

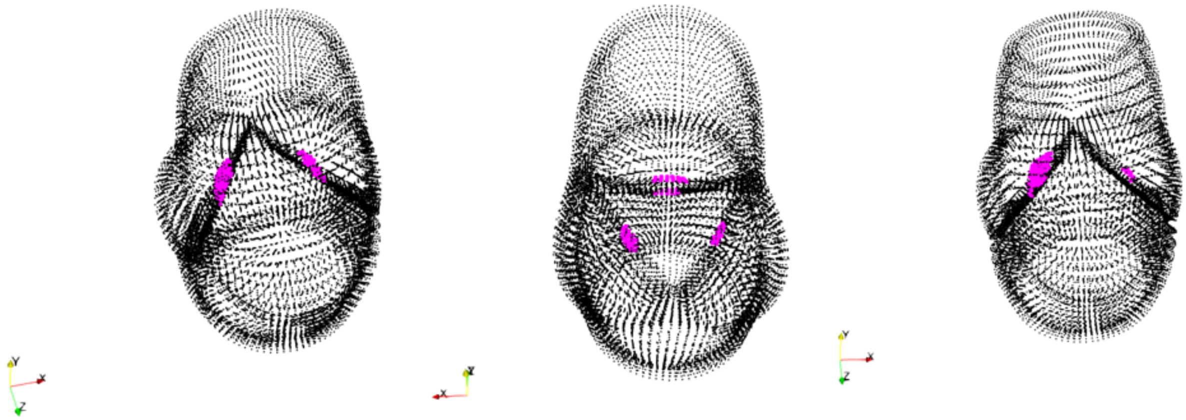


Figure 69: Locations of 100 nodes with the highest absolute error of nodal displacement magnitudes between time steps 1 and 20 of three motion estimation methods - Horn-Schunck optical flow (left), Brox optical flow (middle), demons method (right)

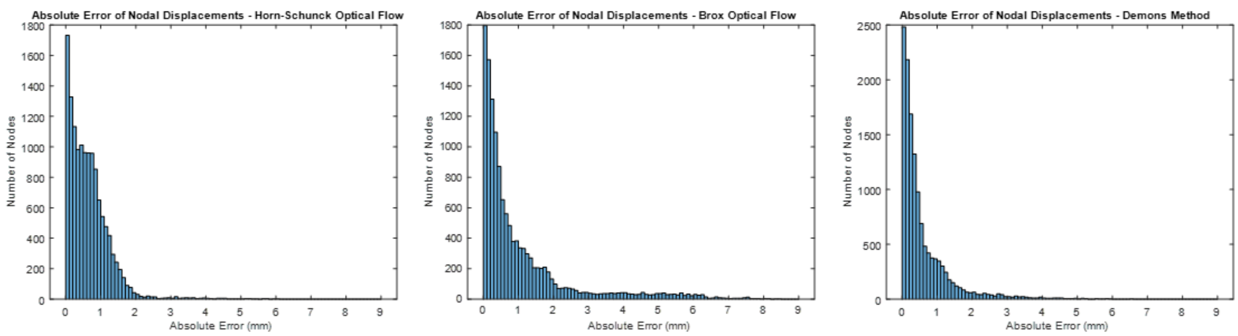


Figure 70: Absolute error of nodal displacement magnitudes between time steps 1 and 20 of three motion estimation methods - Horn-Schunck optical flow (left), Brox optical flow (middle), demons method (right)

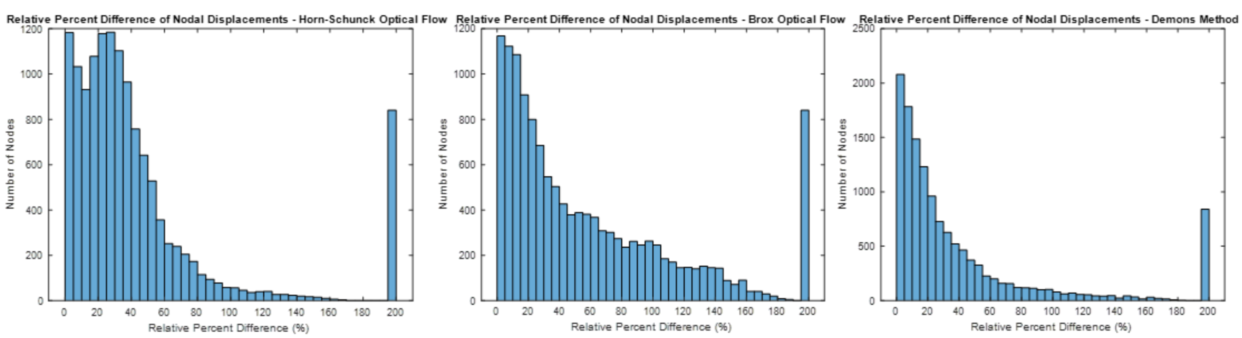


Figure 71: Relative Percent difference of nodal displacement magnitudes between time steps 1 and 20 of three motion estimation methods - Horn-Schunck optical flow (left), Brox optical flow (middle), demons method (right)

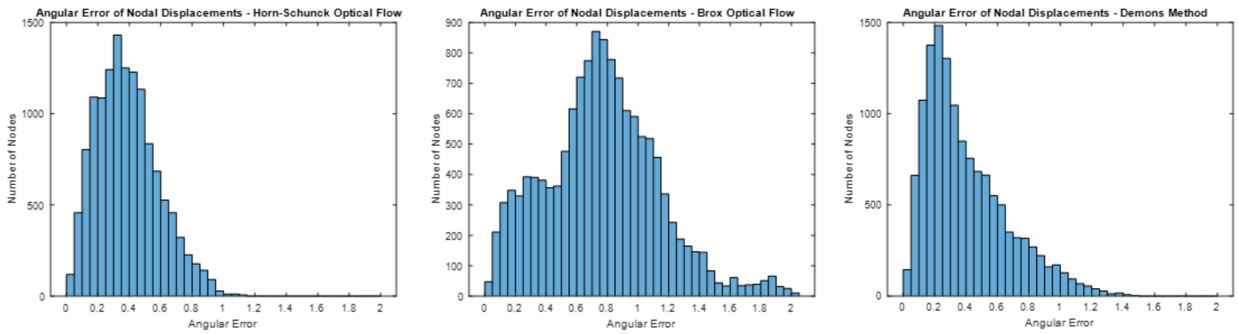


Figure 72: Angular error of nodal displacements between time steps 1 and 20 of three motion estimation methods - Horn-Schunck optical flow (left), Brox optical flow (middle), demons method (right)

4.2 Effectiveness of Horn-Schunck Optical Flow

Horn-Schunck optical flow showed the best results of the three motion estimation methods implemented. In addition to the error metrics shown above, the following section shows the comparison of the reconstructed aortic valve using Horn-Schunck optical flow, and the effect of added noise on the analysis.

4.2.1 Geometric Office Area Comparison

Using the nodal displacements determined by Horn-Schunck optical flow, new sets of nodal coordinates for the aortic valve were determined for time steps 2-20. The new sets of nodal coordinates were used to create a new STL files of the valve at time steps 2-20 which could be compared to the ground truth STL files. Figures 73 to 75 show an overlay of the ground truth STL files (green), and the Horn-Schunck optical flow STL files (purple) for time steps 2, 7, and 20 respectively. Each figure shows 3 cross sections: front view, side view, and top view from left to right.

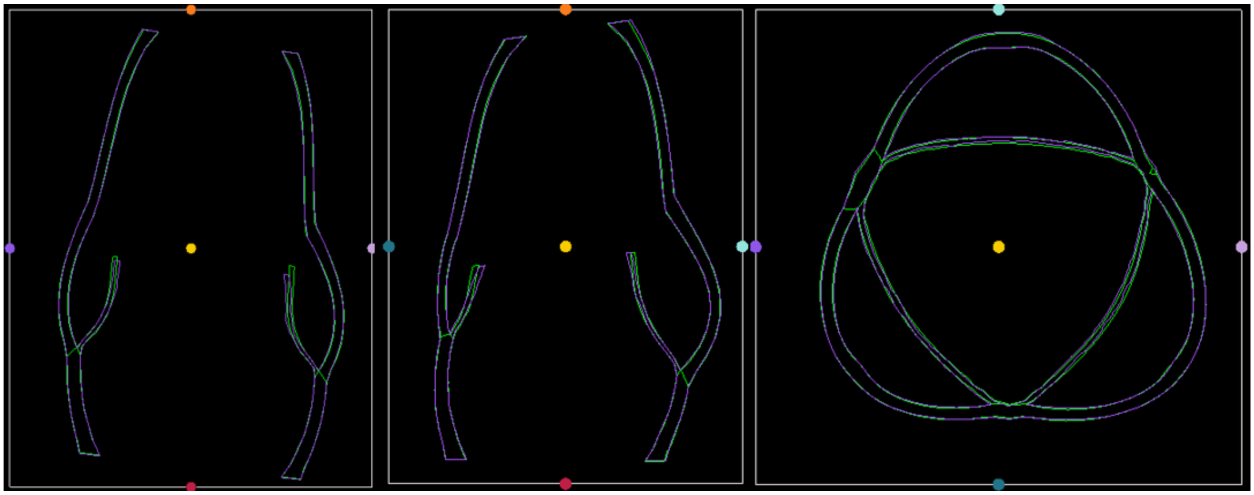


Figure 73: Cross sections of the ground truth STL file overlapped with the STL file created using the Horn-Schuck optical flow nodal displacements for time step 2: front view (left), side view (center), top view (right)

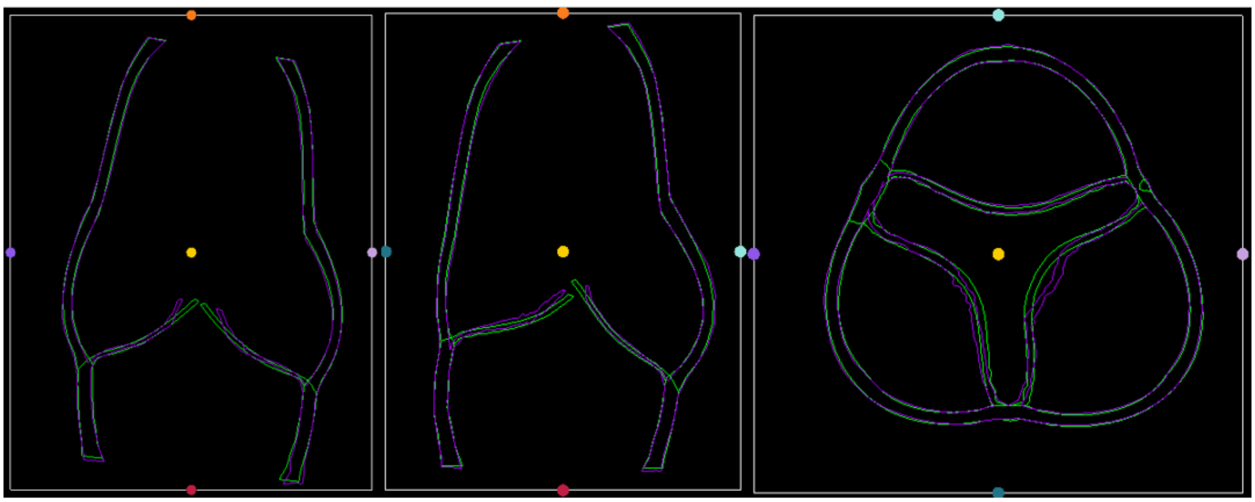


Figure 74: Cross sections of the ground truth STL file overlapped with the STL file created using the Horn-Schuck optical flow nodal displacements for time step 7: front view (left), side view (center), top view (right)

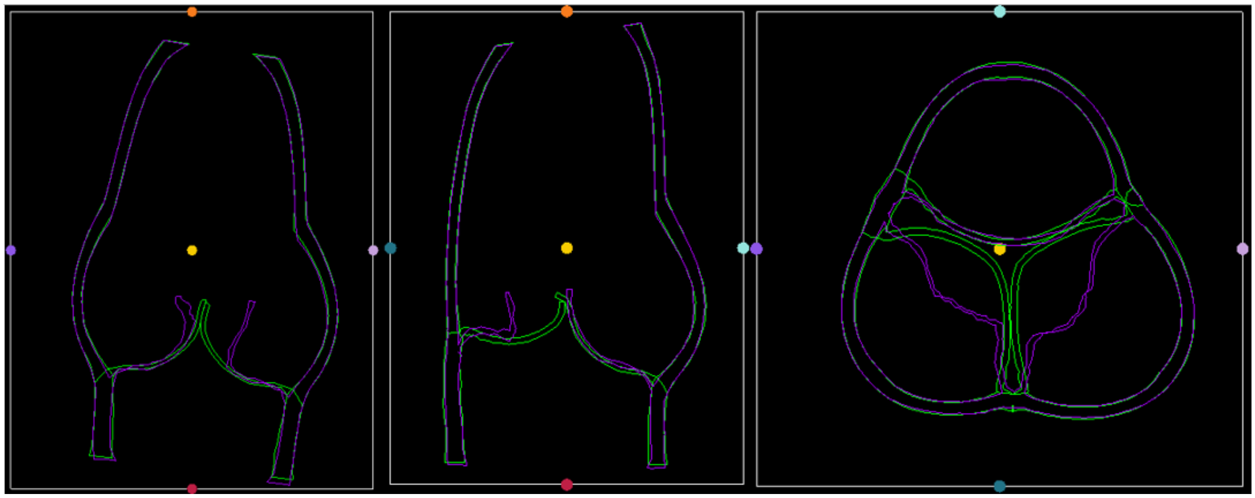


Figure 75: Cross sections of the ground truth STL file overlapped with the STL file created using the Horn-Schuck optical flow nodal displacements for time step 20: front view (left), side view (center), top view (right)

4.2.2 Images with Noise

Gaussian noise was added to the images at SNR levels of 10 dB, 20 dB, 30 dB, and 40 dB. The noisy images were run through the validation process using 3D Horn-Schunck optical flow with a multi-level scheme as the motion estimation method. The nodal displacement results from the noisy images were compared to the nodal displacement results of the noise-free images to determine what affect different levels of noise had on the validation process. The following section addresses how the expected level of noise in real 4D CT images affect the results of the validation process.

4.2.2.1 Correlation of Noisy Images to Noise-Free Images

Table 4 shows the Spearman correlation coefficients and corresponding p-values between the magnitudes of nodal displacements from the images with noise at varying levels of SNR, and the magnitudes of nodal displacements from the noise free images. The null hypothesis was a monotonic relationship does not exist between the magnitudes nodal displacements determined from the validation process of the noise free images compared to the noisy images.

