INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN BIOMEDICAL ENGINEERING
Int. J. Numer. Meth. Biomed. Engng. 2016; **00**:1–22
Published online in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/cnm

Numerical Methods for Polyline-to-Point-Cloud Registration with Applications to Patient-Specific Stent Reconstruction†

Claire Yilin Lin[1], Alessandro Veneziani[2,3], Lars Ruthotto[2*]

[1] *Department of Mathematics, University of Michigan,* [2] *Department of Mathematics and Computer Science, Emory University,* [3] *School of Advanced Studies IUSS Pavia*

SUMMARY

We present novel numerical methods for Polyline-to-Point-Cloud Registration and their application to patient-specific modeling of deployed coronary artery stents from image data. Patient-specific coronary stent reconstruction is an important challenge in computational hemodynamics and relevant to the design and improvement of the prostheses. It is an invaluable tool in large-scale clinical trials that computationally investigate the effect of new generations of stents on hemodynamics and eventually tissue remodeling. Given a point cloud of strut positions, which can be extracted from images, our stent reconstruction method aims at finding a geometrical transformation that aligns a model of the undeployed stent to the point cloud. Mathematically, we describe the undeployed stent as a polyline, which is a piecewise linear object defined by its vertices and edges. We formulate the nonlinear registration as an optimization problem whose objective function consists of a similarity measure, quantifying the distance between the polyline and the point cloud, and a regularization functional, penalizing undesired transformations. Using projections of points onto the polyline structure, we derive novel distance measures. Our formulation supports most commonly used transformation models including very flexible nonlinear deformations. We also propose two regularization approaches ensuring the smoothness of the estimated nonlinear transformation. We demonstrate the potential of our methods using an academic 2D example and a real-life 3D bioabsorbable stent reconstruction problem. Our results show that the registration problem can be solved to sufficient accuracy within seconds using only a few number of Gauss-Newton iterations.

Received . . .

KEY WORDS: Registration, Computational Hemodynamics, Stent Reconstruction, Point Cloud Processing, Polyline Registration, Patient-Specific Modeling

1. INTRODUCTION

In this paper, we present novel numerical methods for the registration of polylines and point clouds algorithms and their use to generate patient-specific models of coronary artery stents. To this end, we design novel distance and regularization functions to be used in the flexible registration framework FAIR [1]. Our method computes a geometrical transformation that aligns a polyline, i.e., a piecewise linear object, to a point cloud.

One of the driving motivations of this work is to improve patient-specific computational modeling of stented coronary arteries. Despite an active debate about the choice of the most appropriate therapy, Percutaneous Coronary Interventions (PCI) have become increasingly popular due to

*Correspondence to: 400 Dowman Dr. NE, Atlanta (GA) USA, 30322, E-mail: lruthotto@emory.edu
†This work has been partially supported by ABBOTT Laboratories. The first author performed this work as Honor Student of the Program of Applied Mathematics at Emory University. The third author's work is supported in part by National Science Foundation (NSF) grant DMS 1522599.

limited invasiveness. PCI implies the deployment of a prosthesis (called *stent*) [2], generally made of biocompatible materials with a metallic core, to open a coronary artery with severe occlusions. Next generation stents feature new bioabsorbable materials (generally absorbed within three years) and are targeted for acute pathologies in young patients [3]. The different mechanical properties of those materials require thicker struts to handle the pressure during and after the deployment. Thicker struts may interfere with the blood flow and eventually trigger biological processes and tissue remodeling with negative outcomes for the patient (reocclusion) [4, 5, 6, 7]. This is why an accurate assessment of the effect of the struts on the hemodynamics in patient-specific scenarios is of utmost importance. To this end the reconstruction of geometries for extensive fluid dynamics simulations based on clinical data and images is needed [8, 9, 10, 11, 12, 13, 14, 15]. An excellent overview of computational modeling of stented arteries, and comparison of imaging modalities used in this application is given in [16].

Computational Fluid Dynamics (CFD) is the tool of choice for this kind of investigations [17, 18, 19, 20, 21, 22], as it allows personalized quantitative analysis with a modest invasiveness for the patient. In particular, we target a fine analysis of the Wall Shear Stress (WSS - i.e. the tangential component of the normal stress) induced by the blood flow on the struts and the tissue [14]. The reliability of the results strongly depends on a precise patient-specific reconstruction of the stent and the lumen after deployment. Developing efficient (i.e., automatic or semi-automatic) methods for stent reconstruction is critical, for example, when processing a statistically significant number of patient datasets in large-scale clinical studies aiming at quantifying the effectiveness of the therapy [14, 23]. The accuracy and the efficiency of the reconstruction are challenged by the complexity of the sequence of steps and the large variability of cases in diverse patient-specific settings. We give a short description of the procedure currently developed in the Emory University Hospital in Sect. 6.1.

To properly reduce patient variability and to improve the automation procedure, it is critical to guide the reconstruction with prior information available from the design of the stent. In fact, there are some practical limitations on patient-specific data. For instance, OCT images cannot resolve the entire vascular section (see Fig. 6) due to the shadow of the catheter. Therefore, a circular (section-dependent) sector is missing in each image. To compensate the missing data, the information provided by the design of the stent provide a ground truth to guide the patient-specific reconstruction in an accurate and highly automated way. This requires to identify a map between the undeployed and the deployed geometries so that the missing information in the latter can be recovered by the mapping of the former one.

This map can be calculated by a *virtual deployment*, i.e. a simulated operation mimicking the act of deployment. This can be done by a series of Boolean operations [24] or by mechanical simulation of the expansion [25, 26, 27]. While these approaches have great potential, the lack of knowledge of the mechanical properties of the wall to be used in the virtual deployment may be critical.

In this work, we privilege a more data-driven approach, related to *registration procedures*. According to a similar guideline, in [28] the OCT-based stent-reconstruction is guided by an educated combination of *a priori* information on the stent design. In fact, the undeployed stent is registered to the point cloud of strut locations using a non-rigid point-to-point registration procedure [29]. The procedure is tested on one case of a metallic stent in a porcine artery.

The key contributions of the present paper are to represent the stent efficiently as a polyline (defined in [30] as piecewise linear objects consisting of vertices and edges) and develop new numerical methods for registering polylines to point clouds. In the context of stent reconstruction, we assume that the elements of the point cloud (e.g., strut positions detected in OCT images) represent post-deployment points of the polyline (e.g., model of the undeployed stent), but the correspondence is unknown. Our goal is to establish the map by geometrically deforming the polyline object such that its distance to the point cloud is minimized. We exploit the polyline structure to compute the correspondence between the polyline and a given point, by projection onto the edges of the polyline. This assignment is (almost everywhere) differentiable with respect to the deformation and derivatives are easy to compute, thus, enabling fast optimization. Registration is known to be a challenging and ill-posed inverse problem and tailored approaches have been

Prepared using cnmauth.cls
DOI: 10.1002/cnm

developed for registering images [31, 32, 33, 1, 34], curves [35, 36] surfaces [37, 38, 39], point sets [40, 41], or polyline objects [30].

We present two novel regularization approaches tailored to the nonlinear polyline-to-point-cloud registration problem. As in most applications of registration, regularization is of paramount importance to address the ill-posedness and under-determinedness of the problem; see also [42] for a discussion on ill-posedness of the related image registration problem. Our first approach discretizes the transformation directly on the vertices of the polyline object. In this way, very complex deformations can be achieved, but tailored regularization is required. We propose two regularizers that enforce smoothness and favor length-preserving transformations in order to improve robustness against noise and enforce plausible solutions. This formulation often leads to fewer degrees of freedom that need to be optimized. We demonstrate that combining these two ideas allows highly accurate nonlinear registration in the presence of noisy data. Our second approach collocates the transformation on a regular grid and deform the stent indirectly by interpolation. Regularity of the transformation can be enforced by choosing a relatively coarse grid. Additionally, we propose using state-of-the-art regularization techniques, for example, based on nonlinear elasticity [43, 34] to guarantee invertibility of the computed transformation.

The paper is organized as follows. In Section 2 we mathematically define the polyline object as an undirected graph, introduce transformations, distance measures, and derive the optimization problem. Section 3 presents to novel regularization techniques for nonlinear polyline registration. The potential of our method is demonstrated with numerical examples in Section 5. A brief summary of our pipeline used to reconstruct 3D stents for CFD analysis and the contribution of the proposed method in this context are provided in Section 6. Finally, an extensive discussion of the main features and limitations of our methodology with comparison with the existing literature is given in Section 7, together with some conclusive statements.

2. MATHEMATICAL FORMULATION

In this section, we provide a general mathematical formulation of the polyline registration problem. We define polylines, their transformations, the distance of a point to the polyline, the optimization problem, its solution through the Gauss-Newton method, and compute the derivative of the distance with respect to the transformation applied to the polyline object. We finally discuss heuristic methods for outlier rejection.

2.1. Polylines and Point Clouds

Following [30] we define a polyline as an object that consists of a number line segments, described by vertices and edges. Let d denote the spatial dimension (here $d \in \{2, 3\}$) and let n_v be th number of vertices of the polyline object. Then we rearrange the vertices $v_1, v_2, \ldots, v_{n_v} \in \mathbb{R}^d$ into a vector of length $d \cdot n_v$, by listing in order the elements in each of the d components,

$$\mathbf{v} = (v_1^{(1)}, v_2^{(1)}, \ldots, v_{n_v}^{(1)}, \ldots, v_1^{(d)}, v_2^{(d)}, \ldots, v_{n_v}^{(d)}) \in \mathbb{R}^{d \cdot n_v},$$

where $v^{(i)}$ denotes the ith component of a vector. Denoting the number of edges of the polyline by n_e, the edge connectivity can be represented by the matrix

$$\mathbf{E} = (e_1, e_2, \ldots, e_{n_e}) \in \mathbb{N}^{2 \times n_e},$$

where $e_i \in \mathbb{N}^2$ contains the indices of the two vertices defining the ith line segment, for $i = 1, \ldots, n_e$. Since all edges are bidirectional, we represent a polyline object as an undirected graph $G = (\mathbf{v}, \mathbf{E})$.

Our goal is to geometrically transform the polyline G, in order to minimize its distance to the given point cloud

$$\mathbf{p} = (p_1^{(1)}, p_2^{(1)}, \ldots, p_{n_p}^{(1)}, \ldots, p_1^{(d)}, p_2^{(d)}, \ldots, p_{n_p}^{(d)}) \in \mathbb{R}^{d \cdot n_p}.$$

DOI: 10.1002/cnm

2.2. Transformed Polylines

For a given transformation $y : \mathbb{R}^d \times \mathbb{R}^d$ we define the deformed polyline object by

$$y(G) = G((y(v_1), y(v_2), \dots, y(v_{n_v})), \mathbf{E}). \tag{1}$$

In other words, the polyline is transformed by shifting its vertices and keeping the edge connectivity fixed. Therefore, we collocate y on the vertices, and denote the discrete transformation by

$$\mathbf{y} = (y(v_1)^{(1)}, y(v_2)^{(1)}, \dots, y(v_{n_v})^{(1)}, \dots, y(v_1)^{(d)}, y(v_2)^{(d)}, \dots, y(v_{n_v})^{(d)}) \in \mathbb{R}^{d \cdot n_v}.$$

In the notion of [1] this transformation model can be categorized as *non-parametric* in the sense that the dimension of the discrete transformation depends on the number of vertices.

