Autonomous Guidewire Navigation in Dynamic Environments

Valentina Scarponi, François Lecomte, Michel Duprez, Florent Nageotte, Stéphane Cotin

To cite this version:

Valentina Scarponi, François Lecomte, Michel Duprez, Florent Nageotte, Stéphane Cotin. Autonomous Guidewire Navigation in Dynamic Environments. International Conference on Intelligent Robots and Systems, Oct 2024, Abu Dhabi, United Arab Emirates. hal-04681908

HAL Id: hal-04681908

https://inria.hal.science/hal-04681908v1

Submitted on 30 Aug 2024

Autonomous Guidewire Navigation in Dynamic Environments

Valentina Scarponi*, † François Lecomte *, † Michel Duprez*
Florent Nageotte‡ Stéphane Cotin*

Abstract

Cardiovascular disease treatment involves the challenging task of navigating guidewires and catheters through the vascular anatomy. This often results in prolonged procedures where both the patient and clinician are subjected to X-ray radiation. As a potential solution, Deep Reinforcement Learning methods have demonstrated potential in learning this task, paving the way for automated catheter navigation during robotic interventions. However, current works show a limited ability to generalize to unseen and/or deforming anatomies.

In this paper, we extend our previous reinforcement learning approach in two main areas: we improve the training strategy to learn a control of the device even when the vascular anatomy is deforming and we propose a method to estimate the motion of the anatomy from single view fluoroscopy images. The combination of these two contributions makes it possible to automatically navigate across a moving vascular anatomy under fluoroscopic imaging, even without injecting a contrast agent. We validate our method on two scenarios: a simulated beating heart and a liver subjected to breathing motion. Our approach leads to an average success rate of 95% in reaching random targets within these anatomies. Our framework is also computationally efficient, enabling the training of our controller to be completed in about 6 hours.

1 Introduction

Navigating a catheter and guidewire through the vascular system in a safe and efficient manner is crucial to minimize the patient's and clinician's exposure to X-ray radiation from the fluoroscopic imaging system. This task demands a thorough understanding of the anatomy, superior device control, and a comprehensive grasp of fluoroscopic visualization. However, even seasoned clinicians may take considerable time to reach specific targets. Robotic systems can potentially enhance this process [1]. Yet, these robots are still master-follower systems that operate the devices based on the clinician's inputs. To further assist the clinician, current research is shifting towards the creation of autonomous and semi-autonomous systems. Among the semi-autonomous systems, Zhang *et al.* [2] proposed an algorithm to maintain the tip of a robotized bronchoscope at the center of the airways. This AI-based

---

[*] MIMESIS-team, Inria de l'université de Loraine, MLMS team, Icube, univesité de Strasbourg, France, `valentina.scarponi@inria.fr`, `francois.lecomte@inria.fr`, `michel.duprez@inria.fr`, `stephane.cotin@inria.fr`

[†] V. Scarponi and F. Lecomte equally contributed to the work

[‡] AVR team, Icube, univesité de Strasbourg, France, `nageotte@unistra.fr`

[§] This work of the Interdisciplinary Thematic Institute HealthTech, as part of the ITI 2021-2028 program of the University of Strasbourg, CNRS and Inserm, was supported by IdEx Unistra (ANR-10-IDEX-0002) and SFRI (STRAT'US project, ANR-20-SFRI-0012) under the framework of the French Investments for the Future Program.

algorithm uses both bronchoscopic images and human commands as inputs and predicts a corrective motion. Autonomous systems generally rely on Deep Reinforcement Learning (DRL) and use fluoroscopic images to predict a control action (rotation and translation) to be executed at the device's proximal end. Some research trains and applies the learned control entirely in simulated environments [3]. Other studies train the neural controller using images of the phantom where the navigation will later be performed [4, 5] while others perform the training in a simulated environment and then use images of the phantom during navigation [6].

The limitations of current research are two-fold. Using fluoroscopic images as input to the neural network can cause uncertainties about the orientation of the tip, leading to prediction errors. Also, the training process does not generalize well and requires individual training for each patient. As reported in [7, 8], learning controllers that can perform tasks in both familiar and unfamiliar environments remains a significant challenge in DRL.

To tackle this problem in the context of endovascular procedures, Kweon *et al.* [5] suggested a segment-wise learning method to speed up training using human demonstrations, transfer learning, and weight initialization. However, this method still necessitates network training each time the environment is altered or expanded. Similarly, in the research conducted by Karstensen *et al.* [9], the controller performance dropped from a 75% success rate in navigating known anatomies to 29% when real patient vessels were used. Chi *et al.* [10, 11] proposed different strategies to obtain an optimal control of the device. In [11], they used a generative adversarial imitation learning method aimed at learning the catheterization of different arteries, with a success rate of about 70% when the aortic type was altered. In [10], they trained a statistical model to perform the cannulation of the innominate aorta and applied the same controller to variations of the aortic arch type. This technique reported an average 98% success rate in new but very similar geometries, using human demonstrations for each new task.