Table 4: Spearman's ρ correlation coefficients of the magnitude of nodal displacements of images with added noise to magnitude of nodal displacements of noise free images

	SNR							
	10		20		30		40	
Time Step	Correlation to noise-free	*p* value	Correlation to noise-free	*p* value	Correlation to noise-free	*p* value	Correlation to noise-free	*p* value
2	0.943	$p<0.05$	0.997	$p<0.05$	0.999	$p<0.05$	0.999	$p<0.05$
3	0.978	$p<0.05$	0.997	$p<0.05$	0.999	$p<0.05$	0.999	$p<0.05$
4	0.909	$p<0.05$	0.992	$p<0.05$	0.998	$p<0.05$	0.998	$p<0.05$
5	0.775	$p<0.05$	0.979	$p<0.05$	0.996	$p<0.05$	0.997	$p<0.05$
6	0.819	$p<0.05$	0.983	$p<0.05$	0.997	$p<0.05$	0.998	$p<0.05$
7	0.936	$p<0.05$	0.995	$p<0.05$	0.999	$p<0.05$	0.999	$p<0.05$
8	0.992	$p<0.05$	0.999	$p<0.05$	0.999	$p<0.05$	0.999	$p<0.05$
9	0.996	$p<0.05$	0.998	$p<0.05$	0.999	$p<0.05$	0.999	$p<0.05$
10	0.991	$p<0.05$	0.999	$p<0.05$	0.999	$p<0.05$	0.999	$p<0.05$
11	0.991	$p<0.05$	0.999	$p<0.05$	0.999	$p<0.05$	0.999	$p<0.05$
12	0.991	$p<0.05$	0.999	$p<0.05$	0.999	$p<0.05$	0.999	$p<0.05$
13	0.994	$p<0.05$	0.999	$p<0.05$	0.999	$p<0.05$	0.999	$p<0.05$
14	0.995	$p<0.05$	0.999	$p<0.05$	0.999	$p<0.05$	0.999	$p<0.05$
15	0.995	$p<0.05$	0.999	$p<0.05$	0.999	$p<0.05$	0.999	$p<0.05$
16	0.991	$p<0.05$	0.999	$p<0.05$	0.999	$p<0.05$	0.999	$p<0.05$
17	0.993	$p<0.05$	0.999	$p<0.05$	0.999	$p<0.05$	0.999	$p<0.05$
18	0.992	$p<0.05$	0.999	$p<0.05$	0.999	$p<0.05$	0.999	$p<0.05$
19	0.994	$p<0.05$	0.999	$p<0.05$	0.999	$p<0.05$	0.999	$p<0.05$
20	0.997	$p<0.05$	0.999	$p<0.05$	0.999	$p<0.05$	0.999	$p<0.05$

Figure 76 shows the boxplots of the magnitude of the nodal displacements determined using 3D Horn-Schunck optical flow with a multi-level scheme on noise-free images, and images with varying levels of SNR; 10 dB, 20 dB, 30 dB, and 40 dB, from top to bottom respectively.

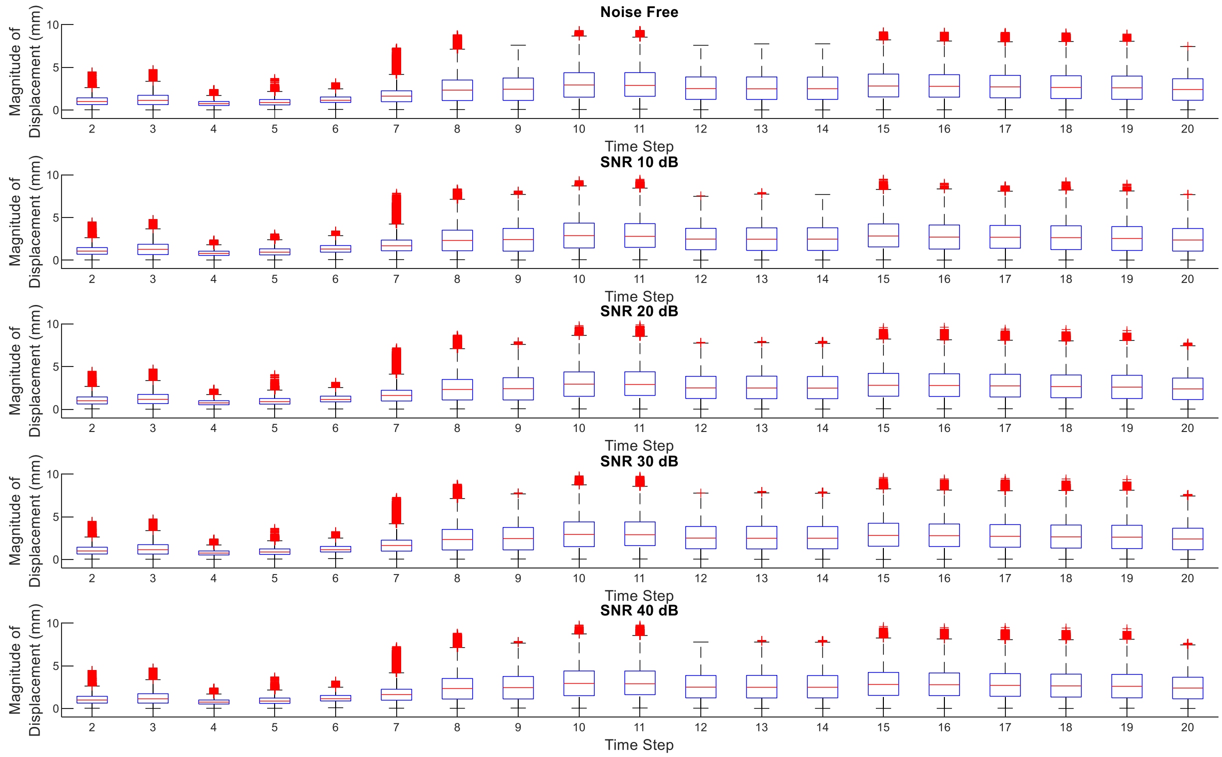


Figure 76: Boxplots of the absolute nodal displacement from the Horn-Schunck optical flow on noise free images (top), and images with varying level of SNR; 10 dB, 20 dB, 30 dB and 40 dB from second from top to bottom respectively

The correlation coefficients of the magnitude of nodal displacements, and box plots of the magnitudes of nodal displacements only show the relationship between the magnitudes of the nodal displacements. The trajectory of the vectors is equally important. Table 5 shows the spearman correlation coefficients and corresponding p-values between the angular error of the nodal displacements compared to the ground truth of the noise free images, and the angular error of the nodal displacements compared to the ground truth of the noisy images.

Table 5: Spearman's ρ correlation coefficients of the angular error of the nodal displacements of the images with added noise to angular error of nodal displacements of noise free images

	SNR							
	10		20		30		40	
Time Step	Correlation to noise-free	*p* value	Correlation to noise-free	*p* value	Correlation to noise-free	*p* value	Correlation to noise-free	*p* value
2	0.734	$p<0.05$	0.970	$p<0.05$	0.992	$p<0.05$	0.993	$p<0.05$
3	0.750	$p<0.05$	0.968	$p<0.05$	0.991	$p<0.05$	0.993	$p<0.05$
4	0.773	$p<0.05$	0.976	$p<0.05$	0.994	$p<0.05$	0.995	$p<0.05$
5	0.631	$p<0.05$	0.965	$p<0.05$	0.994	$p<0.05$	0.996	$p<0.05$
6	0.585	$p<0.05$	0.972	$p<0.05$	0.993	$p<0.05$	0.995	$p<0.05$
7	0.795	$p<0.05$	0.975	$p<0.05$	0.994	$p<0.05$	0.995	$p<0.05$
8	0.896	$p<0.05$	0.992	$p<0.05$	0.995	$p<0.05$	0.996	$p<0.05$
9	0.930	$p<0.05$	0.984	$p<0.05$	0.995	$p<0.05$	0.995	$p<0.05$
10	0.857	$p<0.05$	0.992	$p<0.05$	0.997	$p<0.05$	0.997	$p<0.05$
11	0.883	$p<0.05$	0.992	$p<0.05$	0.997	$p<0.05$	0.997	$p<0.05$
12	0.855	$p<0.05$	0.991	$p<0.05$	0.996	$p<0.05$	0.996	$p<0.05$
13	0.818	$p<0.05$	0.992	$p<0.05$	0.995	$p<0.05$	0.996	$p<0.05$
14	0.912	$p<0.05$	0.991	$p<0.05$	0.996	$p<0.05$	0.996	$p<0.05$
15	0.930	$p<0.05$	0.993	$p<0.05$	0.996	$p<0.05$	0.996	$p<0.05$
16	0.848	$p<0.05$	0.991	$p<0.05$	0.996	$p<0.05$	0.997	$p<0.05$
17	0.910	$p<0.05$	0.988	$p<0.05$	0.996	$p<0.05$	0.997	$p<0.05$
18	0.870	$p<0.05$	0.991	$p<0.05$	0.996	$p<0.05$	0.997	$p<0.05$
19	0.881	$p<0.05$	0.992	$p<0.05$	0.996	$p<0.05$	0.996	$p<0.05$
20	0.934	$p<0.05$	0.990	$p<0.05$	0.993	$p<0.05$	0.994	$p<0.05$

5 Discussion

The concept of a method to validate biological finite element models using only medical images was proposed and implemented using synthetic 4D CT images of an aortic valve based on an existing finite element model. The proposed validation method involved: estimating a voxel displacement field using a direct method of 3D motion estimation, converting the voxel displacement field into a nodal displacement field, and validating the results of a finite element model by comparing the nodal displacement field of the finite element model to the nodal displacement field from the medical images. As a preliminary step in the work towards using image processing techniques to validate finite element models, the method was implemented on ideal synthetic medical images based on an existing finite element model. The nodal displacement field of the existing finite element model acted as the ground truth such that the validity of the proposed validation method could be tested. The implementation of this method on the synthetic images involved: estimating a voxel displacement field using three different direct methods of 3D motion estimation, converting the voxel displacement field into a nodal displacement field, and comparing the resulting nodal displacement field to the ground truth nodal displacement field in order to validate the validation method. The following sections discuss the success of the components of the method, which motion estimation method should be used in future work, and the expectations of how the method may translate to real medical images in the future.

5.1 Selection of Parameters within the Process

Typically, optical flow algorithms are used to estimate the motion where the entire voxel displacement field is of interest. The application of the optical flow algorithm in this case was unique as only the displacement field of a specific set of nodes in known locations were of interest. Not only were the locations of the nodes known, but also what structure or segment they belonged to. This information allowed for different parameters, alpha and gamma values, to be selected for groups of nodes from different segments or structures. The nodes of the aortic valve were divided into two structures; aorta nodes and leaflet nodes. The alpha value was used to define the trade-off between the brightness constancy constraint, and the velocity smoothness constraint. The alpha value was the only parameter that needed to be selected for the Horn-Schunck optical flow motion estimation method. It was expected that the valve leaflets, which undergo larger magnitudes of displacement than the aorta, would yield more accurate results with a smaller alpha value, while the aorta nodes would prefer a larger alpha value. This was reflected in the general trend of the alpha value parameters selected for the Horn-Schunck optical flow motion estimation method. The algorithm iterated through a number of choices for alpha values between 0.2 and 1 and selected the best fit based on how well the image warped using the voxel displacement field matched the original image for two perpendicular cross sections of the valve. The alpha values selected for the leaflets of the valve were smaller than the alpha values selected for the aorta of the valve. The alpha values selected for the leaflets were found to be higher for the earlier time steps than the later time steps, while the alpha values selected for the aorta do not seem to relate to the time step. This was also expected, as the leaflets undergo greater magnitudes of displacement as the time steps increase, while the magnitude of displacement for the aorta remains smaller. For Horn-Schunck optical flow the use of different alpha values for the aorta nodes of the valve and the leaflet nodes of the valve proved to further the ability to capture larger motion without compromising the precision of smaller motion. Brox optical flow required the selection of an alpha value, and a gamma value. The alpha value represented the same trade-off as in Horn-Schunck optical flow. The gamma value was used to define the trade-off between the brightness constancy constraint and the gradient constancy constraint. In theory, when the gamma value is set to zero, Brox optical flow is reduced to Horn-Schunck optical flow. When tuning the parameters for Brox optical flow, the gamma value was set to zero and the alpha value was selected, then the gamma value was selected using the set alpha value. Like with the Horn-Schunck optical flow, it was expected that the valve leaflets, which undergo larger magnitudes of displacement than the aorta, would yield more accurate results with a smaller alpha value, while the aorta nodes would prefer a larger alpha value. This was reflected in the general trend of the alpha value parameters selected for the Brox optical flow motion estimation method. The algorithm set the gamma value to zero and then iterated through a number of choices for alpha values between 0.02 and 0.1 and selected the best fit based on how well the image that was warped using the voxel displacement field matched the original images for two perpendicular cross sections of the valve. Like with the Horn-Schunck, the alpha values selected for the leaflets of the valve were smaller than alpha values selected for the aorta of the valve. Unlike the Horn-Schunck, there seemed to be no change in the alpha values selected for the valve leaflets as the time steps changed. Interestingly, the alphas selected for the aorta were smaller for the later time steps. For the gamma parameter selection, it was expected that the gamma value for the leaflets would be higher during the timesteps the valve was still open, and lower during the time steps when the valve was closing. This was because, in theory, the gradient constancy constraint, where the gradient of the brightness intensity is constant, does not support the motion of a closing valve. It was also expected that the gamma value of the aorta nodes would not be related to which time step was being assessed. These were both reflected in the gamma values selected by the algorithm. The gamma values selected for the valve leaflets were higher for the first six time steps, while the valve was open, and lower for the final 14 time steps, while the valve was closing and closed. The gamma values selected for the aorta of the valve did not seem to follow any particular trend. For Brox optical flow, the use of different alpha and gamma values for the aorta nodes and valve leaflet nodes proved to match the expected trends, which yielded better motion estimation. The shortcomings of the Brox optical flow method within the greater validation method were not related to the process of selecting parameters. Choosing the optical flow parameters based on two perpendicular cross sections of the image proved to be an effective method.

5.2 Comparison of Motion Estimation Methods

The three direct methods of 3D motion estimation implemented within the process were; Horn-Schunck optical flow with a multi level scheme, Brox optical flow, and demons method with a multi level scheme. All three motion estimation methods were run on a Windows laptop running Windows 10 on a dual-core processor with 16 GB of RAM. Horn-Schunck optical flow took approximately 6 hours to complete the 19 time steps. Brox optical flow took approximately 24 to complete the 19 time steps. Demons method took approximately 2 hours to complete the 19 time steps. The following sections discuss the effectiveness of the methods of motion estimation based on their ability to capture the motion of the aortic valve, and which method is best suited for future work.

5.2.1 Box Plots

Box plots of the magnitudes of nodal displacement from the different motion estimation methods, and the ground truth were plotted over all the time steps for comparison. Horn-Schunck optical flow had a larger interquartile range than the ground truth, and underestimated the magnitude of displacement at the time steps corresponding to larger deformations. Brox optical flow had a smaller interquartile range than the ground truth, and underestimated the magnitude of displacement at all the timesteps. The demons method most closely resembled the distribution of the ground truth; however it consistently underestimated the median. The box plots only compare the magnitude of the nodal displacements, they do not account for the directional component of the displacement vectors which are as important to consider as well.

5.2.2 Correlation Coefficient

The Spearman Rank Correlation coefficient between the magnitude of the calculated nodal displacements and the magnitude of the ground truth nodal displacements were computed for each of the motion estimation methods at each time step. The Spearman Rank Correlation test is a nonparametric version of the Pearson Correlation test that is used for data sets which are not normally distributed. It tests the monotonic strength of the relationship between two variables. The null hypothesis was that there was no monotonic relationship between the calculated displacement field and the ground truth displacement field. The p-values calculated were all less than 0.05; therefore, the null hypothesis could be rejected with 95% confidence. For sixteen of the nineteen time steps, there was a stronger monotonic relationship between the ground truth nodal displacement and the calculated nodal displacement when the calculated nodal displacement was determined using Horn-Schunck optical flow instead of the other motion estimation methods. The correlation coefficients on these steps ranged between 0.73 and 0.96. The correlation when using Brox optical flow motion estimation method was generally less strong than the other two motion estimation methods, except for time steps 5 and 6 where a strong relationship was found with r values of 0.71 and 0.80 respectively. The correlation when using the demons method of motion estimation method were also strong for time steps 7-20, where the range of correlation coefficients were between 0.87 and 0.94.