On the other hand, *parametric* transformations are characterized by a parameter $\mathbf{w}$ whose size is independent of the number of vertices. We denote this dependency by $\mathbf{y}(\mathbf{w})$. Classical examples for parametric transformations are rigid transformations, which consists of rotation and translation, or affine transformations, which consists of shearing and scaling; see [1, Ch.4]. Parametric transformations can also model nonlinear deformations, for example, using Thin Plate Splines (TPS) [44] or a piecewise linear transformation model discretized on a regular grid surrounding the polyline. The latter, rather novel, concept will be discussed in more detail in Sec. 3.2.

2.3. Distance Measures

In order to measure the distance between the point cloud $\mathbf{p}$ and the transformed polyline $y(G)$, we need to find the closest point on $y(G)$, denoted by $\widehat{p_i}$, for each point p_i where $i = 1, 2, \dots, n_p$. Note that the corresponding point does not necessarily need to be a vertex but in general is located on an edge of G. The projection of the ith point, p_i, onto the jth edge of the transformed polyline is given by

$$q_{ij}(\mathbf{y}) = y_{\mathbf{E}_{1,j}} + t_{ij}(\mathbf{y})(y_{\mathbf{E}_{2,j}} - y_{\mathbf{E}_{1,j}}),$$

where the Barycentric coordinate of the ith point with respect to the jth edge, t_{ij}, is computed by

$$t_{ij}(\mathbf{y}) = \min(1, \max(0, c_{ij}(\mathbf{y}))) \text{ with } c_{ij}(\mathbf{y}) = \frac{(y_{\mathbf{E}_{2,j}} - y_{\mathbf{E}_{1,j}})^\top (p_i - y_{\mathbf{E}_{1,j}})}{\|y_{\mathbf{E}_{2,j}} - y_{\mathbf{E}_{1,j}}\|^2}. \tag{2}$$

For each p_i, we denote the corresponding point by

$$q_i(\mathbf{y}) := q_{ik}(\mathbf{y}), \quad \text{where} \quad k \in \operatorname{argmin}_{j \in \{1,2,\dots,n_e\}} \|p_i - q_{ij}(\mathbf{y})\|. \tag{3}$$

Note that, depending on the topology of the polyline object, the optimization problem (3) does not necessarily have a unique solution. In principle, finding the corresponding point to all n_p points in the point cloud requires performing $n_p \cdot n_e$ projections and choosing one with minimal distance. In practice, significant computational savings can be realized, for example, by first finding a small number of closest vertices to p_i and then testing only edges starting or ending in these vertices.

There are several options for measuring the distance between the point cloud and the polyline. As a simple but effective option, we consider the sum-of-squared-difference (SSD) function

$$D^{\mathrm{SSD}}(\mathbf{y}, \mathbf{p}) = \frac{1}{2}\|\mathbf{q}(\mathbf{y}) - \mathbf{p}\|^2, \tag{4}$$

where $\mathbf{q}(\mathbf{y})$ is a vector containing the projections of the points in $\mathbf{p}$ computed as in (3). To improve robustness against outliers, we also consider the (smoothed) Euclidean distance

$$D^{\mathrm{Euclid}}(\mathbf{y}, \mathbf{p}) = \mathbf{e}^\top \sqrt{\sum_{k=1}^{d} (\mathbf{Q}_k(\mathbf{q}(\mathbf{y}) - \mathbf{p}))^2 + \beta}. \tag{5}$$

Here, $\mathbf{Q}_k \in \mathbb{R}^{n_p \times d \cdot n_p}$ extracts the entries associated with the kth coordinate, $\mathbf{e} \in \mathbb{R}^{n_p}$ is a vector of all ones, squaring and square root are applied component-wise, and $\beta > 0$ is a conditioning

DOI: 10.1002/cnm

parameter that controls the smoothness of the distance function. Note that for $\beta = 0$ the standard Euclidean distance, which is non-differentiable at the origin, is obtained.

Both distance measures are, in general, non-convex with respect to the transformed polyline and thus local minima can be observed. For example, the distance measure may have several local minima when rotating the polyline around its main axis; see Figure 7.

2.4. Numerical Optimization

Given a polyline G and a point cloud $\mathbf{p}$ we estimate a transformation y establishing correspondence by solving the optimization problem

$$\min_{\mathbf{w}} D(\mathbf{y}(\mathbf{w}), \mathbf{p}) + \alpha S(\mathbf{w}), \tag{6}$$

where D is a distance measure (e.g., D^{SSD} or D^{Euclid}), S is a regularizer (or smoother), and $\alpha > 0$ is a parameter balancing between minimizing the distance and the smoothness of the transformation. Choosing the regularization parameter for nonlinear inverse problem such as the one at hand is a challenging topic. In the absence of a ground truth solution, we use the L-curve method described in [45] as an appropriate heuristic. For generality, we describe the parametric transformation model here. The non-parametric case can be obtained by setting $\mathbf{y}(\mathbf{w}) = \mathbf{w}$.

We use the implementation of the Gauss-Newton method and the default parameters for stopping provided in FAIR; see also [1]. To this end, we compute the gradient and approximated Hessian of the objective function in (6). Derivatives of the regularization will be discussed in Section 3.1. For the distance measures introduced above we apply chain rule to obtain

$$d_{\mathbf{w}} D(\mathbf{y}(\mathbf{w}), \mathbf{p}) = d_{\mathbf{q}} D(\mathbf{q}, \mathbf{p})\, d_{\mathbf{y}} \mathbf{q}(\mathbf{y}(\mathbf{w}))\, d_{\mathbf{w}} \mathbf{y}(\mathbf{w}),$$

where the second term, i.e., derivative of the corresponding point, is obtained in Section 2.5 and the derivative of the third term depends on the transformation model and is, for the most common choices, computed in [1]. The derivative of the first term for D^{SSD} in (4) is

$$d_{\mathbf{q}} D^{\mathrm{SSD}}(\mathbf{q}(\mathbf{y}(\mathbf{w})), \mathbf{p}) = \mathbf{q}(\mathbf{y}(\mathbf{w})) - \mathbf{p}.$$

The Hessian of the objective function is computed only approximately to ensure positive semi-definiteness and for SSD reads

$$d_{\mathbf{ww}} D(\mathbf{y}(\mathbf{w}), \mathbf{p}) \approx H^{\mathrm{SSD}}(\mathbf{y}(\mathbf{w}), \mathbf{p}) = (d_{\mathbf{w}} \mathbf{y}(\mathbf{w}))^{\top} (d_{\mathbf{y}} \mathbf{q}(\mathbf{y}(\mathbf{w})))^{\top} d_{\mathbf{y}} \mathbf{q}(\mathbf{y}(\mathbf{w}))\, d_{\mathbf{w}} \mathbf{y}(\mathbf{w}).$$

Derivatives for D^{Euclid} are slightly more complicated but can be computed similarly; see, for example, using [46, p.84f].

In each iteration of Gauss-Newton a linear system involving the gradient and the approximated Hessian needs to be solved to find a search direction. The size of the system depends on the number of parameters, i.e., number of transformation parameters or the number of vertices for the parametric or non-parametric transformation model, respectively. In the experiments considered in this work the size of the system is relatively small and we use Cholesky factorization. For large-scale problems, iterative methods such as Preconditioned Conjugate Gradient (PCG) schemes can be used; see [47]. Having obtained a search direction, a backtracked Armijo linesearch is performed [48].

2.5. Derivatives of Correspondence

An advantage of exploiting the polyline structure over common point-to-point distances is that the projection onto line segments is differentiable unless $t_{ij} \in \{0, 1\}$. Thus, $q_i(\mathbf{y})$ can (almost everywhere) be differentiated with respect to the transformed vertices. For ease of presentation, we consider the non-parametric transformation model in which each vertex is transformed directly. Derivatives for the parametric case can be computed easily using the chain rule.

For ease of presentation, we introduce the matrices $\mathbf{P}_j, \mathbf{P}_j^1, \mathbf{P}_j^2 \in \mathbb{R}^{d \times d \cdot n_v}$ that extract the jth edge and its associated vertices

$$\mathbf{P}_j \mathbf{y} = (\mathbf{P}_j^2 - \mathbf{P}_j^1)\mathbf{y}, \quad \text{where} \quad \mathbf{P}_j^k \mathbf{y} = y_{\mathbf{E}_{k,j}}, \text{ for } k = 1, 2. \tag{7}$$

DOI: 10.1002/cnm

Using the projection matrices, the projection of the ith point onto the jth edge can be written as

$$\widehat{p}_j(\mathbf{y}) = \mathbf{P}_j^1\mathbf{y} + t_{ij}(\mathbf{v})\mathbf{P}_j\mathbf{y}.$$

The derivative of the projection with respect to the positions of the vertices can be computed by applying product rule

$$\nabla_{\mathbf{y}}\widehat{p}_i(\mathbf{y}) = \mathbf{P}_j^1 + t_{ij}(\mathbf{y})\,\mathbf{P}_j + \mathbf{P}_j\mathbf{y}\,d_{\mathbf{y}}t_{ij}(\mathbf{y})^\top.$$

If $t_{ij}(\mathbf{y}) \in \{0,1\}$ the Barycentric coordinate is non-differentiable due to the min and max operations in (2) and we set $d_{\mathbf{y}}t_{ij}(\mathbf{y}) = 0$. Otherwise we apply the quotient rule, and obtain

$$d_{\mathbf{y}}t_{ij}(\mathbf{y}) = \frac{\|\mathbf{P}_j\mathbf{y}\|^2(\mathbf{P}_j^\top p_j - \mathbf{P}_j\mathbf{P}_j^1\mathbf{y} - \mathbf{P}_j^{1\top}\mathbf{P}_j\mathbf{y}) - 2\mathbf{y}^\top\mathbf{P}_j^\top(p_i - \mathbf{P}_j^1\mathbf{y})\mathbf{P}_j^\top\mathbf{P}_j\mathbf{y}}{\|(\mathbf{P}_j^2 - \mathbf{P}_j^1)\mathbf{y}\|^4}.$$

2.6. Outlier Rejection

The accuracy of the registration depends on correct identification of the *corresponding* point. In (3) the *closest* point is determined. However, these two points might not correspond in the presence of severe nonlinear motion, measurement noise, or outliers and wrong assignments may globally reduce registration accuracy. Therefore, we classify a pair of points as outlier if the distance between the points is significantly greater than the distance between the other pairs and exclude this pair when computing the distance measure. We follow [49] and eliminate the term associated with p_i and q_i if

$$D(q_i, p_i) > c\sigma, \tag{8}$$

where $\sigma > 0$ denotes the standard deviation of the residuals, and $c > 0$ a coefficient that is chosen empirically.