Recently, we proposed a DRL method that achieves excellent generalization thanks to a specific training strategy [12]. Using a set of only 4 bifurcation shapes, and a shape-invariant observation space, the learned controller was able to navigate complex, unseen anatomies. Three main assumptions were made: the vessels have a nearly constant radius, the bifurcations always have 2 exit vessels, and the anatomy is not moving or deforming during navigation. The first assumption is not a limitation of the method but a consequence of using a unique guidewire during the navigation. With a constant tip shape, only vessels of a compatible diameter can be accessed. Branching patterns with one entry vessel and two exit vessels were chosen as bifurcation is the most common pattern [13].

This paper addresses the third assumption (static anatomy) and proposes two main contributions: a training strategy able to learn a control of the device even when the vascular anatomy is moving and/or deforming (see Section 2.1), and a method to estimate the motion of the anatomy from single view fluoroscopy images (see Section 2.2). The combination of these two contributions makes it possible to automatically navigate across a moving anatomy under fluoroscopic imaging, even without injecting a contrast agent. Our results (see Section 3) illustrate the genericity of the training, and the excellent performance of our method, even when applied to complex, deforming anatomies only observed through 2D fluoroscopic imaging.

2 Method

2.1 Guidewire control learning in dynamic environments

Our objective is to develop a generalized neural controller able to control the motion, in particular the rotation, of a guidewire through a complex, deforming vascular tree, from its insertion point until a given target is reached (see Fig. 6 right, 7 right), while it is advanced at a variable speed. This control is performed at the proximal end of the device, and accounts for both the device and anatomy deformations during navigation. Our learning method relies on five main elements: 1) an efficient DRL algorithm; 2) a fast and accurate simulated environment that can be updated based on external input; 3) an observation space robust to affine transformations of the anatomy; 4) a specific reward function; 5) an optimal choice of training anatomies. They are described below.

2.1.1 Training algorithm

Reinforcement Learning (RL) constitutes one of the areas of machine learning. In this specific branch, an agent learns to achieve specific goals by interacting through its actions with an environment. The problem is usually formulated as a Markov Decision Process [14], *i.e.* $(\mathcal{S}, \mathcal{A}, P, r, \gamma)$, in which $\gamma \in [0, 1]$ defines the discount factor, and $\mathcal{S}$ and $\mathcal{A}$ represent a set of states and actions respectively. Each action $a_t \in \mathcal{A}$ induces a transition in the system from the current state $s_t \in \mathcal{S}$ to the next state $s_{t+1} \in \mathcal{S}$, and is chosen by a policy π, mapping states to actions $\mathcal{S} \to \mathcal{A}$. The probability density of the next state $s_{t+1} \in \mathcal{S}$ given the current state $s_t \in \mathcal{S}$ and action $a_t \in \mathcal{A}$ is denoted by $P(s_{t+1}|s_t, a_t)$. For each transition, the agent receives a reward $r(a_t, s_t)$. The agent observes the environment through the observation space Ω, which constitutes a total or partial description of the environment itself.

As in our previous work ([12]), we adopted in this study the Soft Actor-Critic (SAC) algorithm, which outperformed previous algorithms [15] such as the deep deterministic policy gradient (DDPG), largely used for autonomous catheter navigation [3, 6]. In Eq. (1) the objective function of the SAC algorithm is reported, in which the entropy term $\mathcal{H}(\pi(\cdot|s_t))$ is introduced. This term, which constitutes the main novelty of SAC algorithm, promotes the exploration of the environment and discourages the repetition of actions that may exploit inconsistencies in the approximated Q-function.

$$J(\pi) = \sum_{t=0}^{T} \mathbb{E}_{(s_t, a_t) \sim \rho_\pi} [r(s_t, a_t) + \alpha \mathcal{H}(\pi(\cdot|s_t))]. \tag{1}$$

Here $\rho_\pi(s_t)$ and $\rho_\pi(s_t, a_t)$ represent respectively the state and state-action marginals of the trajectory distribution induced by a policy $\pi(a_t|s_t)$.

2.1.2 Simulation of the training environment

The virtual environment used to train the DRL algorithm is based on a physics-based simulation of the device and its interactions with the vessel walls. We developed our simulator using the open-source SOFA framework [16] and relying on Timoshenko beam theory [17] to model the physics of the guidewire. The system to be solved is reported in Equation (2) in its matrix form.

$$(\mathbf{M} - dt^2 \mathbf{K}) \Delta v = dt \cdot f(x(t)) + dt^2 \cdot \mathbf{K} v(t), \tag{2}$$

where $\mathbf{M}$ represents the mass matrix, $\mathbf{K}$ the stiffness matrix and dt the time step. v and Δv denote the velocity and the velocity variation respectively and f, which is a function of the current positions $x(t)$, represents the internal and external forces applied to the system. The interactions between the vessel wall and the guidewire are computed using a constraint-based approach and the position of the vascular anatomy is updated at each time step based on the current fluoroscopic image (see Section 2.2). Equation (2) then becomes:

$$(\mathbf{M} + dt \frac{df}{d\dot{x}} + dt^2 \frac{df}{dx}) \Delta v = -dt(f + dt \frac{df}{dx} v) + dt \mathbf{H}^T \lambda, \tag{3}$$

where $\mathbf{H}^T \lambda$ is the vector of constraint forces, with $\mathbf{H}$ containing the constraint directions arising from the collision detection, and λ the Lagrange multipliers. The physics of the guidewire model is then corrected by computing the contact force λ using a Gauss-Seidel algorithm [18].