5.2.3 Metrics of Agreeability between Calculated Nodal Displacements and Ground Truth Nodal Displacements

The results of the four different metrics of agreeability across the time steps were used to determine the method of motion estimation method that best captured the motion of the aortic valve over a cardiac cycle. The four metrics of agreeability were: difference error, absolute error, relative percent difference and angular error as given by Eq. 123 to Eq. 127. It can be seen that the metrics were not in unanimous agreement with each other as to which motion estimation method was the best for every time step. When looking at the difference error metrics, both the magnitude and trajectory of the nodal displacements were considered. Horn-Schunck optical flow produced the best results for the valve as a whole, as well as the aorta nodes and valve leaflet nodes. Although the demons method had a lower mean error at certain time steps, it also had a larger standard deviation of the difference error than the

Horn-Schunck optical flow. Brox optical flow produced similar mean and standard deviation of the difference error compared to the other two methods for time steps 2-7, however, at time steps 8-20, when the valve was more closed, the mean and standard deviation of the difference errors were much higher. When looking at the absolute error metric, only the magnitude of the nodal displacement was considered. For the majority of the time steps, the valve as a whole and the aorta nodes only, demons method had a lower mean absolute error than the other two methods. If only the leaflet nodes were considered, Horn-Schunck optical flow had a lower mean absolute error for the majority of the time steps. When looking at the metric for relative percent difference, only the magnitude of the nodal displacement was considered. For the valve as a whole, and only the aorta nodes, demons method had a lower mean relative percent difference for majority of the time steps. When looking at only the leaflet nodes, Horn-Schunk optical flow had a lower mean relative percent difference for majority of the time steps. When looking at the angular error metric, only the trajectory of the nodal displacement was considered. For the valve as a whole, the aorta nodes, and the leaflet nodes, Horn-Schunck optical flow had a lower mean angular error than the other two methods for majority of the time-steps.

It was clear that Horn-Schunck optical flow was able to capture the leaflet motion of an aortic valve better than either Brox optical flow or demons method. In metrics considering only the magnitude of the nodal displacements, the demons method had lower mean errors than Horn-Schunck optical flow for the entirety of the valves nodes, and the aorta nodes; however, Horn-Schunck optical was still selected as the best motion estimation method as its standard deviation around those mean errors was smaller, and it outperformed the demons method in the metrics that considered the trajectory of the nodal displacements.

5.2.4 Performance of Motion Estimation Methods at Specific Steps

Individual time steps were examined to determine the distribution of the error metrics amongst the nodes. The warped images using the voxel displacement fields were also presented for qualitative assessment.

5.2.4.1 Time Step 2

At time step 2 it was shown that the warped image cross sections using Horn-Schunck optical flow, and demons method sufficiently captured both the position of the aorta, and the leaflets of the valve. While the warped image cross sections using Brox optical flow did capture the position of the aorta, it did not accurately represent the position of the leaflets. The shapes of the histograms of the difference error for time step 2 showed that while both Horn-Schunck optical flow and demons method peaked around 0.5 mm, the maximum error of Horn-Schunck optical flow was much lower. All three histograms were positively skewed. The Horn-Schunck optical flow histogram had the highest kurtosis, followed by demons method, and Brox optical flow respectively. The location of the problematic nodes for demons method and Horn-Schunck optical flow were randomly clustered, while the problematic nodes for Brox optical flow were concentrated at the tips of the leaflets. The path of the tips of the leaflets were expected to be the most difficult to track as they underwent the highest displacement. The shapes of the histograms of the absolute error for time step 2 showed that all the histograms were positively skewed, and Horn-Schunck optical flow had a higher kurtosis than both demons method, and Brox optical flow. The same can be said about the shapes of the relative percent error histograms for time step 2. The sharp increase at 200% error represents the 800 nodes where the ground truth nodal displacement was 0 mm. The shapes of the histogram of the angular error for time step 2 showed Horn-Schunck optical flow had a higher kurtosis than the other two motion estimation methods. For time-step 2, Horn-Schunck optical flow was clearly the best method of motion estimation.

5.2.4.2 Time Step 7

At time step 7, it was shown that the warped image cross sections using Horn-Schunck optical flow sufficiently captured both the position of the aorta, and the leaflets of the valve. The warped image cross sections using Brox optical flow, and demons method did also capture the position of the aorta, but, it did not accurately represent the position of the leaflets. The shapes of the histograms of the difference error for time step 7 showed that demons method peaked higher at a smaller difference error; however, Horn-Schunck optical flow had a lower maximum error. The location of the problematic nodes for all the methods of motion estimation were a couple of clusters on the edge of the leaflets. Interestingly, the problematic nodes for Horn-Schunck optical flow were all on one leaflet, as opposed to demons method and Brox optical flow, where the problematic nodes spanned multiple leaflets. The shapes of the histograms of the absolute error for time step 7 showed that demons method had a higher peak at zero error, and a similar maximum error than Horn-Schunck optical flow. Brox optical flow had a similar peak to demons method; however, it also produced a much higher maximum error. demons method had the highest kurtosis, followed by Brox optical flow and then Horn-Schunck optical flow. The same can be said about the shapes of the relative percent error histograms for time step 7. The sharp increase at 200% error represents the 800 nodes where the ground truth nodal displacement was 0 mm. The shapes of the histogram of the angular error for time step 7 were similar for demons method and Horn-Schunck optical flow. Demons method peaked at a lower angular error than Horn-Schunck optical flow, but Horn-Schunck optical flow had a lower maximum error. Brox optical flow had a lower kurtosis than the other two motion estimation methods. For time step 7, demons method produced more nodal displacements with lower error than Horn-Schunck optical flow, but it also produced more nodal displacements with higher error than Horn-Schunck optical flow. Based on the cross-section images, and the similarity of the histogram results, Horn-Schunck optical flow was determined to be the best motion estimation method for time step 7.

5.2.4.3 *Time Step 20*

At time step 20, it was shown that the warped image cross sections using Horn-Schunck optical flow sufficiently captured the position of the leaflets of the valve. The warped image cross sections using Brox optical flow, and demons method did not accurately represent the position of the leaflets. The shapes of the histograms of the difference error for time step 20 showed that demons method peaked higher at a smaller difference error; however, Horn-Schunck optical flow had a lower maximum error. Brox optical flow had a smaller and later peak, as well as more nodes at higher errors than the other two motion estimation methods. The location of the problematic nodes for all the methods of motion estimation were a couple of clusters on the edge of the leaflets. The shapes of the histograms of the absolute error for time step 20 showed that demons method had a higher peak at zero error, and a higher maximum error than Horn-Schunck optical flow. Brox optical flow had a lower peak to demons method, and it also produced a much higher maximum error. Demons method also had the highest kurtosis. The same can be said about the shapes of the relative percent error histograms for time step 20. The sharp increase at 200% error represents the 800 nodes where the ground truth nodal displacement was 0 mm. The shapes of the histogram of the angular error for time step 20 were similar for demons method and Horn-Schunck optical flow. Demons method peaked at a lower angular error than Horn-Schunck optical flow, but Horn-Schunck optical flow had a lower maximum error. Brox optical flow had a lower kurtosis than the other two motion estimation methods. For time step 20, demons method produced more nodal displacements with lower error than Horn-Schunck optical flow, but it also produced more nodal displacements with higher error than Horn-Schunck optical flow. Based on the cross-section images, and the similarity of the histogram results, Horn-Schunck optical flow was determined to be the best motion estimation method for time step 20.

5.3 EFFECTIVENESS OF BEST MOTION ESTIMATION METHOD – HORN-SCHUNCK OPTICAL FLOW

Based on the results of the five metrics of agreement, as well as the Spearman correlation coefficients, it was determined that 3D Horn-Schunck optical flow with a multi-level scheme was the most effective method of motion estimation for models of the aortic valve. While it was the best option of the three methods evaluated it was not perfect.

5.3.1 Overlapping STL Files

STL files created using the Horn-Schunck nodal displacements were compared to the ground truth STL files using SLICER. The overlapped cross sections for time steps 2, 7, and 20 were shown in Figures 73 to 75. The Horn-Schunck STL represented the ground truth STL well for time step 2. The curvature of the aorta matched. The direction of the leaflets as they protruded from the aorta matched. The leaflets veered off slightly at the tip; however, this did not have a perceivable effect on the geometric orifice area. The Horn-Schunck STL also represented the ground truth STL well for time step 7. The curvature of the aorta matched with the exception of the bottom on the right side of the cross-section front view. This error was likely due to boundary conditions and would be avoidable going forward. The direction of the leaflets as they protruded from the aorta matched. The leaflets veered off slightly at the tip. The error at the tip was enough to have a perceivable effect on the geometric orifice area; however, compared to the acceptable error of the geometric orifice area in current validation methods, this difference was minimal. The Horn-Schunck STL did not represent the ground truth STL as well for time step 20. The curvature of the aorta matched. The direction of the of the leaflets as they protruded from the aorta was similar. Unfortunately, the middle of two of the leaflets changed direction and veered away from the ground truth which resulted in the end positions of the tips not matching. The tip of the third leaflet represented the ground truth well. The two leaflets caused enough error that the difference in the geometric orifice area was apparent and would need to be improved upon in the future. Overall, using Horn-Schunck optical flow within the presented validation method on ideal images produced promising results.

5.3.2 Noise

Gaussian noise at SNR levels of 10 dB, 20 dB, 30 dB, and 40 dB were added to the images and the validation process was conducted on all of them. The nodal displacements of the noisy images were compared to the nodal displacements of the noise free images. Both components the nodal displacements, magnitude and trajectory, were considered. The results of magnitude of nodal displacement from the validation process were compared to the results from the noise free images. It was found that a strong correlation between the noise-free and noisy images existed. As the SNR increased, the Spearman correlation coefficients at each timestep increased as well. The lowest correlation coefficients were all found at time step 5. Even though they were lower than the other coefficients they still represented a strong correlation: 0.775, 0.979, 0.996, and 0.997 for SNR levels of 10 dB, 20 dB, 30 dB, and 40 dB respectively. All the p-values were determined to be below 0.05, so it can be said with 95% confidence that a monotonic relationship existed between the nodal displacements determined by the images with noise and the images without noise. This is reinforced by the box-plots of the distribution of the magnitudes of nodal displacement for each time step at the different levels of noise. The five box plots representing no noise, 10 dB SNR, 20 dB SNR, 30 dB SNR, 40 dB SNR all look similar. In order to consider the directional component of the nodal displacements, the result of the angular error of noise free images with respect to the ground truth was compared to angular error of the noisy images with respect to the ground truth. For the SNR levels of 20 dB, 30 dB, and 40 dB, it was found that a strong correlation existed between the angular errors found using noise-free images, and the angular errors found using noisy images. The correlation coefficients were calculated using the Spearman correlation coefficient. The null hypothesis was that no monotonic relationship existed between the angular error found using noise-free images and the angular error found using noisy images. All of the corresponding p-values were below 0.05, so the null hypothesis can be rejected with 95% certainty.

With both the magnitude and directional components of the nodal displacements determined using noisy images being strongly correlated to the nodal displacements determined using noise-free images, it can be concluded that adding Gaussian noise to the images at the levels expected in current 4D CT scan technology does not negatively affect the nodal displacement results of the validation method.

5.4 Limitations and Conclusions Drawn of the Techniques

Overall, using Horn-Schunck optical flow within the proposed validation method on ideal images produced promising results. It should be noted that the results presented in this thesis were based on one existing finite element model of an aortic valve. It is expected that the results of the analysis on a different data set would follow the trend. This assumption is based of the fact that each of the 20 time steps of the current analysis were independently examined. The proposed method of validation was able to track the aorta nodes effectively through an entire cardiac cycle, and was able to track leaflet nodes through large displacements up until the valve closed. It is likely that with more work this method could be effective in tracking the leaflet nodes through the entire cardiac cycle as well. A feature-based motion estimation method could be used in conjunction with Horn-Schunck optical flow to capture the entire motion of the valve leaflets. As medical imaging technology progresses, it can be expected that the resolution of an image will increase, which will lead to better performances of the motion estimation techniques explored in this thesis. Although the general trend of the motion of the aortic valve was captured by the validation method using synthetic medical images, node-to-node comparison was not entirely reliable. However, it is arguably superior to the current validation methods for biological finite element models.

The main limitation of the work done in this thesis is that real medical images have not been used in the analysis. Significant hurdles are expected in the application of this validation method on real medical images. These hurdles include the rigid bogy motion of the valve leaflets between time steps, and the addition of other biological structures in the image. The rigid body motion between time steps will require preprocessing to ground the aortic valve to a single origin. The other biological structures will likely require thresholding to determine what is and what is not a part of the aortic valve.

6 CONCLUSION

6.1 BRIEF SUMMARY AND MAJOR CONTRIBUTIONS

This work made progress towards a method of validation for biological finite element models using patient specific 4D medical images. The concept of a method of validation was proposed and implemented on synthetic 4D medical images created from an existing finite element model of an aortic valve. The concept involved three steps. First, the voxel displacement field between the image from the first time step and any subsequent image was found using a direct method of motion estimation. Second, the voxel displacement field was converted into a nodal displacement field using the knowledge of the known node locations at the first time step. Third, the nodal displacement field was compared to the finite element model displacement field using different error metrics.

Three different direct motion estimation methods were implemented within the first step of the method and compared. The three methods were: 3D Horn-Schunck optical flow, and 3D Brox optical flow, and demons method. For two of the three motion estimation methods, implementations existed. The base code for 3D Horn-Schunck optical flow was open source, and the base code for the demons method was a MATLAB function. Both these methods were not able to capture the motion of an aortic valve alone. The addition of a multilevel scheme with a variable scale constant was integrated into each of these motion estimation methods so that larger magnitudes of displacement could by captured. The 3D Brox optical flow method was implemented for the purpose of this work. Brox optical flow added an additional constraint to Horn-Schunck optical flow which was supposed to result in a more accurate voxel displacement field. Within the two optical flow methods input parameters representing the weights of different assumptions needed to be tuned. Typically, in other optical flow application, the tuning process has been done by trial and error. Instead, the variable input parameters for the optical flow motion estimation methods were selected through an optimization process within the method. The tuning process involved a comparison of warped and original cross sections of the images to optimize the parameters. The complexity of the motion of the aortic valve resulted in the optimal set of parameters for the valve leaflets to be different from the aorta. For that reason, each structure was tuned separately. Once the voxel displacement was determined, it had to be converted into a nodal displacement field. The method proposed and implemented in order to convert the displacement fields involved overlaying the coordinates of the nodes on top of the voxels, and finding the nearest voxel with valve information to assign to the node. Finally, the nodal displacement fields found using the three methods of motion estimation were compared to the ground truth to determine which motion estimation was best, and whether the validation method was sufficient.

It was found that Horn-Schunck optical flow was best able to capture the motion of the aortic valve throughout a cardiac cycle. The proposed method of validation was able to track the aorta nodes effectively through an entire cardiac cycle, and was able to track leaflet nodes through large displacements up until the valve closed. It is likely that with more work, this method could be effective in tracking the leaflet nodes through the entire cardiac cycle as well. Although the general trend of the motion of the aortic valve was captured by the validation method using synthetic medical images, node-to-node comparison was not entirely reliable. Comparison of the general trend was still superior to the current validation methods for biological finite element methods as it considered the motion of the entire structure.