3. REGULARIZING NONLINEAR DEFORMATIONS

As noted earlier, minimizing the distance term in (6) alone generally is an underdetermined and, thus, ill-posed problem. To illustrate this, assume the non-parametric transformation model and consider a vertex such that no point in the point cloud is projected onto one of its adjacent edges. Then, a small displacement of this vertex into any direction would not affect the value of the distance measure. Thus, there are infinitely many solutions. This problem is common in registration and typically addressed using regularization. In this section, we present two approaches to regularization of nonlinear deformations of polylines. First, we propose a smoothness and length regularizer for non-parametric registration of polylines. Second, we describe a piece-wise linear transformation model that allows using established regularizers from image registration [1], for example, recently proposed techniques based on nonlinear elasticity [43].

3.1. Regularizing Nonparametric Transformations

Nonparametric transformations discretize the transformation y on the vertices of the polyline. Thus, the displacement applied to two different vertices is independent of one another; see (1). This yields a $n_v \cdot d$ dimensional optimization problem in (6). It is possible to construct a case in which the distance is invariant to the position of a certain vertex. To overcome the resulting ill-posednesss, we introduce two different forms of regularization.

First we define a finite difference matrix $\mathbf{D} \in \mathbb{R}^{d\cdot n_e \times d\cdot n_v}$ for vertex functions such that

$$\begin{aligned}\mathbf{D}\mathbf{y} = \Big(&y^{(1)}_{\mathbf{E}_{1,1}} - y^{(1)}_{\mathbf{E}_{2,1}}, y^{(1)}_{\mathbf{E}_{1,2}} - y^{(1)}_{\mathbf{E}_{2,2}}, \ldots, y^{(1)}_{\mathbf{E}_{1,n_e}} - y^{(1)}_{\mathbf{E}_{2,n_e}}, \ldots,\\ &y^{(d)}_{\mathbf{E}_{1,1}} - y^{(d)}_{\mathbf{E}_{2,1}}, y^{(d)}_{\mathbf{E}_{1,2}} - y^{(d)}_{\mathbf{E}_{2,2}}, \ldots, y^{(d)}_{\mathbf{E}_{1,n_e}} - y^{(d)}_{\mathbf{E}_{2,n_e}}\Big).\end{aligned}$$

DOI: 10.1002/cnm

To enforce smoothness, we consider *diffusion* regularizer

$$S^{\text{diff}}(\mathbf{y}) = \frac{1}{2}\|\mathbf{D}(\mathbf{y}-\mathbf{v})\|^2.$$

The gradient and Hessian of the regularizer are

$$d_{\mathbf{y}}S^{\text{diff}}(\mathbf{y}) = \mathbf{D}^\top\mathbf{D}(\mathbf{y}-\mathbf{v}) \quad \text{and} \quad d_{\mathbf{yy}}S^{\text{diff}} = \mathbf{D}^\top\mathbf{D}.$$

The name of the regularizer is also motivated by the interpretation of its Hessian as the (vector valued) graph Laplacian of G [50].

To allow for large nonlinear transformations, we consider the non-quadratic length regularizer

$$S^{\text{length}}(\mathbf{y}) = \frac{1}{2}\|\mathbf{r}^{\text{length}}(\mathbf{y})\|^2, \quad \text{where} \quad \mathbf{r}^{\text{length}}(\mathbf{y}) = \mathbf{I}\frac{\mathbf{D}\mathbf{y}^2}{\mathbf{D}\mathbf{v}^2} - 1.$$

Here the square root and division are computed component-wise, and $\mathbf{I} \in \mathbb{R}^{n_e \times d \cdot n_e}$ extract entries belonging to the same point and sum them up, such that

$$\mathbf{I}\mathbf{D}\mathbf{y} = \big((y^{(1)}_{\mathbf{E}_{1,1}} - y^{(1)}_{\mathbf{E}_{2,1}}) + \cdots + (y^{(d)}_{\mathbf{E}_{1,1}} - y^{(d)}_{\mathbf{E}_{2,1}}), \ldots, (y^{(1)}_{\mathbf{E}_{1,n_e}} - y^{(1)}_{\mathbf{E}_{2,n_e}}) + \cdots + (y^{(d)}_{\mathbf{E}_{1,n_e}} - y^{(d)}_{\mathbf{E}_{2,n_e}})\big).$$

For this regularizer the gradient and approximated Hessian are

$$d_{\mathbf{y}}S^{\text{length}}(\mathbf{y}) = \mathbf{r}^{\text{length}}(\mathbf{y})^\top d_{\mathbf{y}}\mathbf{r}^{\text{length}}(\mathbf{y}) \quad \text{and} \quad d_{\mathbf{yy}}S^{\text{length}} \approx d_{\mathbf{y}}\mathbf{r}^{\text{length}}(\mathbf{y})^\top d_{\mathbf{y}}\mathbf{r}^{\text{length}}(\mathbf{y}),$$

where the gradient of the residual is given by

$$d_{\mathbf{y}}\mathbf{r}^{\text{length}}(\mathbf{y}) = 2\frac{\mathbf{I}}{\mathbf{D}\mathbf{v}^2}\operatorname{diag}(\mathbf{D}\mathbf{v})\mathbf{D},$$

where $\operatorname{diag}(\mathbf{D}\mathbf{v})$ is a diagonal matrix with entries in $\mathbf{D}\mathbf{v}$ on its main diagonal. This regularizer penalizes changes in edge length due to the transformation and is, thus, invariant against rigid transformations. Thus, large translations and rotations are attainable.

Combining both regularizers above, we obtain a regularizer for non-parametric deformations of the polyline object

$$S^{\text{np}}(\mathbf{y}) = \alpha\big(\lambda S^{\text{diff}}(\mathbf{y}) + \mu S^{\text{length}}(\mathbf{y})\big). \tag{9}$$

The proposed regularizers both enforce smoothness of the transformation and are key to drive the registration in regions with sparse or noisy data. However, it is important to note that they do not ensure invertibility of the transformation. In practice, edges of the polyline might cross each other after applying the transformation and judicious choice of the regularization parameter is required.

3.2. Hyperelastic Polyline-to-Point-Cloud Registration

We now describe our novel approach that allows large deformations with guaranteed invertibility. It uses a piecewise linear transformation model and the numerical implementation of a hyperelastic regularization energy suggested in [43, 34].

The key idea is to surround the polyline object by a rectangular grid of the domain $\Omega \subset \mathbb{R}^d$ on which the transformation parameters, denoted by $\mathbf{w}$, are collocated as described in [43, 34]. In short, each grid cell is divided into four triangles (for $d = 2$) or 24 tetrahedra ($d = 3$) and the transformation is discretized using piecewise linear and globally continuous finite elements. On this class of functions, we can compute the exact value of the hyperelastic regularizer

$$S^{\text{hyper}}(y) = \int_\Omega \frac{\alpha_1}{2}\|y(x) - x\|^2 + \psi(\operatorname{cof}\nabla y) + \varphi(\det \nabla y)dx,$$

where the penalty functions ψ and φ are convex and chosen as, for example, in [43]. The volume penalty, φ, ensures invertibility of the optimal transformation as it satisfies $\varphi(1) = 0$, $\varphi(z) = \infty$ for

DOI: 10.1002/cnm

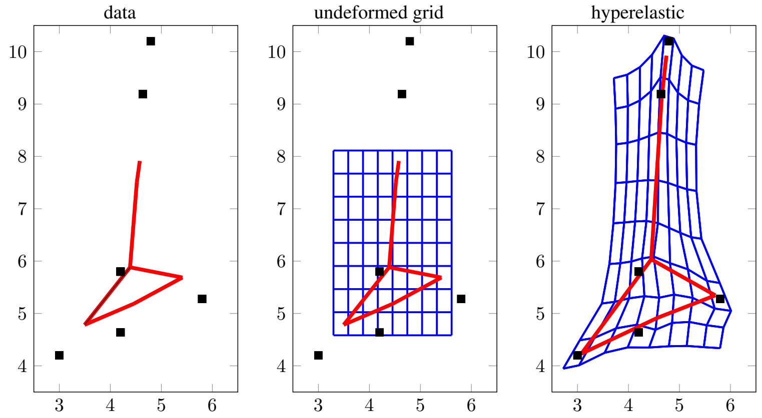


Figure 1. Hyperelastic registration of a 2D polyline (red) to a point cloud (black squares). The polyline is surrounded by a regular mesh (blue) that is subsequently deformed to minimize the distance between the polyline and the point cloud. The triangular finite element mesh is not displayed for the sake of readiness of the picture.

$z \leq 0$, and $\varphi(z) \to \infty$ for $z \to 0^+$ and $z \to \infty$. This choice ensures that the transformation keeps invertible and orientation preserving; see [43, 34] for details.

Transforming the polyline requires interpolation from the nodes of the finite element (triangular) mesh to the vertices of the polyline, written compactly as

$$\mathbf{y}(\mathbf{w}) = \mathbf{A}(\mathbf{v})\mathbf{w},$$

where $\mathbf{A}$ is an (vector field) interpolation matrix, that depends only on the position of the vertices before transformation and can be built once. Overall, the optimization problem becomes

$$\min_{\mathbf{w}} D(G(\mathbf{A}(\mathbf{v})\mathbf{w}), \mathbf{p}) + \alpha S^{\text{hyper}}(\mathbf{w}). \tag{10}$$

An example for hyperelastic polyline registration is given in Figure 1.

4. PROCESSING PIPELINE AND IMPLEMENTATION

In this section we provide implementation details for the polyline-to-point-cloud registration and provide some details about its implementation.

4.1. Registration Pipeline

To register a polyline to a given point cloud, we follow a registration pipeline consisting of the following four steps.

1. *Rigid registration*: To account for translations and rotations between both data sets we perform a registration using the rigid transformation model described in [1]. To reduce the risk of being trapped in a local minimum, after an initial registration, we estimate the main axis of the polyline object and rotate it accordingly. From each starting point constructed in this fashion, we perform up to 10 additional Gauss-Newton iterations. Finally, the solution with the smallest distance is selected.
2. *Affine registration*: To account for scaling and shearing, additional degrees of freedom are added to the transformation model by considering affine linear transformations. The optimization is initialized using the coefficients of the previous step. Finally, the coordinates

of the polyline object are updated by applying the affine transformation. In our experience, performing a rigid registration before affinely registering the data is beneficial to obtain reliable correspondences.

3. *Hyperelastic registration*: The affinely registered polyline is surrounded by a regular rectangular grid on which a hyperelastic transformation is discretized using piecewise linear, globally continuous finite elements. The polyline is transformed by evaluating the transformation on its edges. We typically use a relative coarse mesh to add additional regularization. Based on our experience this step can be skipped in the absence of large nonlinearities in the transformation, e.g., compression and expansion.
4. *Non-parametric polyline registration*: To finalize small-scale local transformations that have not been addressed in the previous steps. The strength of deformation in this step is controlled by the parameters in the regularizer, addressed in (9).