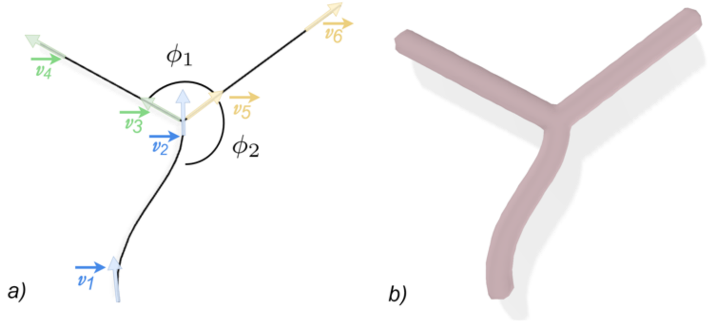

Figure 1: Procedural training shape generation process. a) Centerline $C(\phi_i, \nu_j)$ of the vessel with $i \in \{1, 2\}$ and $j \in \{1, .., 6\}$. ϕ_1 and ϕ_2 are the angles between the bifurcation branches and ν_j are the tangents to the centerline shape at each endpoint. b) 3D shape of the vessel obtained by extrusion of the centerlines C.

Using a Block Tridiagonal solver, the navigation of the virtual device is simulated at 90 frames per second, maintaining both short training times (Section 3) and a sufficient level of accuracy ($2.0 \pm 0.9\ mm$ error between the simulated guidewire and the shape of a scanned guidewire inserted inside a vascular phantom).

2.1.3 Reinforcement learning strategy

As illustrated in Scarponi *et al.* [12], the definition of the observation space, and choice of the training geometries, are essential to learn a generalizable control. In this work, we keep a similar strategy: we train the RL algorithm on a set of bifurcation patterns, unrelated to the test anatomies. This local vascular shape is represented by both a surface mesh and a centerline. The only assumption about the training shapes is that the diameter of the vessels is nearly constant and that they have a Y-shaped topology.

We then extend the work from [12] in two areas. First, we augment the training database by introducing shape variations of the training anatomy during the training process (i.e. similar to sim-to-real approaches). This shape variation is continuous throughout space and time, to avoid discontinuities in the displacement field that would cause errors in the simulation. Second, we formalize the shape generation process by making it procedural, rather than handcrafted. We characterize the 3D vessel shape from its centerline C from which it is extruded. C is defined as $C(\phi_i, \nu_j)$ with $i \in \{1, 2\}$ and $j \in \{1, .., 6\}$, where ϕ_1 and ϕ_2 define the angles between the bifurcation branches and ν_j are the tangent of the centerline shape at each endpoint (see Fig. 1). Starting from the simplest geometry (a Y-shaped bifurcation with straight branches) we progressively deform this shape into a series of other shapes, by varying smoothly ϕ_i and ν_j and maintaining a constant vessel diameter. Fig. 2 illustrates this process. Let's call $\mathcal{B}$ the set of all the bifurcation shapes we generate through our process. We split $\mathcal{B}$ into a series of N subsets $\mathcal{B}_k$ of random length, such that $\cup_{k=1}^{N} \mathcal{B}_k = \mathcal{B}$. Each subset $\mathcal{B}_k$ represents a different range of shape variations, from small deformations to large ones. These shape variations are then used as training anatomies during the learning processes of our neural controller. For each training episode, a target is randomly selected, as well as a subset $\mathcal{B}_k$ of the varying training anatomy. The initial rotation of the guidewire around its axis and its orientation relative to the centerline are chosen randomly to enhance the exploration of the environment and, during each episode, the velocity of the device is also randomly modified.

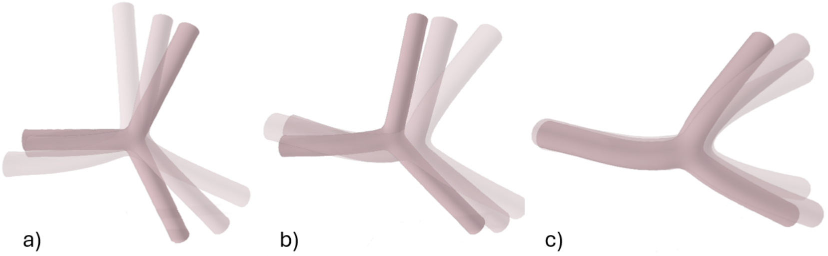

Figure 2: Examples of $\mathcal{B}_k$ subsets. Each subset $\mathcal{B}_k$ represents a range of bifurcation shape variation and $\cup_{k=1}^{N}\mathcal{B}_k = \mathcal{B}$, where $\mathcal{B}$ defines the complete set of bifurcation shapes considered in this work.