6.2 Recommendations for Future Work

Of the motion estimation methods examined, Horn-Schunck optical flow was found to be most effective method and should be the focus of the work going forward. Computationally, the implementation of Horn-Schunck optical flow could be more effective if there were an exit criterion to the iteration loops. Currently, each level of the pyramid runs the computation for a specified number of iterations. That number of iterations has to be sufficient for the top level of the pyramid; however, it is more than is necessary for the lower levels of the pyramid. An exit criterion similar to the one applied in Brox optical flow, where if the difference between the displacement fields between iterations is below a threshold the loop breaks, would save a significant amount of time. Another possibility for improvement of the Horn-Schunck optical flow would be to look into using a more robust method to find the time derivative using the information of the images before and after the time steps of interest. Using a two-point finite difference scheme is generally considered a weak method to estimate a time derivative. The set of 20 sequential images should open the possibility of using more than two time steps to obtain the time derivative. Horn-Schunck optical flow was effectively able to capture the motion of the aorta across a cardiac cycle; however, it was limited in its effectiveness of capturing the motion of the leaflets during the later time steps of the cardiac cycle. Going forward, using Horn-Schunck optical flow in conjunction with a feature-based motion estimation method could be useful. A feature-based motion estimation method may perform better on the leaflet tips. The resulting sparse displacement field of a feature based motion estimation method would not matter if it were used in conjunction with Horn-Schunck optical flow, because, any missing information could be filled in by the Horn-Schunck optical flow displacement field.

To meet the long-term goal of the study, creating a validation method using real medical images, some considerations must be made. There are expected hurdles in using real 4D-CT scans of an aortic valve. The main advantage to the synthetic images was that they were all grounded to a single origin, and there was no rigid body motion of the entire valve. In real images of the aortic valve, transformations occur throughout the cardiac cycle. There would need to be a preprocessing step to align the valves to compensate for this. Of the motion estimation methods explored in this thesis, demons method was found to be most effective in identifying rigid transformations.

7 WORKS CITED

[1] M. Labrosse, C. Beller, M. Boodhwani, C. Hudson and B. Sohmer, "Subject-specific finite-element modeling of normal aortic valve biomechanics from 3D+t TEE images," *Medical Image Analysis,* vol. 20, pp. 162-172, 2015.

[2] H. B. Henninger, S. P. Reese, A. Anderson and J. Weiss, "Validation of Computational Models in Biomechanics," *Journal of Engineering in Medicine,* vol. 224, no. 7, pp. 801-812, 2010.

[3] O. Panagiotopoulou, "Finite element analysis (FEA): Applying an engineering method to functional morphology in anthropology and human biology," *Annals of Human Biology,* vol. 36, no. 5, pp. 609-623, 2009.

[4] A. Satriano, S. Rivolo, G. Martufi, E. A. Finol and E. S. D. Martino, "In vivo strain assesment of the abdominal aortic anueurysm," *Journal of Biomechanics,* vol. 48, pp. 354-360, 2015.

[5] M. A. Mustafa, "A data-driven learning approach to image registration," University of Nottingham, Nottingham, 2016.

[6] J.-P. Thirion, "Image matching as a diffusion process: an analogy with Maxwell's demons," *Medical Image Analysis,* vol. 2, no. 3, pp. 243-260, 1998.

[7] MATLAB, "Image Processing Toolbox Reference," The MathWorks Inc., Natick, 2021.

[8] W. Chen, "High Accuracy Optical Flow Method Based on a Theory for Warping: 3D Extension," in *International Conference Image Analysis and Recognition*, 2010.

[9] Brox, Bruhn, Papenberd and Weickert, "High Accuracy Optical Flow Estimation Based on a Theory for Warping," in *European Conference on Computer Vision*, Heidelberg, 2004.

[10] L. Schwer, "Guide for Verification and Validation in Computational Solid Mechanics," ASME, 2006.

[11] J. Gao, B. Liu, M. Zhang, H. Gong and B. Gao, "Strain Distribution Evaluation of Rat Tibia under Axial Compressive Load by Combining Strain Gauge Measurement and Finite Element Analysis," *Applied Bionics and Biomechanics,* vol. 2019, 2019.

[12] M. S. Reddy, R. Sundram, H. Abdelatty and E. Abdemagyd, "Application of Finite Element Model in Implant Dentistry: A Systematic Review," *Journal of Pharmacy and BioAllied Sciences,* vol. 11, no. 2, pp. S85-S91, 2019.

[13] A. Takacs, C. Dudescu, M. Hardau and A. Botean, "Experimental Validation of a Finite Element Model of an Osteoporotic Human Femoral Bone using Strain Gauge Measurement," *Applied Mechanics and Materials,* vol. 658, pp. 513-519, 2014.

[14] F. Groning, J. Liu, M. Fagan and P. Higgins, "Validating a voxel-based finite element model of a human mandible using digital speckle pattern interferometry," *Journal of Biomechanics,* vol. 42, no. 9, pp. 1224-1229, 2009.

[15] Gharaie, Mosadegh and Morsi, "In Vitro Validation of a Numerical Simulation of Leaflet Kinematics in a polymeric Aortic Valve Under Physiological Conditions," *Cardiovascular Engineering and Technology,* vol. 9, no. 1, pp. 42-52, 2018.

[16] Sodhani, Reese, Aksenov, Soganci, Jockenhovel, Mela and Stapleton, "Fluid-structure interaction simulation of artificial textile reinforced aortic heart valve: Validation with an in-witro test," *Journal of Biomechanics,* vol. 78, pp. 52-69, 2018.

[17] D. Marshall, "The basic premise of motion, objects moving within frames," Cardiff School of Computer Science & Informatics, 10 October 2001. [Online]. Available: https://users.cs.cf.ac.uk/Dave.Marshall/Multimedia/node259.html#:~:text=The%20basic%20premise%20of%20motion,objects%20moving%20within%20the%20frames.&text=The%20problem%20for%20motion%20estimation,between%20these%20two%20video%20frames.. [Accessed 10 06 2020].

[18] B. K. Horn and B. G. Schunck, "Determining Optical Flow," *Artificial Intelligence,* vol. 17, pp. 185-203, 1981.

[19] Heidelberg Collaboratory for Image Processing, "Optical Flow for Industrial Applications," Wiley Industry News, 27 06 2011. [Online]. Available: https://www.wileyindustrynews.com/en/whitepaper/optical-flow-industrial-applications. [Accessed 04 11 2021].

[20] Lucas and Kanade, "An Iterative Image Regisration Technique with an Application to Stereo Vision," *Computer Science Department, Carnegie-Mellon University,* pp. 121-130, 1981.

[21] L. Ng and V. Solo, "A Data-driven Method for Choosing Smoothing Parameters in Optical Flow Problems," in *Proceedngs of International Conference on Image Processing*, Santa Barbara, CA, USA, 1997.

[22] G. A. R. Arboleda, "Numerical Methods - STO241," 25 October 2012. [Online]. Available: https://sites.google.com/site/procesosnumericos0241/fixed-point-method. [Accessed 11 06 2021].

[23] Faial and Barron, "High Accuracy Optical Flow Method Based on a Theory of Warping: Implementation and Qualitative/ Quantitative Evaluation," 2007.

[24] Barron, Fleet and Beauchemin, "Performance of Optical Flow Techniques," *International Journal of Computer Vision,* vol. 12, pp. 43-77, 1994.

[25] T. Vercauteren, X. Pennec, A. Perchant and N. Ayache, "Differomorphic demons: Efficient non-parametric image registration," *Neuro Image,* vol. 45, pp. S61-S72, 2009.

[26] Hall and Guyton, Textbook of Medical Physiology 11th ed., Elsevier Saunders, 2006.

[27] S. Pfensig, S. Kaule and etal., "Assessment of heart valve performance by finite-element design studios of polymeric leaflet-structures," *Current Directions in Biomedical Engineering,* vol. 3, no. 2, pp. 631-634, 2017.

[28] B. Goyal, A. Dogra, S. Agrawal and B. Sohi, "Noise Issues Prevailing in Various Types of Medical Images," *Biomedical and Pharmacology Journal,* vol. 11, no. 3, 2018.

[29] N. P. Hofmanna, M. Schuetza, R. Bekeredjiana and etal., "Image quality and contrast agent exposure in cardiac computed tomograpgy angiography prior to transcatheter aortic valve implantation procedures using different aquisition protocols," *European Journal of Radiology Open,* vol. 4, pp. 75-83, 2017.

[30] Y. Cy, "GitHub - vtkwrite : Exports various 2D/3D data to ParaView in VTK file format," 2020. [Online]. Available: https://github.com/joe-of-all-trades/vtkwrite. [Accessed 2 September 2020].

[31] Mathworks, "mathworks.com," 29 Jan 2017. [Online]. Available: https://www.mathworks.com/matlabcentral/answers/322335-how-to-copy-and-rename-files-in-a-different-folder. [Accessed 23 Febuary 2020].

[32] W. Roberson, "MATLAB Answers," 15 Mar 2017. [Online]. Available: https://www.mathworks.com/matlabcentral/answers/310452-how-can-i-find-26-neighbors-of-point-with-coordinates-of-i-j-k. [Accessed Nov 2019].

8 APPENDIX

8.1 PSEUDO CODE AND BLOCK DIAGRAM OVERVIEWS

8.1.1 Preprocessing

WriteVTKFiles.m pseudo code

(Inputs: *ground truth nodal displacements, first index, second index*)
Set x, y, and z to the *ground truth nodal displacements* corresponding with the *first index* in the x, y, and z directions respectively (these are the original nodal coordinates of the valve)
Set *node displacements* equal to the difference between *ground truth nodal displacements* corresponding with the *second index* and *ground truth nodal displacements* corresponding with the *first index*
Set *u*, *v*, and *w* to the *node displacement* in the x, y, and z directions respectively
Use function **vtkwrite.m** to create VTK file [30]

    (Inputs: *x, y, z, u, v, w*
    Outputs: VTK file for ground truth displacement of each node)
    Save VTK file in specified folder [31]

(Outputs: VTK file for ground truth displacement of each node)

8.1.2 Horn-Schunck Optical Flow

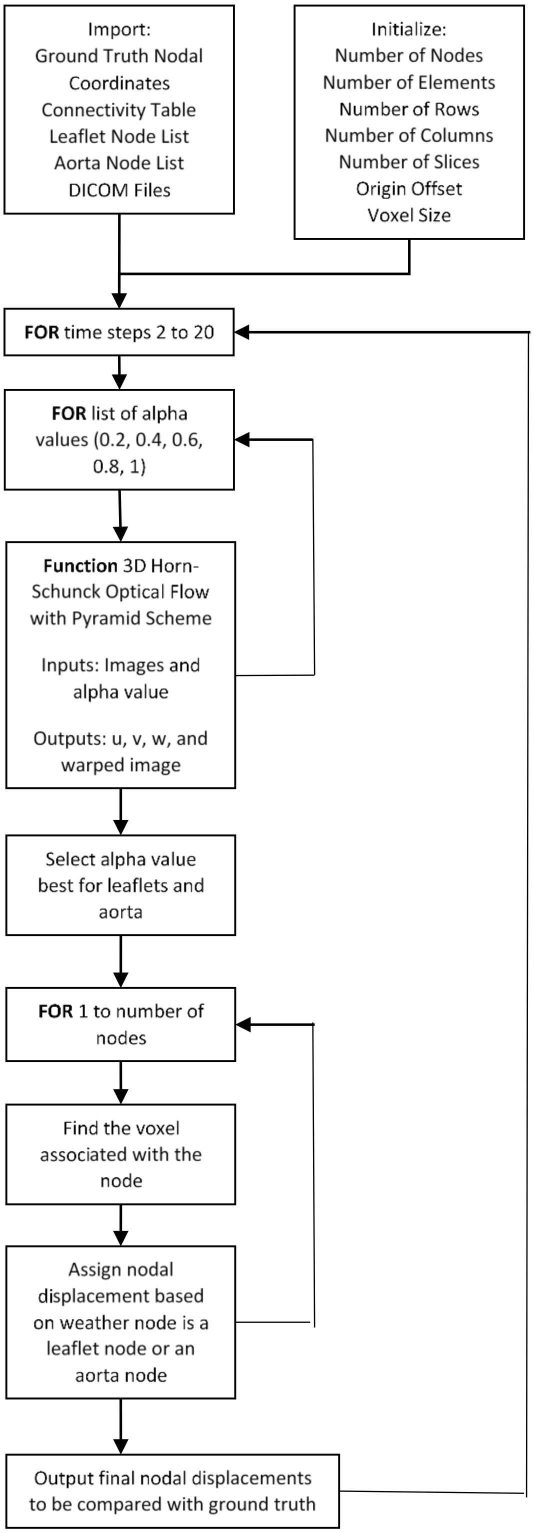


Figure 77: Block Diagram Overview of RunHS3D.m code

RunHS3D.m pseudo code

Initialize the number of *nodes*, and *elements* in FEA ground truth model
Import FEA ground truth *node coordinates* for each of the 20 time steps
Import the *Connectivity Table* of the ground truth FEA model
Import the *Leaflet Nodes List* and the *Aorta Nodes List*
Initialize number of *rows, columns* and *slices* expected in DICOM images
Import *DICOM Files*
Convert *DICOM Files* to double variable arrays
Initialize the images *Origin Offset* between FEA and DICOM files (based on Master Volume of 3DSlicer)
Initialize *Voxel Size* (based on Master Volume of 3DSlicer)
Assign *first index* as 1
For *second index* from 2 to 20 (% number of timesteps in DICOM set)
    Assign the *first image* as the *first index* DICOM Image
    Assign the *second image* as the *second index* DICOM Image
    Initialize a list of *alpha values* to test (0.2, 0.4, 0.6, 0.8, 1)
    For *k*=1:length (*alpha list*)
        Function **optic_flow_HS3D**
            (Inputs: *first image, second image, alpha,*
            Outputs: *u, v, w, warped image*)
        End Function
        Apply a smoothing filter on *u*, *v*, and *w* using **smooth3** box filter with a convolution kernel size 9X9X9
        Save the *u*, *v*, and *w* displacement results under holding variables; *uhold, vhold,* and *whold* which are of size (*rows, columns, slices,* length (*alpha list*))
        Create *side image* using **reshape** of the *first image* and the *warped image*
        Find *Difference Total* which is the sum of the absolute value of the difference of brightness intensities between the *first image* and the *warped image* of the front middle slices and side middle slices
        Find the *Difference Total Leaflets*, and *Difference Total Aorta*, by sectioning out just the rows around the leaflets, and everything but the leaflets respectively
    End For
    Find the *alpha value* associated with the minimum input of *Difference Total Leaflets*, and *Difference Total Aorta* and assign those alpha values as *alpha leaflet* and *alpha aorta*
    For i=1 to *nodes*
        Calculate the *row*, *column*, and *depth* of the node *i* from the FEA model by converting from mm to voxels using the *voxel size* and *voxel offset*
        If *image 1* at the *row, column, and depth* has a value that is not zero (a brightness intensity greater than zero represents a non-black square)
            Assign *nodal displacements* row *i* columns 1, 2 and 3 the *uhold, vhold,* and *whold* corresponding to the *row, column,* and *depth* calculated for node *i*
                and *alpha aorta*
        Else If *image 1* at the *row, column, and depth* has a value of zero
            Use function **neighbour342.m** [32] to determine the neighboring voxels with 3 voxels in any direction of the voxel the node of interest sits in, for a total of 342 voxels
                (Inputs: *row, column, depth, Image 1*