4.2. *Implementation*

We perform the experiments using MATLAB R2016a, with a 2.7GHz dual-core Intel Core i5. Our framework is implemented as an extension of the image registration framework FAIR [1]. FAIR is primarily designed for registration of images, however, it provides valuable tools that can be used for other types of data, e.g., transformation models, optimization routines, and hyperelastic regularization. Our approach is implemented as an add-on to FAIR that uses these existing methods where possible. Since FAIR has no built in support for polylines and point clouds, we implement specifically the distance functions described in Section 2.3, regularization methods described in Section 3.1, and their derivatives.

5. NUMERICAL EXPERIMENTS

We outline the potential of our method in several numerical experiments. In Section 5.1 we provide an illustrative example to show the differenced of polyline registration to more general point cloud registration approaches. In Section 5.2 we demonstrate the robustness of the proposed registration pipeline with respect to noise and outliers using a 2D synthetic phantom that resembles the stent structure. In Section 5.3 we illustrate the non-convexity of the problem and the effectiveness of the L-curve method for selecting a regularization parameter using a realistic 3D software phantom. In the subsequent Section, we present results for bioabsorbable stent reconstruction for two patients in Section 6.

5.1. *Illustrating the Difference to Point-to-Point Registration*

We illustrate the difference of Polyline-to-Point-Cloud and Point-to-Point registration using a synthetic 2D data set. The data set, illustrated in Figure 2, consists of a parallelogram and an affinely transformed version and is designed to clearly show differences polyline and more general point cloud registration approaches. Here, we use the Coherent Point Drift (CPD) algorithm [29], which addresses registration of two point sets as a probability density estimation problem.

A significant difference between both approaches is the representation of the template object. Our approach describes the template parallelogram as a polyline with four vertices and edges (i.e., $n_v = n_e = 4$) and the reference object using randomly chosen points on the edges. Here, we randomly choose 10 points per edge yielding $n_p = 40$ points. In the CPD approach, the template object needs to be represented using a point cloud. We test two different representations. First, we represent the parallelogram by its four corners. Second, we add the midpoints of the edges. The choice of the number of points used to represent the template data stems from a trade-off between accuracy and efficiency.

DOI: 10.1002/cnm

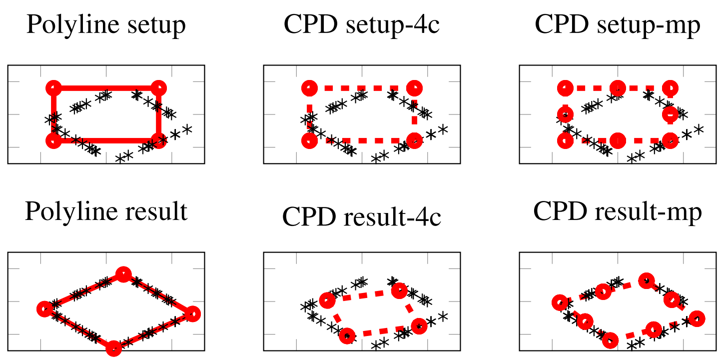


Figure 2. Comparison of Polyline and Point Cloud Registration approaches using a synthetic 2D data set consisting of a parallelogram (black) and its affinely transformed version (red). Left column: polyline setup (top) and registration results (bottom) where template (red) is represented using the four edges of the rectangle. Center column: CPD setup (top) and registration results (bottom) using only four corners to represent template object. Right column: CPD setup (top) and registration results (bottom) after adding midpoints. In the CPD results we use dashed lines to help visualize the polyline structure of the object.

We compare the proposed polyline registration and CPD using an affine transformation model; see Figure 2 for results. As to be expected the accuracy of the CPD algorithm depends crucially on the number of points used to represent the template object. We observe a larger mismatch between the transformed template and the reference parallelogram when representing the template object using its four corners and a smaller misfit when adding the midpoints. Preferable results are obtained by the proposed method as skeletal structure of the data is represented more efficiently.

Clearly the differences between both approaches will be reduced when adding more points to the representation of the template object in CPD or other point cloud registration approaches, however, this goes hand in hand with increased computational costs.

5.2. 2D Stent-resembling Polyline

We validate our approach for a 2D dataset consisting of a polyline and a point cloud generated after applying a known nonlinear deformation. This example is designed to mimic the stent reconstruction problem in 3D addressed in Section 6. To generate the data set, we first define a polyline with vertices and edges (here, $n_v = 46, n_e = 50$) and collocate a nonlinear parametric transformation on a rectangular grid surrounding the polyline. We use cubic B-splines to parameterize the transformation and adjust the weights such that the deformation introduces translation, compression and distortion, which are expected for the real data showing a stent implanted into patient's coronary artery; see Figure 3. We then interpolate the transformation on the vertices of the polyline to get a deformed polyline and sample points along its edges (here $n_p = 200$). To account for inaccuracies of the point classification, we add independent and identically distributed Gaussian white noise.

Results of the four-step pipeline described in Section 4.1 are visualized in Figure 3. For this data set, the hyperelastic registration step is performed using a rectangular grid consisting of 10×4 cells that surrounds the affinely pre-registered stent. The number of cells is chosen rather small to achieve additional regularization. We choose empirically $\alpha = 0.8$ in (10), and in (8) let $c \to \infty$ for rigid, affine, and hyperelastic registrations, to ensure overall alignment of the polyline with the data. Finally, we perform a non-parametric registration with 92 $(2 \times n_v)$ degrees of freedom. We reject outliers with $c = 2$, and to regularize the problem, we let $\alpha = 0.5, \lambda = 0.8, \mu = 2$ in (9). As shown in the bottom row of Figure 3 the Gauss-Newton method converges within a few number of iterations. Computation time is respectively 0.43, 0.50, 0.54, and 0.32 seconds for rigid, affine, hyperelastic, and non-parametric registrations.

We also compare the robustness of the distance measures, D^{SSD} and D^{Euclid}, and different sets of regularization parameters to outliers. Fixing the previous dataset (where $n_p = 200$), we add a set of outliers drawn from a uniform distribution with x values taken from $[-2.5, 2.5]$, and y values from

DOI: 10.1002/cnm

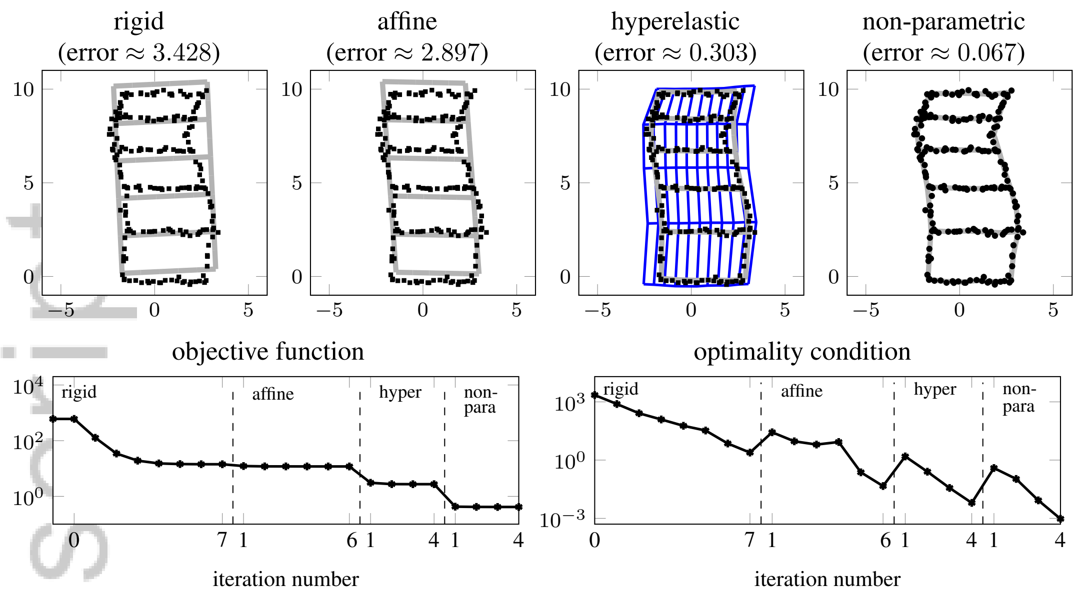


Figure 3. Results of four-step pipeline for a synthetic 2D example with ground truth. Here the dataset is obtained from nonlinear deformation of the polyline with white noise, where $n_p = 200$. Top row shows registration results at all intermediate steps. The SSD function is used as distance measure to drive the registration and the error of the computed solution with respect to the ground truth is reported. Bottom row shows the convergence of the method in terms of the objective function value (left) and norm of the gradient (right), respectively.

$[-1, 11]$. The number of outliers is increased from 0 and 100 in increments of 10 and registration is performed for each experiment in both distance terms. For the smoothed Euclidean distance we use $\beta = 10^{-2}$. We experimentally tune the regularization parameter α in (9), and to simplify the search, we fix the ratio between the diffusion and the length term as $\lambda = 0.8, \mu = 2$ and perform the registration for α taken from 21 logarithmically-spaced points between 10^{-2} and 10^{1}. After each registration we compute the error between the estimated transformation and the ground truth, which is known in this case. We repeat each experiment a hundred times and observe the relationship between optimal α values for different numbers of outliers, as well as their corresponding error to the ground truth. As expected and shown in Figure 4, the smoothed Euclidean distance is more robust against outliers and yields better reconstructions as the number of outliers increases.

5.3. 3D Phantom Registration

In this example, we illustrate the non-convexity of the optimization problem and the effectiveness of our strategy for choosing the regularization parameter using a synthetic 3D stent phantom. The data set is derived from a realistic stent geometry and contains the different components of the stents (see the next Section). Here we consider a polyline with five rings, and to simulate the patient-specific data, we apply a 3D linear spline transformation that introduces compression and distortion of the stent. A point cloud is obtained by sampling the deformed model and adding Gaussian white noise.

The initial and deformed polyline and the generated test data are shown in Figure 5. Here the dataset consists of $n_p = 864$ point and the polyline representing a five ring stent consists of $n_v = 276$ vertices and $n_e = 288$ edges. Results of the three-step pipeline using rigid, affine, non-parametric registration, described in Section 4.1 can be found in Figure 5. We use D^{Euclid} as a distance measure due to its robustness where the conditioning parameter, β, is empirically set as 0.5. We report errors of the transformed template with respect to the ground truth, i.e., the deformed polyline.

DOI: 10.1002/cnm

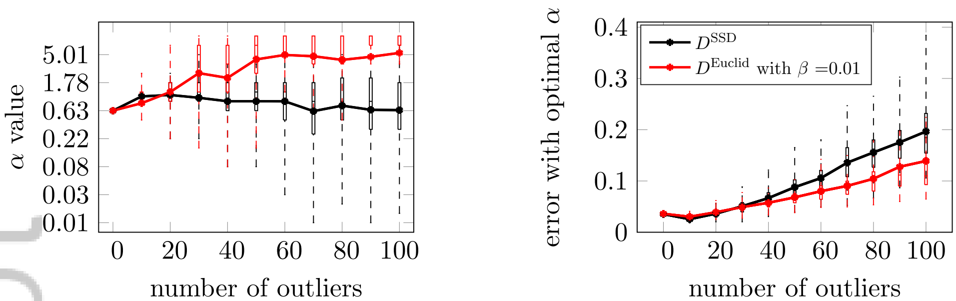


Figure 4. Optimal α value and corresponding error to the ground truth in the 2D example, when different number of outliers are tested. Results using SSD and smoothed Euclidean with $\beta = 10^{-2}$ as distance measures are shown. The experiment is run a hundred times, with α is taken from a sample of 21 logarithmically-spaced values from 10^{-2} to 10 (plotted here in log scale), and outliers from 10 equally-spaced points in the interval $[0, 100]$. The distribution of α and error are illustrated here with a box plot, data shown are within 2.7 standard deviation. To observe the trend, the mean value in each case are connected. On the left, optimal α is in general greater for smooth Euclidean; on the right, error increases as the number of outliers increases, and smooth Euclidean distance is observed with smaller errors in cases with more outliers (here greater than 30).