2.1.4 Nearly shape-invariant observation space

To enforce generalization of the learned control, we proposed as in [12] an observation space that is rotation and translation invariant, but also shows little sensitivity to the shape variation of the bifurcation. This is achieved by defining observations that are relative to the position of the device in the environment. In this work, we expand the observation space by adding elements that permit to navigate geometries that are different both in shape and size, with the sole caution of using a guidewire compatible with the vessel diameter. The observation space Ω is constructed as follows:

$$\Omega = \{\boldsymbol{\zeta}_t, \boldsymbol{\zeta}_{t-ndt}, \lambda_t, \lambda_{t-ndt}, a_t, \omega, d_v\}$$

- Let $\mathbf{t}_i, i \in \{1, ..., N\}$ be the tangent vector at the coordinate $\mathbf{x}_i$ along the tip of the guidewire, and $\mathbf{c}_j, j \in \{1, ..., N\}$ the tangent vector of the centerline at position $\mathbf{x}_j = \mathbf{x}_i + \mathbf{h}$. We define $\zeta_i = \mathbf{t_i} \cdot \mathbf{c_i}\ \forall\ i \in \{1, ..., N\}$. To handle dynamic environments, $\mathbf{c_i}$ must be updated. This does not require changing the observation space defined in [12] but necessitates estimating this change from live images during an intervention. Our method for handling this challenge is described in Section 2.2.
- We then define $\boldsymbol{\zeta}_m = [\zeta_1, \zeta_2, ..., \zeta_N]_m$, with $m \in \{t; t-ndt\}$.
- λ_t and λ_{t-ndt} represent the distance between the tip of the guidewire and the target at time t and $t-ndt$, normalized with respect to the initial distance to the target λ_0. λ_0 is defined as the target distance at the entrance of the bifurcation region.
- a_t is the action that determines the transition of the system from s_{t-ndt} to s_t.
- $\omega = \mathbf{k}_p \cdot \mathbf{w}_p$, where $\mathbf{k}_p$ and $\mathbf{w}_p$ are the projections of the vectors $\mathbf{k}$ and $\mathbf{w}$ onto a plane Γ perpendicular to the centerline of the branch leading to the target (see Fig. 3b). $\mathbf{k}$ represents the radial vector of curvature located in the middle of the curved tip, and $\mathbf{w}$ is the vector describing the direction of the wrong branch. To be robust to different vessel dimensions, both in terms of vessel diameter and exit branch length, the proper choice of $\mathbf{w}$ and Γ normal vector (n_Γ) is crucial. $\mathbf{w}$ norm is proportional to the vessel diameter and its starting point is fixed at the center of the bifurcation, while n_Γ magnitude is proportional to the squared diameter of the vessel and it originates from the projection of the guidewire distal end onto the centerline (see Fig. 3b).
- $d_v = \mathbf{v} \cdot \mathbf{c}$, where $\mathbf{v}$ describes the current velocity of the guidewire and $\mathbf{c}$ the tangent to the centerline near the tip of the guidewire (see Fig. 3c).

It is important to notice that all the parameters used to build the observation space can be computed in both the virtual (training) environment and in a real setup. The vessel geometry can be retrieved from preoperative images and updated intra-operatively (see Section 2.2) and the tip shape of the guidewire can be reconstructed from Fiber Bragg Gratings (FBG) data [19] using an optical fiber embedded in the catheter or guidewire.

2.1.5 Reward function

Another key element to learning the optimal action is the engineering of the reward function. We design our reward function as the weighted sum of three terms:

$$r(s_t, a_t) = \underbrace{\frac{2}{1+e^{5(\omega-0.1)}} - 1}_{a} + \underbrace{0.5\,(1-\lambda_t)}_{b} + \underbrace{(-0.2|a_t|)}_{c},$$

where part a of the reward function encourages the agent to obtain a tip direction $\mathbf{k}_p$ opposite to $\mathbf{w}_p$ (see Fig. 3). This function is a modified version of the sigmoid activation function. The output of part a is a decreasing function taking its values in $[-1; 1] \in \mathbb{R}$. Part b of the reward increases as the target is approached, while part c discourages the agent from rotating the instrument when it is unnecessary.

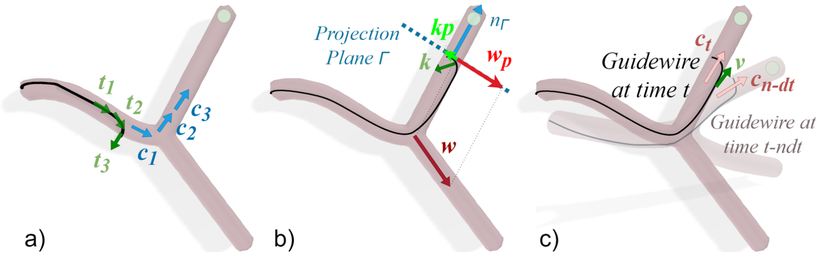

Figure 3: Observation space: 1) dot product between $\mathbf{t}_i$ (tangents to the tip of the device) and $\mathbf{c}_j$, describing the downstream centerline, with $i \in \{1, ..., 3\}$ and $j \in \{1, .., 3\}$ (a), 2) normalized distance between the tip of the guidewire and the target, 3) chosen action, 4) dot product between $\mathbf{k}_p$, describing tip's direction, and $\mathbf{w}_p$, describing the direction of the branch that does not lead to the target (b) 5) dot product between $\mathbf{v}$, describing the velocity of the guidewire and $\mathbf{c}$, describing the centerline (c).