Outputs: List of rows, columns, and pages of each of the 342"neighboring" voxels, *neighbours*)

End Function

Initialize an arbitrary *distance* greater than the 3-voxel perimeter, ex. 10

Initialize *vox* to zero

Initialize a counter, *sum,* to zero

For *ii*=1 to length of *neighbours*

If The rows, columns, and pages coordinate of the *ii* line of *neighbours* exist in the dimensions of the image size (greater than zero, and less than *rows, columns* and *slices* respectively)

If The first image voxel coordinate given by *neighbours* does not equal zero

Find the *Euclidean distance* between node *i* and the center of the neighbouring voxel *ii* in the unit of voxels

Add 1 to the *sum* counter

If *Euclidean distance* is less than *distance*

Set *distance* to *Euclidean distance*

Set *vox* to *ii*

End If

End If

End If

End For

If *sum* counter is less than 1

Set *nodal displacements* row *i,* columns 1, 2, and 3 to zero (in this case there are no neighbouring voxels of the node *i* within the boundary that are part of the aorta)

Else

Set *nodal displacements* row *i,* columns 1, 2, and 3 to *uhold, vhold,* and *whold* corresponding to the *neighbour's* voxel location (the displacement of the closest voxel to the node of interest), and *alpha aorta*

End If

End If

End For

Assign *new node displacement aorta* variable *nodal displacements* converted back into mm to be compared with ground truth

Assign *new node coordinate locations aorta* variable the calculated new node coordinate location (ground truth *Node Coordinate* timestep one + *new node displacement aorta*)

For i=1 to *nodes*

Calculate the *row*, *column*, and *depth* of the node *i* from the FEA model by converting from mm to voxels using the *voxel size* and *voxel offset*

If *image 1* at the *row, column, and depth* has a value that is not zero (a brightness intensity greater than zero represents a non-black square)

Assign *nodal displacements* row *i* columns 1, 2 and 3 the *uhold, vhold,* and *whold* corresponding to the *row, column,* and *depth* calculated for node *i* and *alpha leaflet*

Else If *image 1* at the *row, column, and depth* has a value of zero

Use function **neighbour342.m** [32] to determine the neighboring voxels with 3 voxels in any direction of the voxel the node of interest sits in, for a total of 342 voxels

(Inputs: *row, column, depth, Image 1*
Outputs: List of rows, columns, and pages of each of the 342 "neighboring" voxels, *neighbours*)

End Function

Initialize an arbitrary *distance* greater than the 3-voxel perimeter, ex. 10

Initialize *vox* to zero

Initialize a counter, *sum,* to zero

For *ii*=1 to length of *neighbours*

If The rows, columns, and pages coordinate of the *ii* line of *neighbours* exist in the dimensions of the image size (greater than zero, and less than *rows, columns* and *slices* respectively)

If The first image voxel coordinate given by *neighbours* does not equal zero

Find the *Euclidean distance* between node *i* and the center of the neighbouring voxel *ii* in the unit of voxels

Add 1 to the *sum* counter

If *Euclidean distance* is less than *distance*

Set *distance* to *Euclidean distance*

Set *vox* to *ii*

End If

End If

End If

End For

If *sum* counter is less than 1

Set *nodal displacements* row *i,* columns 1, 2, and 3 to zero (in this case there are no neighbouring voxels of the node *i* within the boundary that are part of the aorta)

Else

Set *nodal displacements* row *i,* columns 1, 2, and 3 to *uhold, vhold,* and *whold* corresponding to the *neighbour's* voxel location (the displacement of the closest voxel to the node of interest), and *alpha leaflet*

End If

End If

End For

Assign *new node displacement leaflet* variable *nodal displacements* converted back into mm to be compared with ground truth

Assign *new node coordinate locations leaflets* variable the calculated new node coordinate location (ground truth *Node Coordinate* timestep one + *new node displacement leaflet*)

For *p*=1:*nodes*

Assign *Final Nodal Displacements* page *second index* the *new node displacement aorta*

For *q*=1 to the length of the *leaflet nodes* list

If node number *p* is on the *leaflet nodes* list

Reassign *Final Nodal Displacements* row *p* to *new node displacement leaf*

End If

End For

End For

Output the statistics to compare the Horn-Schunck Optical Flow algorithm nodal displacements to the ground truth nodal displacements

Create STL file using Horn Schunk nodal displacements

End For

8.1.3 Brox Optical Flow

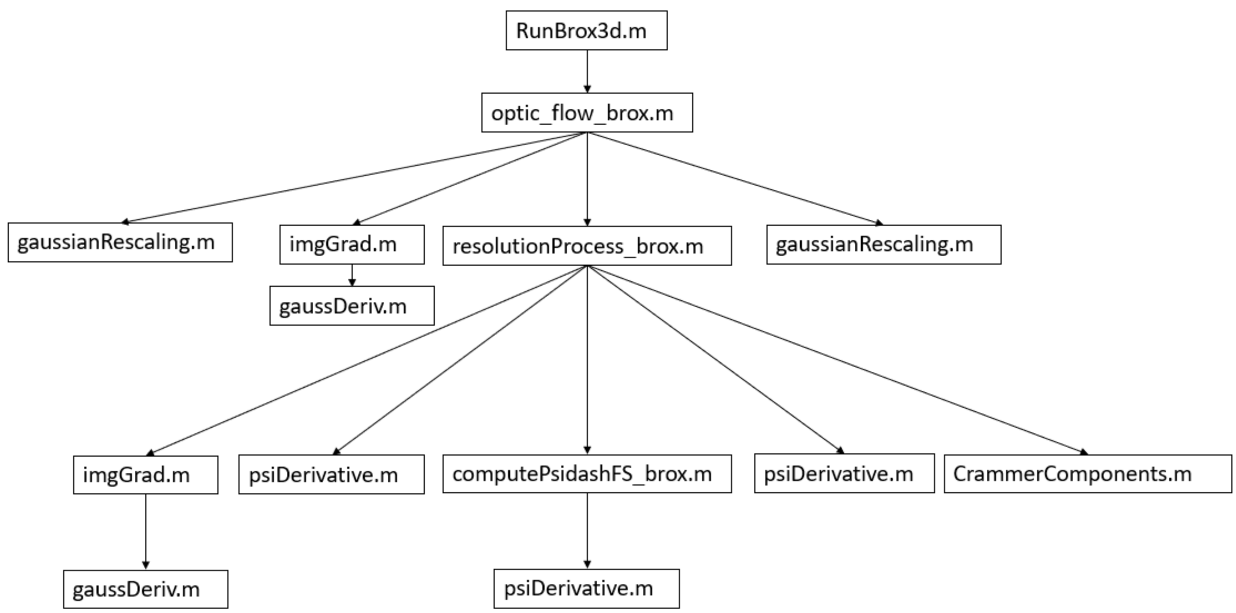


Figure 78: Flow chart of functions used in 3D Brox optical flow

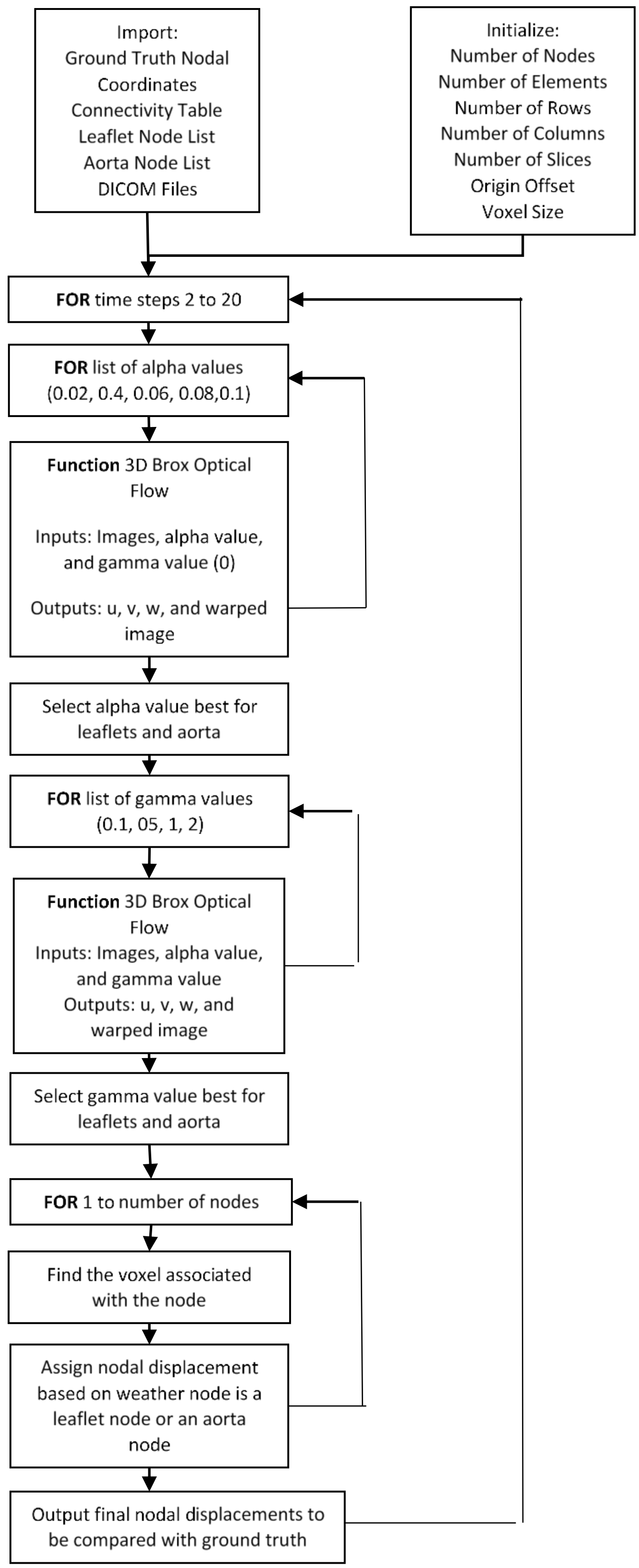


Figure 79:RunBrox3dV3 Block Diagram

RunBrox3dV3.m pseudo code

Initialize the number of *nodes*, and *elements* in FEA ground truth model
Import FEA ground truth *node coordinates* for each of the 20 timesteps
Import the *Connectivity Table* of the ground truth FEA model
Import the *Leaflet Nodes List* and the *Aorta Nodes List*
Initialize number of *rows, columns* and *slices* expected in DICOM images
Import *DICOM Files*
Convert *DICOM Files* to double variable arrays
Initialize the images *Origin Offset* between FEA and DICOM files (based on Master Volume of 3DSlicer)
Initialize *Voxel Size* (based on Master Volume of 3DSlicer)
Assign *first index* as 1
For *second index* from 2 to 20 (% number of timesteps in DICOM set)
    Assign the *first image* as the *first index* DICOM Image
    Assign the *second image* as the *second index* DICOM Image
    Initialize a list of *alpha values* to test (0.04, 0.06, 0.08, 0.1)
    Initialize *gamma*=0
    For *k*=1: length (*alpha list*)
        Function **optic_flow_brox**
            (Inputs: *first image, second image, alpha, gamma*
            Outputs: *u, v, w, warped image*)
        End Function
        Apply a smoothing filter on *u*, *v*, and *w* using **smooth3** box filter with a convolution kernel size 9X9X9
        Save the *u*, *v*, and *w* displacement results under holding variables; *uhold, vhold,* and *whold* which are of size (*rows, columns, slices,* length (*alpha list*))
        Create *side image* using **reshape** of the *first image* and the *warped image*
        Find *Difference Total* which is the sum of the absolute value of the difference of brightness intensities between the *first image* and the *warped image* of the front middle slices and side middle slices
        Find the *Difference Total Leaflets*, and *Difference Total Aorta*, by sectioning out just the rows around the leaflets, and everything but the leaflets respectively
    End For
    Find the *alpha value* associated with the minimum input of *Difference Total Leaflets*, and *Difference Total Aorta* and assign those alpha values as *alpha leaflet* and *alpha aorta*
    Set *alpha* to *alpha aorta*
    Initialize *gamma list* as [0.1, 0.5, 1, and 2]
    For *k*=1:length (*gamma list*)
        Function **optic_flow_brox**
            (Inputs: *first image, second image, alpha, gamma*
            Outputs: *u, v, w, warped image*)
        End Function
        Apply a smoothing filter on *u*, *v*, and *w* using **smooth3** box filter with a convolution kernel size 9X9X9
        Save the *u*, *v*, and *w* displacement results under holding variables; *uhold, vhold,* and *whold* which are of size (*rows, columns, slices,* length (*gamma list*))
        Create *side image* using **reshape** of the *warped image*

Find *Difference Total* which is the sum of the absolute value of the difference of brightness intensities between the *first image* and the *warped image* of the front middle slices and side middle slices

Find the *Difference Total Leaflets*, and *Difference Total Aorta*, by sectioning out just the rows around the leaflets, and everything but the leaflets respectively

End For

Find the *gamma value* associated with the minimum input of *Difference Total Aorta* and assign those gamma values as *gamma aorta*

For i=1 to *nodes*

Calculate the *row*, *column*, and *depth* of the node *i* from the FEA model by converting from mm to voxels using the *voxel size* and *voxel offset*

If *image 1* at the *row, column, and depth* has a value that is not zero (a brightness intensity greater than zero represents a non-black square)

Assign *nodal displacements* row *i* columns 1, 2 and 3 the *uhold, vhold,* and *whold* corresponding to the *row, column,* and *depth* calculated for node *i*

and *aorta gamma*

Else If *image 1* at the *row, column, and depth* has a value of zero

Use function **neighbour342.m** [32] to determine the neighboring voxels with 3 voxels in any direction of the voxel the node of interest sits in, for a total of 342 voxels

(Inputs: *row, column, depth, Image 1*

Outputs: List of rows, columns, and pages of each of the 342 “neighboring” voxels, *neighbours*)

End Function

Initialize an arbitrary *distance* greater than the 3-voxel perimeter, ex. 10

Initialize *vox* to zero

Initialize a counter, *sum,* to zero

For *ii*=1 to length of *neighbours*

If The rows, columns, and pages coordinate of the *ii* line of *neighbours* exist in the dimensions of the image size (greater than zero, and less than *rows, columns* and *slices* respectively)

If The first image voxel coordinate given by *neighbours* does not equal zero

Find the *Euclidean distance* between node *i* and the center of the neighbouring voxel *ii* in the unit of voxels

Add 1 to the *sum* counter

If *Euclidean distance* is less than *distance*

Set *distance* to *Euclidean distance*

Set *vox* to *ii*

End If

End If

End If

End For

If *sum* counter is less than 1

Set *nodal displacements* row *i,* columns 1, 2, and 3 to zero (in this case there are no neighbouring voxels of the node *i* within the boundary that are part of the aorta)

Else

Set *nodal displacements* row *i,* columns 1, 2, and 3 to *uhold, vhold,* and *whold* corresponding to the *neighbour's* voxel location (the displacement of the closest voxel to the node of interest), and *gamma aorta*

End If

End If

End For

Assign *new node displacement aorta* variable *nodal displacements* converted back into mm to be compared with ground truth