To illustrate the non-convexity of the objective function we rotate the template object around its main axis and show the misfit in the second row of Figure 5. To increase the chances of computing the global minimizer, we initialize the affine registration with 6 different starting guesses associated with different rotation angles. Finally, we choose the result that minimizes the objective function to initialize the non-parametric registration. To increase robustness against false correspondences, here we reject data by letting $c = 2.8$ in (8).

We use the L-Curve criterion [45] to determine an appropriate value of α in (9). The criterion does not make use of the ground truth transformation and can thus be used in real applications albeit there is no theoretical guarantee that an *optimal* value is found. We also stress that automatic choice of optimal regularization parameters, especially for nonlinear inverse problems, is an open research problem. To simplify the search for the regularization parameters needed in non-parametric registration, we keep the weights of the individual terms fixed at $\lambda = 2$ and $\mu = 1.2$. We sample 101 logarithmically-spaced points between 10^2 and 10^{-5}, perform non-parametric registration, and compute the distance between the dataset and the transformed polyline, as well as the value of the regularizer, S^{np}; results are shown in the second row of Figure 5. Finally, following the L-curve methodology we select a corner by visual inspection, which in this case yields the regularization parameter $\alpha = 0.0603$. The computational time per value of α is between 0.4 and 1 second and does not depend strongly on the actual value; see bottom rightmost plot in Figure 5.

Following the above steps, we obtain the transformed template polyline whose distance to the ground truth is 0.693, with the largest term in the sum being 0.244. This is relatively small in the scale of our simulated polyline, which spans across a 3D rectangle of size $4 \times 10 \times 2$.

6. 3D BIOABSORBABLE STENT RECONSTRUCTION

We provide a brief description of our stent reconstruction pipeline and its use in CFD analysis in Section 6.1. We then discuss the incorporation of the proposed registration methods into the pipeline and show registration results for two patients in Section 6.2.

6.1. *The Basic Geometrical Reconstruction Procedure*

Our specific interest in hemodynamics of bioabsorbable stents comes from their abnormal thickness, designed to handle pressure and stress during and after the deployment. As opposed to metallic stents, which due to their small size have minor impact on the blood flow, bioabsorbable stents

DOI: 10.1002/cnm

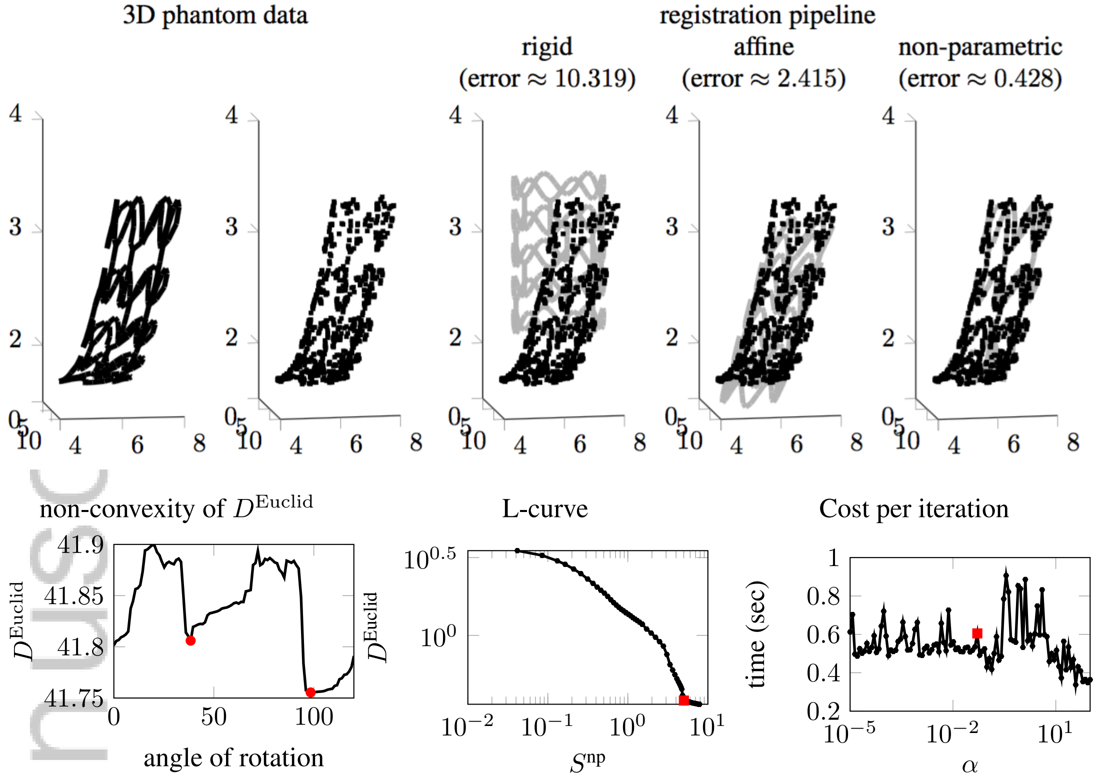


Figure 5. Registration results for 3D software phantom. Top row (left to right): 3D deformed polyline with five rings (black), sampled data with white noise (black dots), three-step reconstructed results (grey) imposed on data. The errors are computed using D^{SSD}, with respect to the ground truth. Bottom row: Non-convexity of D^{Euclid} demonstrated by rotating the 3D data around its main axis. Computing the distance for each rotation results in (at least) two local minima (visualized as red dots) (left). L-curve plot using 101 logarithmically spaced values of α between 10^2 and 10^{-5} and showing the selected value $\alpha = 0.0603$ marked by a red square (center). Runtime per iteration of non-parametric registration for different values of α. The optimal value $\alpha = 0.0603$ by the L-curve is marked by a red square.

may negatively affect outcomes since the local hemodynamics may be adversely affected leading to remodeling and reocclusion. CFD may provide a quantitative assessment of these speculations.

To enable extensive CFD simulations over a large number of patients in the framework of Computer Aided Clinical Trials (CACT) we have set up a procedure to (semi-)automatically reconstruct the vascular geometry of a specific coronary artery after deployment of a bioabsorbable stent. The current procedure (without registration of the undeployed prosthesis presented in this paper) was validated with a virtual phantom and illustrated extensively in [15]. For the sake of completeness and better understanding of the impact of the present contribution, we recall its basic steps.

1. *Strut detection in OCT images (see Fig. 6)*: The large struts are prone to an easy detection for rectangular regions in the OCT images. Most of the struts are identified automatically, even if a manual check is required for possible corrections. Note that the shadow of the catheter in the OCT images prohibits identifying the position of all the struts.
2. *Point cloud computation*: the 3D coordinates of the center of mass of each strut are stored as a point cloud.
3. *Categorization*: The stent in its undeployed configuration features two elementary components, the *rings*, i.e. the circular structures with a wavy profile; the *vertical connectors* or *beam*, connecting pairs of rings. To proceed with the geometrical reconstruction, we

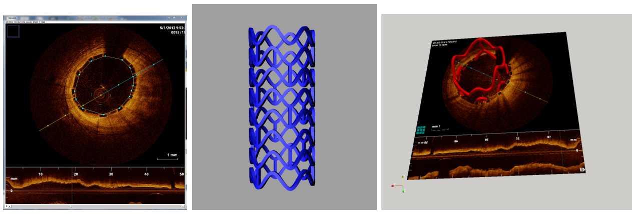


Figure 6. Left: Example of an OCT image of a slice of a coronary artery with a bioabsorbable stent. White dots denote the struts of the stent. Middle: Geometry of the undeployed stent. Right: Overlapping of a 3D volumetric reconstruction on the real OCT image.

classify the points of the cloud as part of the different components. This step is currently done manually, through a graphical user interface specifically developed for this purpose.

4. *Piecewise skeletonization*: after the categorization, the points of each component (ring or beam) are used for spline interpolation yielding a wireframe representation of the patient-specific stent.
5. *Volumetric reconstruction*: from the skeleton we obtain a 3D volume by expanding the rectangular shape around the interpolating lines according to an intrinsic frame of reference. Special procedures are undertaken for managing the intersection between different components - see Figure 6.
6. *Bending*: as OCT images are obtained from inside the vessel, accurate information about the centerline in space is missing. Thus, registration of OCT with images from other modalities able of detecting the real 3D profile of the coronary, like bi-planar angiography, is required. This is done by the commercial package `Meddis` (see www.medis.nl). Once the real 3D centerline of the coronary is obtained, a map from the rectilinear to the curved centerline is applied to the stent so to have the real 3D volumetric reconstruction on the bent artery.
7. *Stented lumen reconstruction*: after a standard reconstruction of the lumen (with no struts) with the `Vascular Modeling ToolKit` (see www.vmtk.org), we extract the stented lumen volume by boolean subtraction of the stent to the lumen. This step is performed by commercial packages like `Rhinoceros` (www.rhino3d.com) and `NetFabb` (www.netfabb.com). The stented lumen - generally after several manual repairing operation - is ready for meshing and eventually hosting CFD simulations.

Finally, CFD simulations and postprocessing (in particular, the computation of the Wall Shear Stress) can be undertaken according to standard procedures largely validated in the literature (see e.g. [51]).

The registration procedure presented in this paper occurs precisely between the point cloud after Step 2 and the skeleton of the undeployed stent extracted from the original design - Fig. 6 - as the collection of the centers of mass of rings and beams. It is sought to replace steps 3-4, as it automatizes the categorization procedure, and it includes information unaccessible from the patient-specific data that guides the skeletonization (i.e., filling the gap induced by the catheter shadow). We examplarily demonstrate the registration in the next subsection using two patient data sets of a larger data set (a total of 16 patients have been simulated so far on post-op morphologies).

Remark: Our registration procedure is not limited to reconstructing bioabsorbable stents. For example, in the case of metallic prostheses that feature smaller struts only the first two steps of the pipeline need to be changed (most importantly strut detection). For metallic stents the accuracy of the strut detection of Step 1 may be reduced and thus our registration procedure may provide less reliable results. However, we are positive that with appropriate strut-detection techniques can be used for metallic stents with minor adjustments.