2.2 3D vascular motion estimation from fluoroscopic images

The observation space Ω described previously includes the relative position of the device tip with respect to the vessel centerline and target, among other things. When the shape and position of the vessels are changing (e.g. due to cardiac or respiratory motion) we must update this information such that the neural controller can perform optimally.

To recover this motion, it is necessary to use a real-time imaging modality that presents sufficient contrast between the vessels and the surrounding tissue. Fluoroscopy is the only imaging modality that meets these criteria and is currently used in the vast majority of endovascular interventions. However, it requires the injection of a contrast agent to be able to visualize vessels in the image, and it provides only a two-dimensional image.

Various methods have been devised to overcome this limitation and recover 3D motion from a single fluoroscopy, mostly in the context of free-breathing radiotherapy [20, 21, 22]. While these methods demonstrate clinically relevant target localization accuracy, their use of a statistical motion model to generate training data limits them to recovering predetermined motion patterns, which can restrict their clinical applicability. Moreover, the accuracy of these methods to recover the shape of vessels has not yet been evaluated.

2.2.1 Fluoroscopy-based vessel motion prediction

To remediate the lack of methods suited for vessel shape estimation, we presented in a previous work [23] a method to recover, in real-time, the shape of the vessels from a single fluoroscopy. In addition, our method predicts the 3D displacement of the vascular network without requiring the injection of a contrast agent, which is a significant benefit for the patient. Even though fluoroscopy is a 2D projective imaging modality, it is still possible to recover 3D motion from fluoroscopy thanks to the penetrating nature of X-rays. A fluoroscopic image is formed by measuring rays' attenuation from the X-ray source through matter to the detector. Our method, like the works cited above, uses a preoperative CT image as a prior on the patient anatomy. The intra-operative fluoroscopic image is then processed to extract the motion of the anatomy relative to this preoperative CT scan. The fluoroscopic image formation process can be approximated by Eq. 4 below:

$$p(\mathbf{u}) \approx \int I(\mathbf{x})\mathbf{dl_u}, \tag{4}$$

where p is the fluoroscopic image, $\mathbf{u}$ the pixel coordinate, I the CT scan of the patient, $\mathbf{x}$ the voxel coordinate along the projection ray, and l_u the projection ray from the X-ray source to the pixel u. Consequently, a deformation of I that produces a change in the value $p(\mathbf{u})$ will be visible in the fluoroscopic image. Only deformations of I that leave $p(\mathbf{u})$ unchanged, that is motion colinear to the projection ray l_u, cannot be observed in the fluoroscopic image.

This method seamlessly integrates into any fluoroscopy-guided intervention with routine pre-operative CT scan acquisition. Using the pre-operative CT scan of the patient, the vessels and other structures of interest are segmented and the planning of the intervention is performed by the clinician. In our current approach, we assume that the pose of the fluoroscopic imager with respect to the patient's anatomy is determined before the intervention, which is generally true.

2.2.2 Training data generation

The objective of our neural network is to predict a 3D displacement field from a single fluoroscopic image in which the anatomy is viewed in a deformed configuration (see Fig. 4). Since such data is impossible to obtain from actual interventions, we generate both a 3D Displacement Vector Field (DVF) and the associated deformed fluoroscopic image, obtained as a Digitally Reconstructed Radiograph (DRR). These DRR and DVF pairs are computed from the preoperative CT image of the patient, assuming a known pose of the C-arm. To enable the network to recover arbitrary deformations of the anatomy, the DVFs are generated in a randomized way. For the deformation to be realistic, the DVF must be diffeomorphic, meaning that no self-intersection occurs in the deformation. The Large Deformation Diffeomorphic Metric Mapping (LDDMM) framework, which is tailored to meet these conditions,

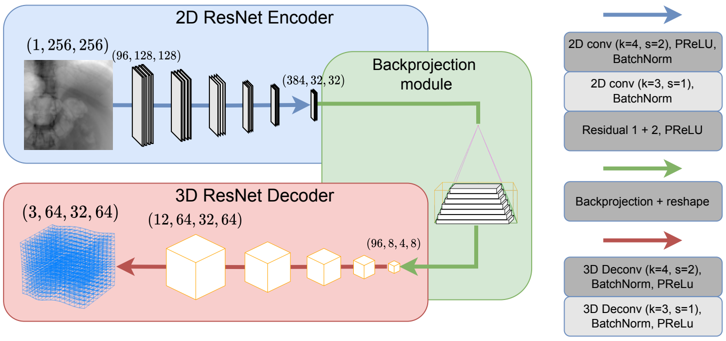

Figure 4: Each block in the Encoder downscales the feature maps and increases their number by a factor of 2. In the Decoder, this is reversed. The last decoder layer transforms the 12 feature maps into a 3-channel 3D image.

emerges as a promising solution [24]. Within this framework, the DVF $\boldsymbol{\phi}$ that registers a deformed image with a reference image is obtained by integrating a set of differential equations in time. At each step of the integration, the parameters of a velocity field $\mathbf{V}(t, \mathbf{x})$ are updated to minimize both an energy-based term and an image similarity term.