Assign *new node coordinate locations aorta* variable the calculated new node coordinate location (ground truth *Node Coordinate* timestep one + *new node displacement aorta*)

Set *alpha* to *alpha aorta*

Initialize *gamma list* as [0.1, 0.5, 1, and 2]

For *k*=1:length (*gamma list*)

Function **optic_flow_brox**

(Inputs: *first image, second image, alpha, gamma*

Outputs: *u, v, w, warped image*)

End Function

Apply a smoothing filter on *u*, *v*, and *w* using **smooth3** box filter with a convolution kernel size 9X9X9

Save the *u*, *v*, and *w* displacement results under holding variables; *uhold, vhold,* and *whold* which are of size (*rows, columns, slices,* length (*gamma list*))

Create *side image* using **reshape** of the *warped image*

Find *Difference Total* which is the sum of the absolute value of the difference of brightness intensities between the *first image* and the *warped image* of the front middle slices and side middle slices

Find the *Difference Total Leaflets*, and *Difference Total Aorta*, by sectioning out just the rows around the leaflets, and everything but the leaflets respectively

End For

Find the *gamma value* associated with the minimum input of *Difference Total Leaflet* and assign those gamma values as *gamma leaflet*

For i=1 to *nodes*

Calculate the *row*, *column*, and *depth* of the node *i* from the FEA model by converting from mm to voxels using the *voxel size* and *voxel offset*

If *image 1* at the *row, column, and depth* has a value that is not zero (a brightness intensity greater than zero represents a non-black square)

Assign *nodal displacements* row *i* columns 1, 2 and 3 the *uhold, vhold,* and *whold* corresponding to the *row, column,* and *depth* calculated for node *i* and *aorta leaflet*

Else If *image 1* at the *row, column, and depth* has a value of zero

Use function **neighbour342.m** [32] to determine the neighboring voxels with 3 voxels in any direction of the voxel the node of interest sits in, for a total of 342 voxels

(Inputs: *row, column, depth, Image 1*

Outputs: List of rows, columns, and pages of each of the 342 "neighboring" voxels, *neighbours*)

End Function

Initialize an arbitrary *distance* greater than the 3-voxel perimeter, ex. 10

```
                    Initialize vox to zero
                    Initialize a counter, sum, to zero
                    For ii=1 to length of neighbours
                        If The rows, columns, and pages coordinate of the ii line of neighbours
                        exist in the dimensions of the image size (greater than zero, and less
                        than rows, columns and slices respectively)
                            If The first image voxel coordinate given by neighbours does not
                            equal zero
                                Find the Euclidean distance between node i and the
                                center of the neighbouring voxel ii in the unit of voxels
                                Add 1 to the sum counter
                                If Euclidean distance is less than distance
                                    Set distance to Euclidean distance
                                    Set vox to ii
                                End If
                            End If
                        End If
                    End For
                    If sum counter is less than 1
                        Set nodal displacements row i, columns 1, 2, and 3 to zero (in this case
                        there are no neighbouring voxels of the node i within the boundary that
                        are part of the aorta)
                    Else
                        Set nodal displacements row i, columns 1, 2, and 3 to uhold, vhold, and
                        whold corresponding to the neighbour's voxel location (the
                        displacement of the closest voxel to the node of interest), and gamma
                        leaflet
                    End If
                End If
            End For
            Assign new node displacement leaflet variable nodal displacements converted back into mm to
            be compared with ground truth
            Assign new node coordinate locations leaflets variable the calculated new node coordinate
            location (ground truth Node Coordinate timestep one + new node displacement leaflet)
            For p=1:nodes
                Assign Final Nodal Displacements page second index the new node displacement aorta
                For q=1 to the length of the leaflet nodes list
                    If node number p is on the leaflet nodes list
                        Reassign Final Nodal Displacements row p to new node displacement
                        leaf
                    End If
                End For
            End For
        Output the statistics to compare the Brox Optical Flow Algorithm nodal displacements to the ground
        truth nodal displacements
        Create STL file using Brox nodal displacements
        End For
```

optic_flow_brox.m pseudo code

(Inputs: Images (*I1*, *I2*))
Initialize *alpha* (global smoothness variable) and *gamma* (global weight for derivatives) parameters
Initialize the *number of pyramid levels*
Initialize the *scale constant*
Rescale the images to their smallest size using **gaussianRescaling.m**
    (inputs: images, *scale constant* ^ number of pyramid levels
    outputs: smallest images of the pyramid)
Initialize the displacement field (*u,v,w*) to a zero array the size of the smallest image
FOR *i=number of pyramid levels* to 1 do
    Find derivatives of *I1* and *I2* in the x, y, and z directions (*Ikx1, Iky1, Ikz1, Ikx2, Iky2*, and *Ikz2*) using **imgGrad.m**
        (inputs: *I1, I2*
        outputs: *Ikx, Iky, Ikz, Ikx2, Iky2*, and *Ikz2*) *Note: use **imgGrad.m** separately for each image
    Find time derivatives (*Ikt, Ixt, Iyt, Izt*) by:
    Ikt=I2-I1
    Ixt=Ikx2-Ikx1
    Iyt=Iky2-Iky1
    Izt=Ikz2-Ikz1
    Find change of displacement (*du*, *dv*, *dw*) using **resolutionProcess_brox.m**
        (Inputs: *Ikt*, *Ikx2*, *Iky2*, *Ikz2*, *Ixt*, *Iyt*, *Izt*, *alpha*, *gamma*, *u*, *v*, *w*, *#outer iterations*, *#inner iterations*
        Outputs: *du*, *dv*, *dw*)
    u=u+du, *v=v+dv*, *w=w+dw*
    Increase the size of the images to the next level of the pyramid using **gaussianRescaling.m**
        (inputs: *image*s, *scale constant* ^ i)
        (outputs: next level *images* of the pyramid)
    Upsize *u,v*, and *w* to the new pyramid size using MATLAB function **imresize3**
    Warp the new sized img2 using the new sized displacement field using mywarp.m
END
(Outputs: u, v, w)

gaussianRescaling.m pseudo code
(Inputs: *image*, *scale factor*, *sigma*)
Default *sigma* is 1
Apply Gaussian smoothing to input images using MATLAB function **imgaussfilt3**
    (inputs: *image*, 1/*scale factor*
    outputs: *smoothed image*)
Resize *smoothed image* by *scale factor* using MATLAB function **imresize3**
(Outputs: *scaled image*)

imgGrad.m pseudo code
(Inputs: array (*I*) and *sigma*)
Default *sigma* is 1
Find gauss derivative (*gd*) using **gaussDeriv.m**
    (inputs: *sigma*

outputs: *gd*)

When *sigma*=1, *gd* = [0.0 0.0005 0.0133 0.1080 0.2420 -0.2420 -0.1080 -0.0133 -0.0005 -0.0]

Set z-direction gauss derivative (*zgd*) as an array of size (1:1:length(*gd*)) with the same inputs as *gd*

Find the derivative of the image in the x-direction (*xd*) using the MATLAB function **convn**(*I*, *gd*, 'same')

Find *yd* using the MATLAB function **convn**(*I*, *gd'*, 'same')

Find zd using the MATLAB function **convn**(*I*, *zgd*, 'same')

(Outputs: *xd, yd, zd*)

gaussDeriv.m pseudo code

(Input: sigma)

Initialize *threshold* to 0.000001

Set *x* as a linspace of -10000 to 10000 with 20001 points

Evaluate $gd = -\frac{e^{\left(\frac{-x^2}{2\sigma^2}\right)}}{\sqrt{(\pi 2\sigma^2)}}\frac{x}{\sigma^2}$

Remove any *gd* values less than the threshold

(Output: *gd*)

resolutionProcess.m pseudo code

(Inputs: *Ikt, Ikx2, Iky2, Ikz2, Ixt, Iyt, Izt, alpha, gamma, u initial, v initial, w initial, # outer iterations, # inner iterations*)

Initialize *ht*=number of rows in *Ikt*, *wt* = number of columns in *Ikt*, and *dt*=number of pages in *Ikt*

Initialize *u*, *v*, and *w* to *uinit*, *vinit*, and *winit*

Find derivatives of *Ikx*, *Iky* and *Ikz* in the x, y, and z directions (*Ixx, Ixy, Ixz, Iyx, Iyy, Iyz, Izx, Izy, Izz*) using **imgGrad.m** these are the second derivatives of the second image

(inputs: *Ikx, Iky, Ikz*) *Note use imgGrad separately for each array

(outputs: *Ixx, Ixy, Ixz, Iyx, Iyy, Iyz, Izx, Izy, Izz*)

Initialize *du, dv*, and *dw* to a zero array the size of *Ikt*

For *k*=1 to *# outer iterations*

Find $(\psi')^i_{Data}$ (psidash) using **psiDerivative.m**

(inputs: function: $\left(\left(I_{kt} + I_{kx}du + I_{ky}dv + I_{kz}dw\right)^2 + \gamma\left(\left(I_{xt} + I_{xx}du + I_{xy}dv + I_{xz}dw\right)^2 + \left(I_{yt} + I_{yx}du + I_{yy}dv + I_{yz}dw\right)^2 + \left(I_{zt} + I_{zx}du + I_{zy}dv + I_{zz}dw\right)^2\right)\right)$

Outputs: $(\psi')^k_{Data}$)

Find $(\psi')^k_{Smooth}$ (psidashFS) using **computePsidashFS_brox**

(inputs: *u+du*, *v+dv*, *w+dw*)

(outputs: *psidashFS*)

For *l*=1 to *# inner iterations*

If *l*==1

Initialize change of displacement at the next iteration step, $du^{k,l+1}$, $dv^{k,l+1}$, and $dw^{k,l+1}$to zero arrays the same size as *u*

End If

Find the average displacement field, $\overline{u^k}$, $\overline{v^k}$, and $\overline{w^k}$ (*uAvg*, *vAvg*, and *wAvg*) and the average change of displacement for the next step, $\overline{du^{k,l+1}}$, $\overline{dv^{k,l+1}}$, and $\overline{dw^{k,l+1}}$ (*duplus1Avg*, *dvplus1Avg*, and *dwplus1Avg*) by taking the average of a 7X7X7 neighborhood around each voxel

Find $(\psi')^{k,l+1}_{Data}$ (psidashplus1) using **psiDerivative.m**

(Input: function: $\left(\left(I_{kt} + I_{kx} duplus1 + I_{ky} dvplus1 + I_{kz} dwplus1\right)^2 + \gamma\left(\left(I_{xt} + I_{xx} duplus1 + I_{xy} dvplus1 + I_{xz} dwplus1\right)^2 + \left(I_{yt} + I_{yx} duplus1 + I_{yy} dvplus1 + I_{yz} dwplus1\right)^2 + \left(I_{zt} + I_{zx} duplus1 + I_{zy} dvplus1 + I_{zz} dwplus1\right)^2\right)\right)$

Find the crammer components using **CrammerComponents.m**

(inputs: *Ikx, Iky, Ikz, Ikt, Ixx, Ixy, Ixz, Iyx, Iyy, Iyz, Izx, Izy, Izz, Ixt, Iyt, Izt, psidashplus1, psidash, alpha*psidashFS, u, v, w, gamma*)

(outputs: *A, B, C, D, E, F, G, H, I, j, k, l*)

Evaluate the determinant by the equation:

$$det = \left(A + \alpha(\psi')^{k,l}_{Smooth}\right)\left(EI + E\alpha(\psi')^{k,l}_{Smooth} + I\alpha(\psi')^{k,l}_{Smooth} + \alpha^2 {(\psi')^{k,l}_{Smooth}}^2 - HF\right) - D\left(BI + B\alpha(\psi')^{k,l}_{Smooth} - HC\right) + G\left(BF - CE - C\alpha(\psi')^{k,l}_{Smooth}\right)$$

Evaluate the new $du^{k,l+1}$, $dv^{k,l+1}$, and $dw^{k,l+1}$ (duplus1n, dvplus1n, and dwplus1n) by the equations:

$$\begin{aligned} du^{k,l+1} = {} & \frac{1}{det}\left(\alpha(\psi')^{k,l}_{Smooth}\left(\overline{u^k} + \overline{du^{k,l+1}} - u^k\right) - j\right)\left(\left(E + \alpha(\psi')^{k,l}_{Smooth}\right)\left(I + \alpha(\psi')^{k,l}_{Smooth}\right) - HF\right) \\ & - \frac{1}{det}\left(\alpha(\psi')^{k,l}_{Smooth}\left(\overline{v^k} + \overline{dv^{k,l+1}} - v^k\right) - k\right)\left(B\left(I + \alpha(\psi')^{k,l}_{Smooth}\right) - CH\right) \\ & + \frac{1}{det}\left(\alpha(\psi')^{k,l}_{Smooth}\left(\overline{w^k} + \overline{dw^{k,l+1}} - w^k\right) - l\right)\left(BF + C\left(E + \alpha(\psi')^{k,l}_{Smooth}\right)\right) \end{aligned}$$

$$\begin{aligned} dv^{k,l+1} = {} & \frac{1}{det}\left(\alpha(\psi')^{k,l}_{Smooth}\left(\overline{u^k} + \overline{du^{k,l+1}} - u^k\right) - j\right)\left(D\left(I + \alpha(\psi')^{k,l}_{Smooth}\right) - FG\right) \\ & - \frac{1}{det}\left(\alpha(\psi')^{k,l}_{Smooth}\left(\overline{v^k} + \overline{dv^{k,l+1}} - v^k\right) - k\right)\left(\left(A + \alpha(\psi')^{k,l}_{Smooth}\right)\left(I + \alpha(\psi')^{k,l}_{Smooth}\right) - CG\right) \\ & + \frac{1}{det}\left(\alpha(\psi')^{k,l}_{Smooth}\left(\overline{w^k} + \overline{dw^{k,l+1}} - w^k\right) - l\right)\left(F\left(A + \alpha(\psi')^{k,l}_{Smooth}\right) - CD\right) \end{aligned}$$

$$\begin{aligned} dv^{k,l+1} = {} & \frac{1}{det}\left(\alpha(\psi')^{k,l}_{Smooth}\left(\overline{u^k} + \overline{du^{k,l+1}} - u^k\right) - j\right)\left(DH - G\left(E + \alpha(\psi')^{k,l}_{Smooth}\right)\right) \\ & - \frac{1}{det}\left(\alpha(\psi')^{k,l}_{Smooth}\left(\overline{v^k} + \overline{dv^{k,l+1}} - v^k\right) - k\right)\left(\left(A + \alpha(\psi')^{k,l}_{Smooth}\right)H - BG\right) \\ & + \frac{1}{det}\left(\alpha(\psi')^{k,l}_{Smooth}\left(\overline{w^k} + \overline{dw^{k,l+1}} - w^k\right) - l\right)\left(\left(A + \alpha(\psi')^{k,l}_{Smooth}\right)\left(E + \alpha(\psi')^{k,l}_{Smooth}\right) - BD\right) \end{aligned}$$

If the difference between the new *d**u**plus1* and the old *d**u**plus1* is small enough

Then *duplus1=duplus1n*, *dvplus1=dvplus1n*, and *dwplus1=dwplus1n* and **break**

Else

duplus1=duplus1n, *dvplus1=dvplus1n*, and *dwplus1=dwplus1n*

End If

Set *du=duplus1*, *dv=dvplus1*, and *dw=dw+1*

End For

End For

(Outputs: *du, dv, dw*)

psiDerivative.m pseudo

(Input: a function (*x*))