6.2. *Registration of the Bioabsorbable Stent*

To increase the robustness of the approach in the presence of highly nonlinear transformations, we divide the data sets into a small number of longitudinal sections that are processed independently

DOI: 10.1002/cnm

and finally combined. The first data set corresponds to a 17-ring stent, which we divide into four sections, and the second data sets corresponds to a 23-ring, which we divide into five sections.

The point cloud and the pre-operative stent model for the top five rings of the first data set are shown in Figure 7. The polyline object in this example consists of $n_v = 276$ vertices and $n_e = 288$ edges and the point cloud consists of $n_p = 421$ points. As in the phantom case, we follow a registration pipeline using rigid, affine, and non-parametric transformations, shown in Figure 7. For distance measure, we choose the robust D^{Euclid}, and let $\beta = 0.5$ as before. After rigid registration, we rotate along the main axis of the polyline six times, as initial guesses for affine registration. This is to issue the non-convexity of the objective function (see Figure 7), as discussed in Section 5.3. We reject data by again letting $c = 2.8$ in (8), to achieve robustness.

To determine the regularization parameters in (9), we use the L-curve Criterion for the first section of each patient data set. We fix $\lambda = 2$, $\mu = 1.2$, perform the non-parametric registration using 151 logarithmically-spaced values of α between 10^6 and 10^{-10}, and identify a corner in the L-curve by visual inspection. This gives the value $\alpha = 1.359$ used to obtain the solution for this section (see Figure 7) and the parts of the data.

The non-parametric registration requires to solve for a vector $\mathbf{v}$ of length $3 \times n_v = 828$. To accelerate the search for the closest edge we compute the closest vertex using a Delaunay triangulation (here 1555 tetrahedra) and then select the projection onto one of the 2 to 3 adjacent edges as the corresponding point. While this does not necessarily give the closest point on the polyline, the approximation speeds up computation considerably. The registration process takes 40 iterations in total, with respectively 2.04, 1.89, 1.35 seconds for rigid, affine, and non-parametric transformations.

Applying the same steps to the remaining sections of the data set reduces the smoothed Euclidean distance from approximately 3.67×10^4 to $D^{\text{Euclid}} \approx 9.68$; see also Figure 8. The overall runtime for the registration process is 24.38 seconds.

We apply the same pipeline to the second dataset consisting of 23 rings where the L-curve criterion gives the regularization parameter $\alpha = 0.955$; see right plot in Figure 8. Using this value the overall distance is reduced from $D^{\text{Euclid}} \approx 2.31 \times 10^4$ to $D^{\text{Euclid}} \approx 10.97$. The total runtime in this case is approximately 34.15 seconds.

7. DISCUSSION

One key idea of our new numerical methods for nonlinear polyline-to-point-cloud registration is to exploit the polyline structure to compute the correspondence between the line and a given point by projection onto the edges of the polyline. We show that this assignment is (almost everywhere) differentiable with respect to the deformation and derivatives are easy and efficient to compute, thus enabling fast optimization. In contrast to similar point-to-point registration such as [40] we compute derivatives analytically and thereby avoid unnecessary function evaluations and nearest-neighbor searches. These searches are a computationally challenging part in many point-cloud registration approaches and our method can be extended to benefit from accelerated search strategies, for example, using kd-trees [52]. The differentiability and the potential of efficiently describing data with skeleton structure are key advantages over commonly used point-to-point registration algorithms based on ICP.

We formulate the registration as an unconstrained optimization problem that supports the most commonly used distance measures and transformation models. We provide two methods for regularizing nonlinear deformations. Maximal flexibility is obtained using a non-parametric transformation model, which directly transforms the vertices of the polyline. In this case, we derive a regularizer ensuring smoothness and penalizing length-changes caused by the transformation. It is important to note, that in this approach invertibility of the transformation is not guaranteed. In other words, transformations that cause edges of the polyline to intersect give a finite value of the regularizer and thus might be attained. To avoid this, we also suggest a more restrictive method for nonlinear registration that is based on the nonlinear elastic regularizer described in [43]. In addition to ensuring invertibility, this approach also allows to limit the flexibility of the transformation by,

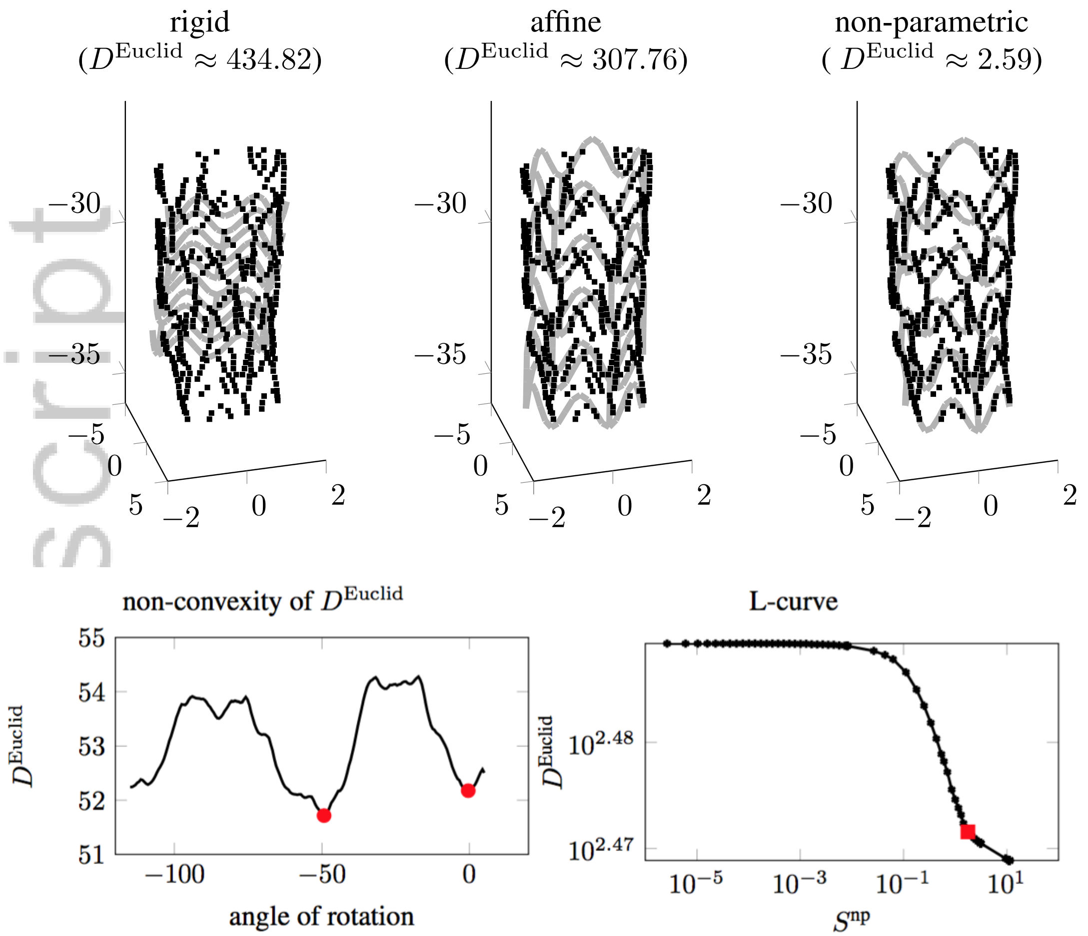


Figure 7. 3D Registration result for a section of a bioabsorbable Stent extracted from patient data. First row shows the intermediate and final results for the three-step registration pipeline. Second row shows the non-convexity of D^{Euclid} and the L-curve for 151 logarithmically spaced values of α between 10^6 and 10^{-10}. The value of $\alpha = 1.359$ is marked by a red square.

for example, discretizing the transformation on a relatively coarse mesh and appropriate choice of regularization parameters.

We use a Gauss-Newton scheme with Armijo line search to approximately solve the registration problem. Optimization-based methods have also been shown to be highly effective for point-to-point registration in [53]. In addition to the correspondence searches, another computationally costly step in the optimization is computing the search direction, which requires solving a linear system involving a positive definite approximation of the Hessian. In the experiments considered here, the size of the linear system is rather small and Cholesky factorization is used. To enable efficient non-parametric registration of polylines with a larger number of vertices, iterative linear solvers can be used.

Clearly, our formulation requires that one object is given as a polyline. While this is a limitation in some applications, our work can also be applied for registration of vascular objects; see, for example, [54, 55]. One advantage of integrating polyline registration into an image registration framework such as FAIR is the potential of combining image, polyline, and point cloud data. This will be a major item of future work. Our methods support polylines of arbitrary topology, a wide range of transformation models (such as rigid, affine and nonlinear transformations), most commonly used distance measures, as well as physically motivated regularization. Here we use

DOI: 10.1002/cnm

patient-specific reconstruction results

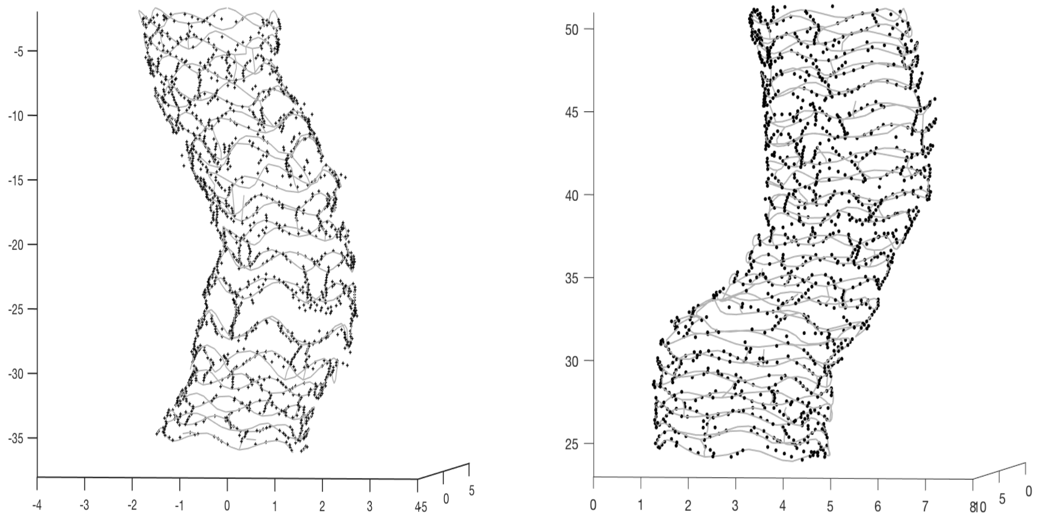


Figure 8. Final results for OCT-based reconstruction of bioabsorbable coronary artery stents for two patients with 17 rings (left) and 23 rings (right).

the FAIR package as a computational tool and integrate our methods into this general purpose registration framework; see [1].