The authors show that the velocity field $\mathbf{V}(t, \mathbf{x})$ can be constructed as a sum of Gaussian kernels, each centered at control points $\mathbf{y_k}$ and associated with corresponding weights $\boldsymbol{\alpha_k}$:

$$\mathbf{V}(t, \mathbf{x}) = \sum_{k=1}^{N_{cp}} \boldsymbol{\alpha_k(t)} \cdot \mathbf{K_k}(\mathbf{x}, \mathbf{y_k}(\mathbf{t})).$$

Here, $\mathbf{K_k}(t)$ denotes elements from a Reproducing Kernel Hilbert Space, in this case, Gaussian kernels. The control points $\mathbf{y_k}$, covariance matrices $\boldsymbol{\sigma_k} \in \mathbb{R}^{3\times3}$, and weights $\boldsymbol{\alpha_k}$ of the Gaussian kernels are randomized in our approach to compute a displacement field $\boldsymbol{\varphi}(\mathbf{x})$, that does not depend on time. N_y control points $\mathbf{y_k}$ are sampled by generating random points in the volume, with a rejection process to avoid generating control points that are too close to each other, which would result in sharp variations of $\boldsymbol{\varphi}$. Since $\boldsymbol{\alpha_k}$ are independent of each other, a sample will typically contain a mixture of small, medium, and large displacements. However, to be robust to scenarios where no deformations occur, we randomly scale the DVF with a global scaling parameter chosen between -1 and 1. Additionally, covariance matrices $\boldsymbol{\sigma_k}$ are generated as $N_y \times 3 \times 3$ i.i.d. variables within a specified range. Finally, one DRR is generated for each deformation of the preoperative CT using the DeepDRR framework [25].

Our fully convolutional network architecture (detailed in Fig. 4) is based on the architecture described in our previous work [23], with some modifications. First, a convolutional encoder extracts a condensed feature representation from the 2D input image. We increased the number of convolutional filters in the first layer from 64 to 96, which slightly improved performances. Then, the 2D feature maps are transformed into 3D feature maps using back-projection instead of a simple reshaping as in our previous work. The back-projection algorithm uses the C-arm camera matrix to transform the 2D feature maps in the 2D fluoroscopic image space to 3D feature maps in the preoperative 3D CT space. Owing to the local nature of convolutions, the feature maps output by the encoder are in the same space as the input image. We treat the feature maps F extracted by the encoder as a single-channel 3D image, where the original channel dimension of F corresponds to a third spatial dimension, corresponding to the depth along projection rays. This is motivated by the fact that the input image is formed by measuring the attenuation of rays traveling from the projection center to each pixel on a 2D detector, following Eq. 4. To transform the feature maps into the same space as the pre-operative CT scan, we make use of the projection matrix $\mathbf{P}$, formed by the intrinsic parameters of the C-arm and the C-arm pose with respect to the CT scan. First, we define a regular grid of voxel coordinates $\mathbf{G}$ in the 3D CT scan space. We convert $\mathbf{G}$ to a grid of pixel coordinates $\mathbf{G_u} = \mathbf{PG}$ in 2D homogeneous coordinates u_i. u_0 and u_1 represent the height and width of the projection of the voxel on the detector plane, while u_2 represents the depth of the voxel along the projection ray. Since convolutions are local operations, we can index F in the same way as the input image, by normalizing G_u to the range $[-1, 1]$. By interpolating F using G_u, we obtain a 3D volume F_{3D} in the same space as the preoperative CT scan. This process is described in Fig. 5, where F is back-projected to obtain F_{3D}. F_{3D} is non-zero only in the regions that are both in the CT volume and visible in the projection, tracing a square frustum.

F_{3D} is a three-dimensional image without a feature dimension. To transform F_{3D} into a 3D feature map, a feature dimension is created by splitting the depth dimension into a depth and a feature dimension. This choice is motivated by the fact that the depth dimension was originally the channel dimension of F before the back-projection process. Finally, F_{3D} is decoded into a 3D displacement field by a fully convolutional 3D decoder. This displacement field represents the non-rigid transform of the CT to the anatomy visible in the fluoroscopic image. The network is trained using an MSE Loss between the synthetic ground truth displacement field and the predicted displacement field.