Initialize *epsilon* to 0.001

$\psi'(x) = \dfrac{1}{2\sqrt{x+\epsilon}}$

(Output: $\psi'(x)$)

computePsidashFS_brox.m pseudo code

(Input: *u, v, w*)

Find the derivatives of *u*, *v*, and *w* in the *x*, *y*, and *z* directions (*ux, uy, uz, vx, vy, vz, wx, wy, wz*) using the MATLAB function **convn**

Find *psidashFS* using the function **psiDerivative.m** on the input function ${u_x}^2 + {u_y}^2 + {u_z}^2 + {v_x}^2 + {v_y}^2 + {v_z}^2 + {w_x}^2 + {w_y}^2 + {w_z}^2$

(Output: *psidashFS*)

CrammerComponents.m pseudo code

(Input: *Ikx, Iky, Ikz, Ikt, Ixx, Ixy, Ixz, Iyx, Iyy, Iyz, Izx, Izy, Izz, Ixt, Iyt, Izt, psidashplus1, psidash, alpha*psidashFS, u, v, w, gamma*)

A= $(\psi')_{Data}^{k,l+1}\left(I_{kx}I_{kx} + \gamma\big(I_{xx}I_{xx} + I_{xy}I_{yx} + I_{xz}I_{zx}\big)\right)$

B= $(\psi')_{Data}^{k,l+1}\left(I_{kx}I_{ky} + \gamma\big(I_{xx}I_{xy} + I_{xy}I_{yy} + I_{xz}I_{zy}\big)\right)$

C= $(\psi')_{Data}^{k,l+1}\left(I_{kx}I_{kz} + \gamma\big(I_{xx}I_{xz} + I_{xy}I_{yz} + I_{xz}I_{zz}\big)\right)$

D= $(\psi')_{Data}^{k,l+1}\left(I_{ky}I_{kx} + \gamma\big(I_{yx}I_{xx} + I_{yy}I_{yx} + I_{yz}I_{zx}\big)\right)$

E= $(\psi')_{Data}^{k,l+1}\left(I_{ky}I_{ky} + \gamma\big(I_{yx}I_{xy} + I_{yy}I_{yy} + I_{yz}I_{zy}\big)\right)$

F= $(\psi')_{Data}^{k,l+1}\left(I_{ky}I_{kz} + \gamma\big(I_{yx}I_{xz} + I_{yy}I_{yz} + I_{yz}I_{zz}\big)\right)$

G= $(\psi')_{Data}^{k,l+1}\left(I_{kz}I_{kx} + \gamma\big(I_{zx}I_{xx} + I_{zy}I_{yx} + I_{zz}I_{zx}\big)\right)$

H= $(\psi')_{Data}^{k,l+1}\left(I_{kz}I_{ky} + \gamma\big(I_{zx}I_{xy} + I_{zy}I_{yy} + I_{zz}I_{zy}\big)\right)$

I= $(\psi')_{Data}^{k,l+1}\left(I_{kz}I_{kz} + \gamma\big(I_{zx}I_{xz} + I_{zy}I_{yz} + I_{zz}I_{zz}\big)\right)$

j= $(\psi')_{Data}^{k,l}\left(I_{kx}I_{kt} + \gamma\big(I_{xx}I_{xt} + I_{xy}I_{yt} + I_{xz}I_{zt}\big)\right)$

k= $(\psi')_{Data}^{k,l}\left(I_{ky}I_{kt} + \gamma\big(I_{yx}I_{xt} + I_{yy}I_{yt} + I_{yz}I_{zt}\big)\right)$

l= $(\psi')_{Data}^{k,l}\left(I_{kz}I_{kt} + \gamma\big(I_{zx}I_{xt} + I_{zy}I_{yt} + I_{zz}I_{zt}\big)\right)$

(Output: A, B, C, D, E, F, G, H, I, j, k, l)

8.1.4 Demons

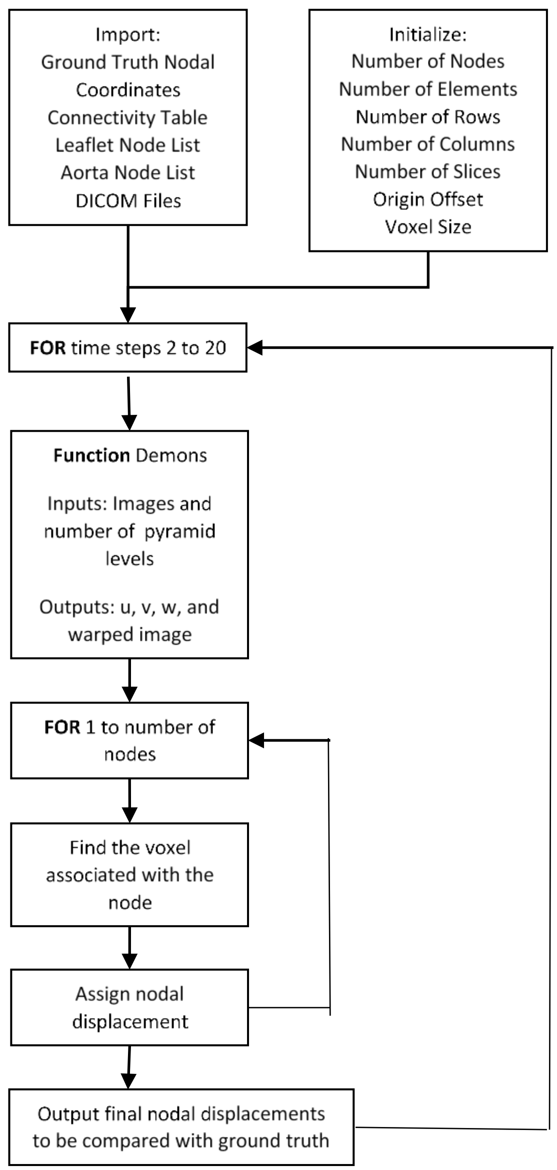


Figure 80:RunDemon3D.m Block Diagram

RunDemon3D.m

Initialize the number of *nodes*, and *elements* in FEA ground truth model
Import FEA ground truth *node coordinates* for each of the 20 time steps
Import the *Connectivity Table* of the ground truth FEA model
Import the *Leaflet Nodes List* and the *Aorta Nodes List*
Initialize number of *rows, columns* and *slices* expected in DICOM images
Import *DICOM Files*
Convert *DICOM Files* to double variable arrays
Initialize the images *Origin Offset* between FEA and DICOM files (based on Master Volume of 3DSlicer)
Initialize *Voxel Size* (based on Master Volume of 3DSlicer)

Assign *first index* as 1
For *second index* from 2 to 20 (% number of timesteps in DICOM set)
Assign the *first image* as the *first index* DICOM Image
Assign the *second image* as the *second index* DICOM Image
Initialize *number of levels* for the pyramid scheme to 10
Function **demon.m**
(Inputs: *first image, second image, number of levels*
Outputs: *u, v, w, Iwarped*)
End Function
Create cropped images, *first image leaflets* and *second image leaflets,* from *first image* and *second image* that only contain the rows encompassing the valve movement
Function **demon.m**
(Inputs: *first image leaflets, second image leaflets, number of levels*
Outputs: *u leaflets, v leaflets, w leaflets, Iwarped*)
End Function
For i=1 to *nodes*
Calculate the *row*, *column*, and *depth* of the node *i* from the FEA model by converting from mm to voxels using the *voxel size* and *voxel offset*
If *image 1* at the *row, column, and depth* has a value that is not zero (a brightness intensity greater than zero represents a non-black square)
Assign *nodal displacements* row *i* columns 1, 2 and 3 the *u, v,* and *w* corresponding to the *row, column,* and *depth* calculated for node *i*
Else If *image 1* at the *row, column, and depth* has a value of zero
Use function **neighbour342.m** [32] to determine the neighboring voxels with 3 voxels in any direction of the voxel the node of interest sits in, for a total of 342 voxels
(Inputs: *row, column, depth, Image 1*
Outputs: List of rows, columns, and pages of each of the 342 "neighboring" voxels, *neighbours*)
End Function
Initialize an arbitrary *distance* greater than the 3-voxel perimeter, ex. 10
Initialize *vox* to zero
Initialize a counter, *sum,* to zero
For *ii*=1 to length of *neighbours*
If The rows, columns, and pages coordinate of the *ii* line of *neighbours* exist in the dimensions of the image size (greater than zero, and less than *rows, columns* and *slices* respectively)
If The first image voxel coordinate given by *neighbours* does not equal zero
Find the *Euclidean distance* between node *i* and the center of the neighbouring voxel *ii* in the unit of voxels
Add 1 to the *sum* counter
If *Euclidean distance* is less than *distance*
Set *distance* to *Euclidean distance*
Set *vox* to *ii*
End If
End If
End If

End For
If *sum* counter is less than 1
Set *nodal displacements* row *i,* columns 1, 2, and 3 to zero (in this case there are no neighbouring voxels of the node *i* within the boundary that are part of the aorta)
Else
Set *nodal displacements* row *i,* columns 1, 2, and 3 to *u, v,* and *w* corresponding to the *neighbours* row *vox* voxel location (the displacement of the closest voxel to the node of interest)
End If
End If
End For
Assign *new node displacement aorta* variable *nodal displacements* converted back into mm to be compared with ground truth
Assign *new node coordinate locations aorta* variable the calculated new node coordinate location (ground truth *Node Coordinate* timestep one + *new node displacement aorta*)
For i=1 to *nodes*
Calculate the *row*, *column*, and *depth* of the node *i* from the FEA model by converting from mm to voxels using the *voxel size* and *voxel offset*
If *image 1* at the *row, column, and depth* has a value that is not zero (a brightness intensity greater than zero represents a non-black square)
Assign *nodal displacements* row *i* columns 1, 2 and 3 the *u leaflet, v leaflet,* and *w leaflet* corresponding to the *row, column,* and *depth* calculated for node *i*
Else If *image 1* at the *row, column, and depth* has a value of zero
Use function **neighbour342.m** [32] to determine the neighboring voxels with 3 voxels in any direction of the voxel the node of interest sits in, for a total of 342 voxels
(Inputs: *row, column, depth, Image 1*
Outputs: List of rows, columns, and pages of each of the 342 "neighboring" voxels, *neighbours*)
End Function
Initialize an arbitrary *distance* greater than the 3-voxel perimeter, ex. 10
Initialize *vox* to zero
Initialize a counter, *sum,* to zero
For *ii*=1 to length of *neighbours*
If The rows, columns, and pages coordinate of the *ii* line of *neighbours* exist in the dimensions of the image size (greater than zero, and less than *rows, columns* and *slices* respectively)
If The first image voxel coordinate given by *neighbours* does not equal zero
Find the *Euclidean distance* between node *i* and the center of the neighbouring voxel *ii* in the unit of voxels
Add 1 to the *sum* counter
If *Euclidean distance* is less than *distance*
Set *distance* to *Euclidean distance*
Set *vox* to *ii*
End If
End If

End If
End For
If *sum* counter is less than 1
Set *nodal displacements* row *i,* columns 1, 2, and 3 to zero (in this case there are no neighbouring voxels of the node *i* within the boundary that are part of the aorta)
Else
Set *nodal displacements* row *i,* columns 1, 2, and 3 to *u leaflet, v leaflet,* and *w leaflet* corresponding to the *neighbours* row *vox* voxel location (the displacement of the closest voxel to the node of interest)
End If
End If
End For
Assign *new node displacement leaf* variable *nodal displacements* converted back into mm to be compared with ground truth
Assign *new node coordinate locations leaf* variable the calculated new node coordinate location (ground truth *Node Coordinate* timestep one + *new node displacement leaf*)
For *p*=1:*nodes*
Assign *Final Nodal Displacements* page *second index* the *new node displacement aorta*
For *q*=1 to the length of the *leaflet nodes* list
If node number *p* is on the *leaflet nodes* list
Reassign *Final Nodal Displacements* row *p* to *new node displacement leaf*
End If
End For
End For
Output the statistics to compare the demons algorithm nodal displacements to the ground truth nodal displacements
Create STL file using demons nodal displacements
End For

demon.m

(Inputs: *image 1, image 2, number of pyramid levels*)
initialize *scale constant*
Rebuild the *images 1 and 2* at the coarsest level (least number of voxels) using the function **gaussianRescaling** using a scale factor of *scale constant* ^ *number of pyramid levels*
(Inputs: *Image* to be rescaled, *scale factor*
Outputs: *Image* scaled to specified size)
End Function
Initialize the displacement field; *u, v, and w* as an array of zeros the size of the coarsest image
Initialize a total displacement field; *Utot, Vtot, Wtot* as an array of zeros the size of the coarsest image
For *i=number of pyramid levels*-1 by increments of -1 to 0
Use function **imregdemons** where image 1 is set as the fixed image, image 2 is set as the moving image, the number of iterations per pyramid level is set to 500, the number of pyramid levels is set to 1, and the accumulated field smoothing is set to 1
Set *du*, *dv*, and *dw* to the corresponding values of the displacement field *D*

Use the function **gaussianRescaling** to resize *images 1 and 2* using a scale factor of *scale constant ^ i* (Increase size of images by one level in the pyramid)
Use **imresize3** to resize *du*, *dv*, and *dw* to the size of the *images* on the current level using nearest neighbour interpolation
Use **imresize3** to resize *utot*, *vtot*, and *wtot* to the size of the *images* on the current level using nearest neighbour interpolation
Reassign *utot*, *vtot*, and *wtot* to *utot+du*, *vtot+dv*, *wtot+dw* respectively
Scale *utot*, *vtot*, and *wtot* by a factor of 1/*scale factor*
Use **imwarp** to warp *image 2* using a displacement field made of *utot*, *vtot*, and *wtot*
Assign *image 2* to the warped image

End For

(Outputs: *utot*, *vtot*, *wtot,* final warped *image 2*)

8.2 Raw Data of Error Metrics – Tables

8.2.1 Difference Error

Table 6: Difference between nodal displacement determined by motion estimation methods and ground truth nodal displacement