Our methods bear similarities with methods for registration of point clouds. For example, nonrigid registration of point clouds has been explored in [44] using thin-plate spline (TPS). Probabilistic methods have also been introduced; e.g., the Coherent Point Drift (CPD) algorithm [29] addresses point matching with a Gaussian mixture model. Nonrigid registration of surfaces is discussed,e.g., in [39] as an extension of the Iterative Closest Point (ICP) algorithm, using affine transformations with locally affine regularization. The ICP algorithm and many of its variants follow a pipeline of correspondence search and transformation update using a closed-form minimization [37, 30, 36, 56, 39, 57]. Note that that the correspondence search is computationally expensive, since in a naive implementation it requires finding the nearest neighbor in the reference object for each point of the template object. While this process can be significantly accelerated, for example, using tree-based data structures [52] the complexity in general grows with the number of elements in both point clouds and, thus, representing the undeployed stent as a polyline increases the efficiency and also leads – as we will show – to an (almost everywhere) differentiable problem.

In practice, our resulting registration problems are solved within seconds requiring only a few iterations of a Gauss-Newton method, thus limiting further the number of (expensive) correspondence searches. In this sense, our method is similar to the Levenberg-Marquardt method proposed in [40], but with two major distinctions. First, we compute the derivative of the point-to-point correspondence analytically instead of approximating it using finite differences. This avoids additional correspondence searches and the choice of a step parameter for the finite difference scheme. Second, we use physically inspired regularization, which by design leads to a positive definite approximate Hessians. It has been shown in [40] that optimization-based algorithms are more flexible and competitive to special-purpose methods such as ICP that rely on closed form solutions. In addition, instead of iterating between neighbor search and computing the transformation, a general purpose optimization scheme allows us to jointly compute the correspondence estimation and the transformation.

We present four numerical experiments to demonstrate the potential of our methods.

Prepared using cnmauth.cls
DOI: 10.1002/cnm

First, we use a simple 2D example to illustrate the differences of our method to general point cloud registration approaches. In particular, a polyline object utilizes knowledge of the structure of a given point cloud, and shows significant advantages in terms of registration accuracy.

Second, we consider a 2D academic test problem with available ground truth transformation to study the robustness of our method to outliers. We find that the smoothed Euclidean distance measure outperforms the sum-of-squared-difference function in the presence of a large number of outliers.

Using a 3D software phantom resembling the structure of a cardiovascular stent we show the non-convexity of the problem and validate our L-curve based strategy for choosing the regularization parameter; see also [45].

Finally, we use our methods for reconstructing bioabsorbable stents from OCT-data for two patients. To enable patient-specific analysis, we aim at aligning a polyline representing a pre-operative cardiovascular stent to a point cloud representing locations of the stent after surgery. In our experiments we found that dividing the data sets into smaller sections considerably reduces the risk of being trapped in a local minimum when large and nonlinear transformations are present. This requires some user intervention and deriving a fully automatic process is an important item of future work.

In all experiments, good registration results are obtained in only a few number of Gauss-Newton iterations and, thus, requiring only a small number of nearest-neighbor searches.

7.1. Conclusions and Perspectives

The real-life example illustrates the potential of our method to further automize patient-specific stent reconstruction from post-surgery images. Our techniques will positively impact the current methodology described in [15] - where no registration procedure is currently performed - in two ways. First, interpolation procedures for the rings and beams of the stents have been observed to be error prone at the junctions among them. Currently, the junctions are managed by an empirical intersection of the different parts and require manual intervention. The procedure illustrated here introduces prior knowledge about the location of the intersections and is thus expected to improve the regularity and reliability of the reconstruction. Second, the procedure in [15] is intended to be used in clinical trials involving a large number of patients [58, 59, 14, 60]. Thus, automation is of great relevance. The approach in [15] requires a manual identification of the different components (rings and beams) that drives the interpolation and intersection procedures. As it is easy and convenient to label the different components on the undeployed configurations (which has to be done only once per stent model), the registration approach presented here naturally perform the labeling as a by-product of the mapping of labeled (undeployed) lines. In this way, we can eliminate the need for manual identification of rings and beams in each patient dataset. This will reduce the computational costs, as no manual procedure is required, with a significant impact on the usability of the geometrical reconstruction in clinical trials.

8. ACKNOWLEDGEMENTS

We thank Dr. Boyi Yang for his help in preparing the data set and invaluable feedback as well as the Core Lab in the Interventional Cardiology of the Emory University Hospital (H. Samady, B. Gogas) for providing the data set.

REFERENCES

[1] Modersitzki J. *FAIR: Flexible Algorithms for Image Registration*. SIAM: Philadelphia, 2009.

[2] Serruys PW, Kutryk MJ, Ong AT. Coronary-artery stents. *New England Journal of Medicine* 2006; **354**(5):483–495.

[3] Serruys PW, Chevalier B, Dudek D, Cequier A, Carrié D, Iniguez A, Dominici M, Van Der Schaaf RJ, Haude M, Wasungu L, *et al.*. A bioresorbable everolimus-eluting scaffold versus a metallic everolimus-eluting stent for ischaemic heart disease caused by de-novo native coronary artery lesions (absorb ii): an interim 1-year analysis of clinical and procedural secondary outcomes from a randomised controlled trial. *The Lancet* 2015; **385**(9962):43–54.

[4] Wentzel JJ, Krams R, Schuurbiers JC, Oomen JA, Kloet J, van der Giessen WJ, Serruys PW, Slager CJ. Relationship between neointimal thickness and shear stress after wallstent implantation in human coronary arteries. *Circulation* 2001; **103**(13):1740–1745.

[5] Gijsen FJ, Oortman RM, Wentzel JJ, Schuurbiers JC, Tanabe K, Degertekin M, Ligthart JM, Thury A, de Feyter PJ, Serruys PW, *et al.*. Usefulness of shear stress pattern in predicting neointima distribution in sirolimus-eluting stents in coronary arteries. *The American journal of cardiology* 2003; **92**(11):1325–1328.

[6] Sanmartín M, Goicolea J, García C, García J, Crespo A, Rodríguez J, Goicolea JM. Influence of shear stress on in-stent restenosis: in vivo study using 3D reconstruction and computational fluid dynamics. *Revista Española de Cardiología (English Edition)* 2006; **59**(1):20–27.

[7] Mejia J, Ruzzeh B, Mongrain R, Leask R, Bertrand OF. Evaluation of the effect of stent strut profile on shear stress distribution using statistical moments. *Biomedical engineering online* 2009; **8**(1):8.

[8] Carlier SG, van Damme LC, Blommerde CP, Wentzel JJ, van Langehove G, Verheye S, Kockx MM, Knaapen MW, Cheng C, Gijsen F, *et al.*. Augmentation of wall shear stress inhibits neointimal hyperplasia after stent implantation. *Circulation* 2003; **107**(21):2741–2746.

[9] LaDisa JF, Olson LE, Molthen RC, Hettrick DA, Pratt PF, Hardel MD, Kersten JR, Warltier DC, Pagel PS. Alterations in wall shear stress predict sites of neointimal hyperplasia after stent implantation in rabbit iliac arteries. *American Journal of Physiology-Heart and Circulatory Physiology* 2005; **288**(5):H2465–H2475.

[10] Chen HY, Hermiller J, Sinha AK, Sturek M, Zhu L, Kassab GS. Effects of stent sizing on endothelial and vessel wall stress: potential mechanisms for in-stent restenosis. *Journal of Applied Physiology* 2009; **106**(5):1686–1691.

[11] Koskinas KC, Chatzizisis YS, Antoniadis AP, Giannoglou GD. Role of endothelial shear stress in stent restenosis and thrombosis: pathophysiologic mechanisms and implications for clinical translation. *Journal of the American College of Cardiology* 2012; **59**(15):1337–1349.

[12] Zhang Y, Bourantas CV, Farooq V, Muramatsu T, Diletti R, Onuma Y, Garcia-Garcia HM, Serruys PW, *et al.*. Bioresorbable scaffolds in the treatment of coronary artery disease. *Med Devices (Auckl)* 2013; **6**:37–48.

[13] Yang B, Gogas B, Esposito G, Hung O, Arzrumly ER, Piccinelli M, King S, Giddens D, Veneziani A, Samady H. Novel in-human four dimensional wall shear stress calculation of a coronary bioresorbable scaffold using optical coherence tomography images and blood flow simulations. *Journal of the American College of Cardiology* 2015; **65**(10):A1832.

[14] Gogas BD, Yang B, Piccinelli M, Giddens DP, King SB, Kereiakes DJ, Ellis SG, Stone GW, Veneziani A, Samady H. Novel 3-dimensional vessel and scaffold reconstruction methodology for the assessment of strut-level wall shear stress after deployment of bioresorbable vascular scaffolds from the absorb iii imaging substudy. *JACC: Cardiovascular Interventions* 2016; **9**(5):501–503.

[15] Yang B, Piccinelli M, Esposito G, Han T, Y B, Gogas B, Giddens D, Samady H, Veneziani A. Patient-specific 3D volumetric reconstruction of bioresorbable stents. *Technical Report TR-2017-001*, Department of Mathematics and Computer Science, Emory University 2017.

DOI: 10.1002/cnm

[16] Morlacchi S, Migliavacca F. Modeling stented coronary arteries: where we are, where to go. *Annals of biomedical engineering* 2013; **41**(7):1428–1444.

[17] Steinman DA. Image-based computational fluid dynamics modeling in realistic arterial geometries. *Annals of biomedical engineering* 2002; **30**(4):483–497.

[18] Formaggia L, Quarteroni A, Veneziani A. *Cardiovascular Mathematics: Modeling and simulation of the circulatory system*, vol. 1. Springer Science & Business Media, 2010.

[19] Taylor CA, Steinman DA. Image-based modeling of blood flow and vessel wall dynamics: applications, methods and future directions. *Annals of biomedical engineering* 2010; **38**(3):1188–1203.

[20] Balossino R, Gervaso F, Migliavacca F, Dubini G. Effects of different stent designs on local hemodynamics in stented arteries. *Journal of Biomechanics* 2008; **41**(5):1053 – 1061, doi: https://doi.org/10.1016/j.jbiomech.2007.12.005.

[21] Dehlaghi V, Shadpoor MT, Najarian S. Analysis of wall shear stress in stented coronary artery using 3d computational fluid dynamics modeling. *Journal of Materials Processing Technology* 2008; **197**(13):174 – 181,

[22] LaDisa JF, Guler I, Olson LE, Hettrick DA, Kersten JR, Warltier DC, Pagel PS. Three-dimensional computational fluid dynamics modeling of alterations in coronary wall shear stress produced by stent implantation. *Annals of Biomedical Engineering* 2003; **31**(8):972–980,

[23] Radu MD, Pfenniger A, Raber L, de Marchi SF, Obrist D, Kelbæk H, Windecker S, Serruys PW, Vogel R. Flow disturbances in stent-related coronary evaginations: a computational fluid-dynamic simulation study. *EuroIntervention* 2014; **10**(1):113–123.

[24] Ellwein LM, Otake H, Gundert TJ, Koo BK, Shinke T, Honda Y, Shite J, LaDisa Jr JF. Optical coherence tomography for patient-specific 3D artery reconstruction and evaluation of wall shear stress in a left circumflex coronary artery. *Cardiovascular Engineering and Technology* 2011; **2**(3):212–227.

[25] Chiastra C, Wu W, Dickerhoff B, Aleiou A, Dubini G, Otake H, Migliavacca F, LaDisa JF. Computational replication of the patient-specific stenting procedure for coronary artery bifurcations: From OCT and CT imaging to structural and hemodynamics analyses. *Journal of Biomechanics* 2015; **49**(11):2102–2111.