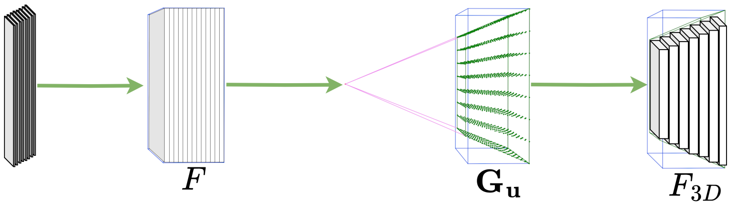

Figure 5: The 2D feature maps are reshaped into a volume F in ray space. The volume is then sampled using ray space coordinates $\mathbf{G_u}$, in green, to obtain a volume F_{3D} in CT space. Elements of F_{3D} that are outside of the view frustum are set to zero.

Since the neural network receives a single 2D fluoroscopic image as input, the 3D motion estimation task is inherently an ill-posed problem. In particular, the 3D motion component in the direction of the projection rays cannot be observed in the image. According to Eq. 4, any motion along the projection rays l_u will not induce a change in image intensity. Indeed, the value of the line integral along l_u is not modified by a voxel displacement along p_u. To train the network, we found it necessary to compute the MSE Loss only on the prediction components parallel to the image plane. This limits the ability of the network to predict the motion of the anatomy perpendicular to the image plane and induces errors in the prediction of the displacement field. We show below that our control is, however, robust to this uncertainty.

3 Results

We illustrate the performance of our neural controller and fluoroscopy-based motion estimation in two examples. The first one is a typical example of endovascular navigation, where a guidewire is advanced in the coronary arteries of a beating heart. In the second example, we show a scenario where a guidewire is advanced through the venous system of the liver, as done during the diagnosis and treatment of portal hypertension. Based on the method described in Section 2.1 and using Stable Baselines3's SAC implementation [26], we train the neural controller using a learning rate of 10^{-4}, a buffer size of 10,000 and a batch size of 256. The discount factor is set to 0.98 and the entropy coefficient is learned during the training. The actor and the critic networks are composed of three 256-neuron layers and the model, updated at every time step, is trained for 175,000 time steps, with a dt of $0.01\,s$. The whole training process only requires 6 hours of computation on an Intel(R) Core(TM) i7-13700KF processor with 32 GB of RAM. The training anatomies are generated as explained in Section 2.1 with a constant vessel diameter of $4\,mm$. A suitable guidewire is used to navigate the anatomies, with a $4.5\,mm$ long tip and a tip curvature of $0.38\,mm^{-1}$. For each anatomy and test case, the controller is evaluated on a total of 100 episodes, where an episode is defined as the navigation from the insertion point to the target location. An episode is considered successful if the guidewire, steered by the controller, reaches the target location. Four and five distinct target locations were chosen for respectively the heart and the liver, each involving the navigation of a minimum of 2 and a maximum of 4 bifurcations. For each test episode, a random target and a random starting rotation of the guidewire were chosen. We report in Table 1 the percentage of successful episodes for each test.

3.1 Navigation in coronary arteries during cardiac motion

In this example, we demonstrate the ability of our neural controller to navigate a dynamic environment without prior training on either this anatomy or this particular deformation (see Fig 6 left). The heart model was reconstructed from Magnetic Resonance imaging data

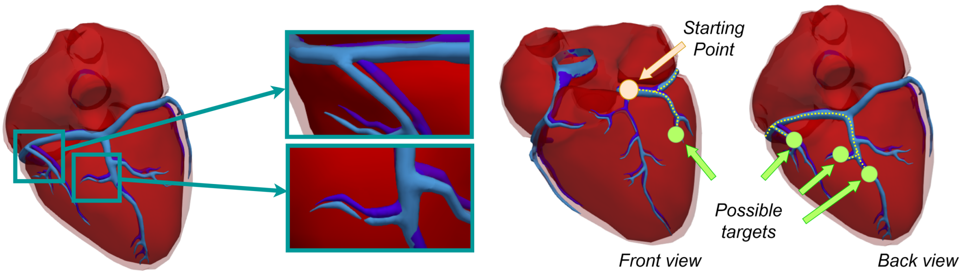

Figure 6: **Right**Visualization of the coronary motion during a cardiac cycle. The heart volume varies by about 12% in 1 second during a cardiac cycle. **Left**Illustration of the different paths and targets the neural controller needs to navigate. The insertion point is shown in orange, and the randomly selected targets are shown in green. An example of a path is depicted as a dotted line.

but the motion was generated synthetically, allowing to know the shape of the vascular tree and the centerline position at each time step. In this case, since the dimensions of the anatomy are similar to the dimensions of the training geometries, with an entry diameter of $3.8\,mm$, the same guidewire used during the training is adopted. The efficacy of the controller is tested in three different scenarios: a static case, in which the anatomy is not moving, a dynamic case in which the heart is beating, but the location of the centerline is not updated and a dynamic case in which the heart is beating and the position of the centerline is updated. For this anatomy, we selected 4 targets shown in the right part of Fig. 6 and randomly chose one for each of the 100 test episodes. In all test cases, our controller proves its ability to navigate the coronaries both in static (90% success rate, Table 1 a) and dynamic (97% success rate, Table 1 c) conditions, maintaining its performance also when navigating the anatomy without any knowledge about vessel deformation (89% success rate, Table 1 b).