	Difference Error of nodal displacement determined by motion estimation methods compared to ground truth nodal displacement								
Time Step	Horn-Schunck Optical Flow Mean ± Standard Deviation (mm)			Brox Optical Flow Mean ± Standard Deviation (mm)			Demons Method Mean ± Standard Deviation (mm)		
	All Nodes	Aorta Nodes	Leaflet Nodes	All Nodes	Aorta Nodes	Leaflet Nodes	All Nodes	Aorta Nodes	Leaflet Nodes
2	**0.45±0.31**	**0.45±0.31**	**0.48±0.28**	0.83±0.56	0.7±0.41	1.4±0.79	0.88±0.57	0.89±0.6	0.81±0.4
3	**0.52±0.3**	**0.53±0.31**	**0.47±0.23**	0.93±0.56	0.81±0.45	1.5±0.66	0.92±0.64	0.93±0.67	0.85±0.42
4	**0.42±0.24**	**0.44±0.25**	**0.29±0.13**	0.6±0.34	0.57±0.35	0.73±0.27	0.91±0.57	0.93±0.6	0.8±0.36
5	0.79±0.55	0.84±0.57	0.54±0.29	**0.53±0.35**	**0.53±0.37**	**0.48±0.27**	0.87±0.57	0.91±0.59	0.67±0.34
6	0.83±0.45	0.87±0.46	0.62±0.27	**0.51±0.34**	**0.53±0.34**	**0.42±0.34**	0.88±0.54	0.91±0.57	0.75±0.34
7	1.15±0.64	1.06±0.53	**1.62±0.91**	1.04±1.1	**0.67±0.45**	2.86±1.46	**0.95±0.79**	0.78±0.57	1.85±1.14
8	1.37±0.97	1.2±0.75	**2.26±1.42**	1.86±1.65	1.25±0.69	4.84±1.73	**1.14±1.42**	**0.85±0.74**	2.74±2.65
9	**1.1±1.02**	**0.89±0.56**	**2.27±1.82**	2.42±1.85	1.77±1.01	5.58±1.79	1.36±1.77	0.93±0.73	3.71±3.26
10	**1.44±1.01**	1.27±0.78	**2.4±1.49**	2.57±1.95	1.94±1.2	5.99±1.69	1.44±1.81	**0.98±0.74**	3.96±3.25
11	**1.42±1**	1.24±0.75	**2.39±1.48**	2.61±1.94	1.97±1.16	6.03±1.69	1.43±1.81	**0.96±0.7**	3.99±3.25
12	**1.06±1.05**	**0.82±0.52**	**2.36±1.9**	2.61±1.93	1.98±1.16	6.02±1.68	1.41±1.72	0.97±0.68	3.82±3.11
13	**1.07±1**	**0.85±0.55**	**2.23±1.8**	2.6±1.92	1.99±1.22	5.92±1.65	1.35±1.55	0.95±0.68	3.49±2.73
14	**1.08±0.99**	**0.88±0.55**	**2.18±1.8**	2.51±1.87	1.9±1.1	5.84±1.66	1.34±1.59	0.95±0.67	3.5±2.87
15	1.41±0.97	1.25±0.76	**2.31±1.42**	2.44±1.82	1.83±1.02	5.73±1.67	**1.34±1.64**	**0.94±0.73**	3.5±2.98
16	1.41±0.96	1.25±0.76	**2.29±1.37**	2.35±1.8	1.75±0.98	5.63±1.66	**1.3±1.58**	**0.95±0.74**	3.23±2.95
17	1.38±0.94	1.22±0.75	**2.24±1.33**	2.28±1.78	1.68±0.97	5.53±1.68	**1.26±1.51**	**0.92±0.74**	3.09±2.8
18	1.37±0.94	1.21±0.75	**2.26±1.32**	2.18±1.79	1.58±0.97	5.43±1.69	**1.24±1.47**	**0.89±0.69**	3.14±2.66
19	1.4±0.97	1.22±0.77	**2.36±1.33**	2.04±1.72	1.46±0.9	5.19±1.7	**1.15±1.4**	**0.85±0.68**	2.8±2.62
20	**1.16±0.9**	0.99±0.59	**2.07±1.52**	2.01±1.72	1.44±0.92	5.12±1.71	1.17±1.36	**0.87±0.68**	2.84±2.47

8.2.2 Absolute Error

Table 7: Absolute Error between magnitude of nodal displacement using motion estimation methods and the ground truth nodal displacement

	Absolute error of nodal displacement determined by motion estimation methods compared to ground truth nodal displacement								
Time Step	Horn-Schunck Optical Flow Mean ± Standard Deviation (mm)			Brox Optical Flow Mean ± Standard Deviation (mm)			Demons Method Mean ± Standard Deviation (mm)		
	All Nodes	Aorta Nodes	Leaflet Nodes	All Nodes	Aorta Nodes	Leaflet Nodes	All Nodes	Aorta Nodes	Leaflet Nodes
2	**0.22±0.18**	**0.23±0.19**	**0.18±0.13**	0.44±0.47	0.34±0.31	0.92±0.75	0.4±0.33	0.41±0.32	0.35±0.33
3	**0.26±0.23**	**0.26±0.23**	**0.25±0.22**	0.49±0.48	0.39±0.34	0.98±0.69	0.52±0.53	0.52±0.54	0.53±0.45
4	**0.2±0.18**	0.22±0.19	**0.1±0.08**	**0.2±0.18**	**0.16±0.13**	0.4±0.26	0.48±0.43	0.51±0.45	0.31±0.23
5	0.44±0.4	0.46±0.42	0.32±0.28	**0.18±0.15**	**0.18±0.15**	**0.2±0.14**	0.49±0.44	0.52±0.45	0.37±0.32
6	0.47±0.36	0.49±0.37	0.39±0.29	**0.22±0.23**	**0.23±0.21**	**0.22±0.31**	0.5±0.42	0.5±0.43	0.45±0.35
7	0.71±0.53	0.69±0.47	**0.82±0.75**	0.57±0.99	**0.24±0.27**	2.17±1.52	**0.43±0.52**	0.35±0.34	0.9±0.92
8	0.86±0.67	0.83±0.62	**1.01±0.88**	0.96±1.34	0.49±0.44	3.27±1.8	**0.49±0.57**	**0.39±0.44**	1.02±0.87
9	0.67±0.74	0.54±0.42	**1.38±1.41**	1.3±1.55	0.77±0.74	3.87±1.9	**0.58±0.8**	**0.42±0.45**	1.44±1.46
10	0.92±0.73	0.89±0.67	**1.09±0.99**	1.41±1.67	0.88±0.91	4.27±1.91	**0.63±0.87**	**0.46±0.52**	1.56±1.55
11	0.9±0.72	0.87±0.66	**1.07±0.98**	1.43±1.65	0.9±0.87	4.32±1.88	**0.65±0.88**	**0.47±0.51**	1.61±1.57
12	0.67±0.78	0.52±0.4	1.52±1.49	1.45±1.66	0.91±0.88	4.37±1.86	**0.64±0.84**	**0.48±0.51**	**1.52±1.48**
13	0.66±0.72	0.54±0.43	**1.32±1.35**	1.43±1.65	0.91±0.92	4.24±1.87	**0.65±0.88**	**0.48±0.57**	1.59±1.44
14	0.67±0.7	0.56±0.43	**1.25±1.31**	1.38±1.61	0.85±0.81	4.24±1.86	**0.64±0.82**	**0.5±0.55**	1.42±1.4
15	0.92±0.7	0.9±0.66	**1.05±0.91**	1.34±1.58	0.81±0.73	4.22±1.84	**0.65±0.77**	**0.5±0.55**	1.46±1.19
16	0.92±0.7	0.9±0.65	**1.06±0.92**	1.3±1.57	0.76±0.69	4.19±1.86	**0.63±0.75**	**0.51±0.58**	1.28±1.13
17	0.89±0.69	0.86±0.63	**1.04±0.92**	1.24±1.53	0.72±0.67	4.03±1.85	**0.62±0.76**	**0.49±0.59**	1.3±1.12
18	0.87±0.69	0.83±0.63	**1.06±0.95**	1.19±1.54	0.68±0.67	3.99±1.87	**0.61±0.79**	**0.47±0.58**	1.41±1.2
19	0.84±0.69	0.8±0.62	**1.1±0.94**	1.09±1.45	0.61±0.58	3.74±1.84	**0.53±0.63**	**0.41±0.46**	1.16±0.93
20	0.65±0.59	0.59±0.43	**1±1.03**	1.06±1.4	0.59±0.58	3.57±1.83	**0.59±0.76**	**0.46±0.57**	1.31±1.14

8.2.3 Relative Percent Difference

Table 8: Relative percent difference between magnitude of nodal displacement using motion estimation methods and the ground truth nodal displacement

	Relative percent difference of nodal displacement magnitudes determined by motion estimation methods compared to ground truth magnitudes of nodal displacement								
Time Step	Horn-Schunck Optical Flow Mean ± Standard Deviation (%)			Brox Optical Flow Mean ± Standard Deviation (%)			Demons Method Mean ± Standard Deviation (%)		
	All Nodes	Aorta Nodes	Valve Nodes	All Nodes	Aorta Nodes	Valve Nodes	All Nodes	Aorta Nodes	Valve Nodes
2	**37.7±50.4**	**42.2±53.3**	**13.1±14.8**	60.9±52.2	60.7±55	62.1±35.7	52.5±52.3	58±54.3	22.8±22.9
3	**36.3±48.4**	**40.3±51.4**	**14.1±11.4**	60.3±51.2	58.5±53.5	69.3±37.6	52±51.6	56.7±54	26.8±22.3
4	**42.2±50.2**	47.4±52.6	**13.7±13.4**	46.5±49.3	**43.8±51.7**	59.9±33.5	65±52.4	70.4±54.1	35.9±28.3
5	66.2±53.6	69.8±55.1	46.8±39.4	**46.3±49.7**	**47.7±52.7**	**40.3±31.9**	71.6±55	75.4±56.3	51.1±42.1
6	57.5±52	62.1±54.5	32.3±22.8	**43.2±50.7**	**47.7±53.7**	**21.3±22.1**	59.1±52.7	63.4±55.2	36.2±26.1
7	62.9±48.4	69.9±48.6	**24.8±23**	56.9±55.3	**48.3±53.7**	99.3±42.1	**45.6±48.7**	49.4±51	25±25
8	52.5±46.2	58.6±47.1	19.1±17.9	60.2±53.7	53.8±53.6	92.4±41.8	**38.1±48.2**	**41.8±51.3**	**18.2±14.3**
9	39.1±45.8	42±48.4	**23.2±21.3**	65.9±53.9	59.8±54.5	96.3±39.5	**36.9±49**	**38.9±51.6**	25.8±29.3

10	47±46.1	52.4±47.6	**17.3±16.5**	67.6±54	61.1±54	102.5±38.4	**37.7±49.2**	**39.6±51.7**	27.3±30.5
11	46±46.2	51.3±47.9	**17±16.3**	69.2±53.9	62.8±54	103.8±37.9	**38.3±49.1**	**40.2±51.5**	28.2±31
12	**38±46.3**	**40.3±49.1**	**25.1±22.6**	70.4±54.5	63.8±54.6	105.9±37.6	38.2±49	40.5±51.6	25.8±28.5
13	38.9±46.6	42.1±49.3	**21.8±20.2**	69.5±54.4	63.2±54.6	103.5±38.1	**37.8±48.6**	**40.1±51.5**	25.3±24.5
14	39.6±46.3	43.1±48.9	**20.8±19.6**	68.8±54.6	62.1±54.5	105.2±38.2	**38.8±49**	**41.7±51.7**	23.4±25.2
15	49±46.5	54.8±47.9	**17.2±15.6**	68.9±55.2	61.8±54.8	106.9±39.2	**39.5±49**	**42.4±52.1**	24.2±20.8
16	49.8±46.7	55.8±48.1	**17.4±15.7**	67.8±55.7	60.5±55.1	107.3±39.9	**40±49.8**	**43.5±52.9**	21±19
17	49.5±46.8	55.4±48.2	**17.4±15.9**	65.3±55.5	58.4±55.1	103.1±40.1	**39.6±50**	**43±53.2**	20.9±17.6
18	49±47	54.8±48.6	**17.8±16.4**	63.6±55.6	56.3±54.9	103.2±40.3	**39.2±50**	**42.2±53.2**	22.4±18.7
19	48.9±47.3	54.4±49	**18.9±16.8**	59.6±55.5	52.7±55.1	97.2±41.2	**38.4±50.5**	**41.9±53.9**	19.3±14.8
20	43.9±47.7	48.7±50	**17.8±16.4**	58.3±55.3	52±55.3	92.1±41.5	**40.5±51.7**	**43.9±55**	21.6±18

8.2.4 Angular Error

Table 9: Angular error between nodal displacement using motion estimation methods and the ground truth nodal displacement

	Angular error of nodal displacement determined by motion estimation methods compared to ground truth nodal displacement								
Time Step	Horn-Schunck Optical Flow Mean ± Standard Deviation			Brox Optical Flow Mean ± Standard Deviation			Demons Method Mean ± Standard Deviation		
	All Nodes	Aorta Nodes	Valve Nodes	All Nodes	Aorta Nodes	Valve Nodes	All Nodes	Aorta Nodes	Valve Nodes
2	**0.28±0.18**	**0.29±0.19**	**0.22±0.13**	0.53±0.27	0.5±0.25	0.68±0.31	0.54±0.33	0.57±0.34	0.36±0.19
3	**0.3±0.16**	**0.32±0.17**	**0.19±0.08**	0.57±0.25	0.54±0.25	0.72±0.2	0.48±0.29	0.51±0.3	0.28±0.12
4	**0.32±0.18**	**0.34±0.18**	**0.2±0.11**	0.48±0.25	0.47±0.26	0.54±0.19	0.62±0.33	0.63±0.33	0.56±0.29
5	0.55±0.31	0.58±0.31	**0.39±0.2**	**0.44±0.27**	**0.44±0.28**	0.4±0.23	0.61±0.31	0.63±0.32	0.5±0.24
6	0.51±0.24	0.54±0.24	0.32±0.12	**0.37±0.23**	**0.4±0.23**	**0.26±0.17**	0.54±0.28	0.56±0.29	0.39±0.16
7	0.56±0.25	0.59±0.25	**0.36±0.15**	0.6±0.37	**0.51±0.31**	1.03±0.37	**0.5±0.3**	0.52±0.31	0.4±0.18
8	0.45±0.22	0.48±0.21	**0.33±0.18**	0.82±0.39	0.74±0.33	1.23±0.42	**0.44±0.33**	**0.46±0.32**	0.38±0.33
9	**0.34±0.2**	**0.35±0.2**	**0.29±0.22**	0.92±0.38	0.85±0.34	1.23±0.38	0.46±0.35	0.45±0.31	0.55±0.5
10	**0.39±0.23**	**0.41±0.24**	**0.29±0.17**	0.93±0.37	0.87±0.34	1.26±0.37	0.46±0.35	0.44±0.3	0.58±0.51
11	**0.39±0.23**	**0.4±0.24**	**0.28±0.17**	0.95±0.37	0.89±0.33	1.27±0.37	0.45±0.34	0.43±0.28	0.58±0.52
12	**0.31±0.2**	**0.31±0.19**	**0.29±0.23**	0.95±0.37	0.89±0.33	1.27±0.37	0.45±0.33	0.43±0.29	0.54±0.49
13	**0.31±0.2**	**0.32±0.19**	**0.29±0.22**	0.95±0.37	0.89±0.33	1.26±0.37	0.42±0.3	0.42±0.29	0.42±0.37
14	**0.32±0.2**	**0.33±0.19**	**0.29±0.22**	0.94±0.36	0.88±0.33	1.26±0.37	0.42±0.29	0.41±0.27	0.46±0.4
15	**0.39±0.22**	**0.41±0.22**	**0.29±0.17**	0.93±0.36	0.87±0.33	1.24±0.36	0.42±0.29	0.41±0.27	0.45±0.4
16	**0.4±0.22**	**0.42±0.22**	**0.29±0.16**	0.9±0.36	0.84±0.32	1.23±0.36	0.41±0.29	0.42±0.27	0.41±0.39
17	**0.4±0.22**	0.42±0.22	**0.29±0.15**	0.87±0.36	0.81±0.32	1.21±0.37	0.41±0.28	**0.41±0.27**	0.38±0.35
18	0.41±0.22	0.43±0.22	**0.29±0.15**	0.83±0.36	0.77±0.31	1.19±0.38	**0.4±0.28**	**0.41±0.26**	0.37±0.33
19	0.43±0.22	0.45±0.22	**0.31±0.16**	0.78±0.36	0.71±0.31	1.15±0.39	**0.4±0.28**	**0.41±0.26**	0.35±0.33
20	**0.39±0.2**	**0.4±0.19**	**0.31±0.21**	0.76±0.37	0.7±0.32	1.14±0.41	0.4±0.26	0.41±0.25	0.34±0.31