[26] Migliavacca F, Gervaso F, Prosi M, Zunino P, Minisini S, Formaggia L, Dubini G. Expansion and drug elution model of a coronary stent. *Computer methods in biomechanics and biomedical engineering* 2007; **10**(1):63–73.

[27] Gijsen FJ, Migliavacca F, Schievano S, Socci L, Petrini L, Thury A, Wentzel JJ, van der Steen AF, Serruys PW, Dubini G. Simulation of stent deployment in a realistic human coronary artery. *Biomedical engineering online* 2008; **7**(1):23.

[28] O'Brien CC, Kolandaivelu K, Brown J, Lopes AC, Kunio M, Kolachalama VB, Edelman ER. Constraining OCT with knowledge of device design enables high accuracy hemodynamic assessment of endovascular implants. *PloS one* 2016; **11**(2):e0149 178.

[29] Myronenko A, Song X. Point set registration: Coherent point drift. *IEEE transactions on pattern analysis and machine intelligence* 2010; **32**(12):2262–2275.

[30] Besl PJ, McKay ND. A method for registration of 3-D shapes. *IEEE Trans. Pattern Anal. Mach. Intell.* Feb 1992; **14**(2):239–256.

[31] Zitová B, Flusser J. Image registration methods: a survey. *Image and Vision Computing* 2003; **21**(11):977–1000.

[32] Brown LG. A survey of image registration techniques. *ACM Computing Surveys (CSUR)* 1992; **24**(4):325–376.

[33] Maintz JBA, Viergever MA. A survey of medical image registration. *Medical Image Analysis* 1998; **2**(1):1–36.

[34] Ruthotto L, Modersitzki J. Non-linear image registration. *Handbook of Mathematical Methods in Imaging*. Springer New York: New York, NY, 2015; 2005–2051.

[35] Guéziec A, Ayache N. Smoothing and matching of 3-D space curves. *International Journal of Computer Vision* 1994; **12**(1):79–104.

[36] Zhang Z. Iterative point matching for registration of free-form curves and surfaces. *International journal of computer vision* 1994; **13**(2):119–152.

[37] Chen Y, Medioni G. Object modelling by registration of multiple range images. *Image Vision Comput.* Apr 1992; **10**(3):145–155.

[38] Audette MA, Ferrie FP, Peters TM. An algorithmic overview of surface registration techniques for medical imaging. *Medical image analysis* 2000; **4**(3):201–217.

[39] Amberg B, Romdhani S, Vetter T. Optimal step nonrigid icp algorithms for surface registration. *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, IEEE, 2007; 1–8.

[40] Fitzgibbon AW. Robust registration of 2D and 3D point sets. *Image and Vision Computing* 2003; **21**(13):1145–1153.

[41] Mitra NJ, Gelfand N, Pottmann H, Guibas L. Registration of point cloud data from a geometric optimization perspective. *Proceedings of the 2004 Eurographics/ACM SIGGRAPH symposium on Geometry processing*, ACM, 2004; 22–31.

[42] Fischer B, Modersitzki J. Ill-posed medicine—an introduction to image registration. *Inverse Problems* 2008; **24**(3):034 008.

[43] Burger M, Modersitzki J, Ruthotto L. A hyperelastic regularization energy for image registration. *SIAM Journal on Scientific Computing* 2013; **35**(1):B132–B148.

[44] Chui H, Rangarajan A. A new point matching algorithm for non-rigid registration. *Computer Vision and Image Understanding* Feb 2003; **89**(2-3):114–141.

[45] Hansen PC. *Rank-deficient and discrete ill-posed problems*. SIAM Monographs on Mathematical Modeling and Computation, SIAM: Philadelphia, PA, 1998.

[46] Haber E. *Computational Methods in Geophysical Electromagnetics*. SIAM, Philadelphia, PA, 2015.

[47] Saad Y. *Iterative Methods for Sparse Linear Systems*. Second Edition, SIAM: Philadelphia, 2003.

[48] Nocedal J, Wright S. *Numerical Optimization*. Springer Series in Operations Research and Financial Engineering, Springer Science & Business Media, 2006.

[49] Masuda T, Sakaue K, Yokoya N. Registration and integration of multiple range images for 3-D model construction. *Pattern Recognition, 1996., Proceedings of the 13th International Conference on*, vol. 1, IEEE, 1996; 879–883.

DOI: 10.1002/cnm

[50] Bollobás B. Random graphs. *Modern Graph Theory*. Springer, 1998; 215–252.

[51] Passerini T, Quaini A, Villa U, Veneziani A, Canic S. Validation of an open source framework for the simulation of blood flow in rigid and deformable vessels. *International journal for numerical methods in biomedical engineering* 2013; **29**(11):1192–1213.

[52] Friedman JH, Bentley JL, Finkel RA. An Algorithm for Finding Best Matches in Logarithmic Expected Time. *ACM Transactions on Mathematical Software (TOMS)* Sep 1977; **3**(3):209–226,

[53] Castellani U, Bartoli A. *3D Shape Registration*. Springer London, 2012; 221–264.

[54] Beuthien B. Restringierte Medizinische Bildregistrierung. PhD Thesis, Universitaet zu Luebeck 2014.

[55] Groher M, Zikic D, Navab N. Deformable 2D-3D Registration of Vascular Structures in a One View Scenario. *Medical Imaging, IEEE Transactions on* 2009; **28**(6):847–860.

[56] Jost T, Hügli H. A multi-resolution ICP with heuristic closest point search for fast and robust 3D registration of range images. *3-D Digital Imaging and Modeling, 2003. 3DIM 2003. Proceedings. Fourth International Conference on*, IEEE, 2003; 427–433.

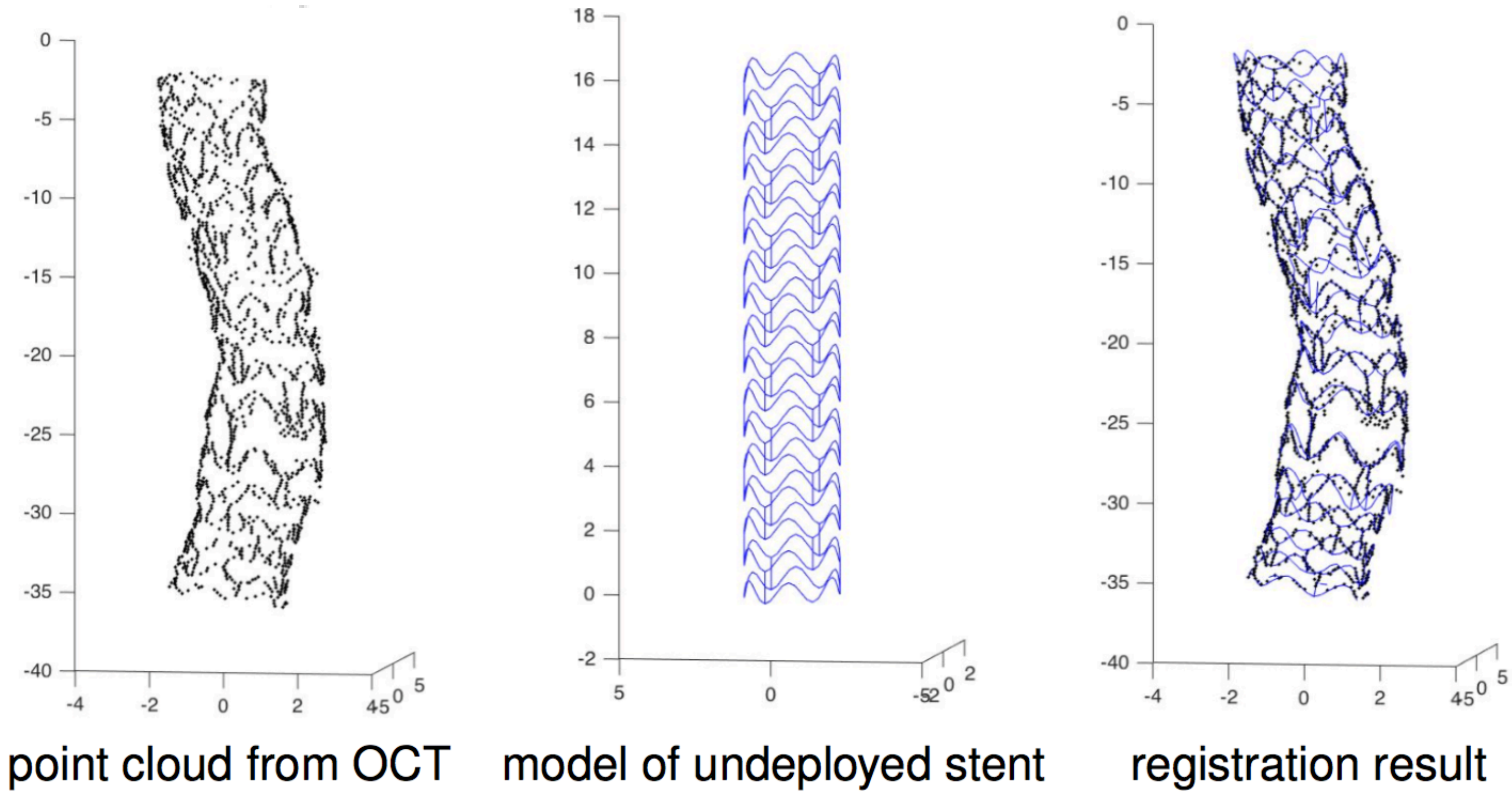
[57] Censi A. An ICP variant using a point-to-line metric. *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, IEEE, 2008; 19–25.

[58] Gogas BD, Yang B, Passerini T, Veneziani A, Piccinelli M, Esposito G, Rasoul-Arzrumly E, Awad M, Mekonnen G, Hung OY, *et al.*. Computational fluid dynamics applied to virtually deployed drug-eluting coronary bioresorbable scaffolds: Clinical translations derived from a proof-of-concept. *Global cardiology science & practice* 2014; **2014**(4):428.

[59] Gogas BD, Yang B, Passerini T, Veneziani A, Piccinelli M, Esposito G, Rasoul-Arzrumly E, Awad M, Mekonnen G, Hung OY, *et al.*. Computational fluid dynamics applied to virtually deployed drug-eluting coronary bioresorbable scaffolds: Clinical translations derived from a proof-of-concept. *Global Cardiology Science and Practice* 2015; :56.

[60] Gogas BD, Benham JJ, Hsu S, Sheehy A, Lefer DJ, Goodchild TT, Polhemus DJ, Bouchi YH, Hung OY, Yoo SY, *et al.*. Vasomotor function comparative assessment at 1 and 2 years following implantation of the absorb everolimus-eluting bioresorbable vascular scaffold and the xience v everolimus-eluting metallic stent in porcine coronary arteries. *JACC: Cardiovascular Interventions* 2016; **9**(7):728–741.

DOI: 10.1002/cnm



CNM_2934_GTOC.tiff