3.2 Navigation in hepatic veins during respiratory motion

In this section, we focus on a different clinical context, such as the endovascular treatment of hepatic venous outflow obstruction [27] or the endovascular treatment of portal hypertension [28]. The key difference when compared to the previous scenario is that, in this case, the vascular tree motion caused by the breathing of the subject is unknown, engaging ourselves in a true clinical scenario, in which the moving anatomy is only visible in fluoroscopic images, in 2D. Using the neural network described in Section 2.2, we estimate, in real-time, the 3D position of the vessels' centerlines. Using a patient's abdominal CT scan, a training dataset, composed of 18,000 samples, was generated and used to train the motion prediction neural network, as described in Sec. 2.2.2. The test dataset contains a series of fluoroscopic images covering 5 inhale/exhale periods for a total of 50 samples. The main direction of motion is along the Inferior-Superior (IS) and Antero-posterior (AP) axes, with a small motion in the Left-Right (LR) direction, and a sliding motion of the organs against the thoracic cage. The trained network was evaluated using the Target Registration Error (TRE) on the hepatic veins centerlines and the hepatic veins mesh. Across the testing dataset, the mean displacement was $8.74{\pm}4.06\,mm$ and $8.66{\pm}4.05\,mm$ while the mean TRE was $3.74{\pm}2.33\,mm$ and $3.84 \pm 2.37\,mm$ for the centerlines and the hepatic veins respectively. This error is not similar in each direction, since the motion along the direction perpendicular to the image plane is more difficult to estimate compared to the other two directions. This is reflected in the error of the network, which was, on average, below $2\,mm$ for the IS and LR directions and, on average, below $2.6\,mm$ for the AP direction.

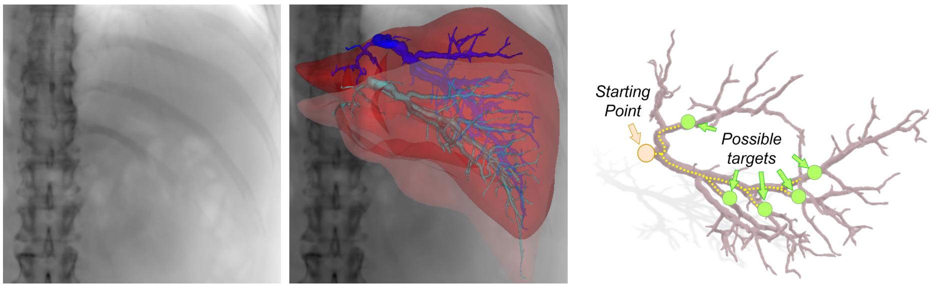

Figure 7: **Left**: fluoroscopic image seen by the neural network. **Center**: preoperative position and shape of the liver and its venous system (opaque colors) and prediction of the 3D shape (in semi-transparent color) of both the liver shape and its vascular tree. The centerlines of the veins are also predicted, in real-time, and used by the neural controller. **Right** Liver venous system with 5 different targets. The controller has to navigate the moving anatomy from the insertion point (in orange) until the designated target, randomly chosen among the possible targets, in green.

We report in Table 1 the success rate of our controller, in similar conditions as presented for the heart: a static case, a dynamic case without centerline update, and a dynamic case in which the updated position of the anatomy is reconstructed from non-contrasted fluoroscopic images, thanks to our neural network. Given the dimension of the anatomy, a different guidewire, with a $6.5\,mm$-long tip and a tip curvature of $0.26\,mm^{-1}$ is used. For this second test anatomy, we chose 5 different target locations, as shown in Fig. 7 right, and randomly selected one for each of the 100 test episodes. In this more complex context, our controller demonstrates its efficacy with an 89% success rate (Table 1 d) in the static scenario, which is maintained when transitioning to dynamic conditions. In this case, the agent reports a success rate of 93% (Table 1 f), while it shows an important performance drop (24% success rate, Table 1 e) when trying to navigate the dynamic anatomy without any information regarding the movement of the vessels (no centerline update in the observation space Ω). This shows the significance of our 3D vascular motion estimation.

4 Discussion

As illustrated by the results in Table 1, our new controller demonstrates its ability to navigate new complex anatomies, composed of various subsequent bifurcations, different in shapes and dimensions, with a mean success rate of 95% in the dynamic anatomies. This is very significant compared to the probability of reaching the designated targets when taking random actions which would lead to an average success rate of 15% for the two scenarios we have considered. This value can be obtained by considering the set $\boldsymbol{Z} = \{z_1, z_2, ..., z_N\}$ of N different targets, each requiring the successful navigation of a number, b_{z_i}, of bifurcations. The mean probability of successfully reaching the target is equal to $\frac{1}{N}\sum_{j=1}^{N} 1/2^{b_{z_j}}$.

We compared our new controller with the agent described in our previous work [12], trained for 150,000 time steps of 0.01 s on anatomies with a $4\,mm$ diameter, consistent with the diameter of the training anatomies used in this work, allowing to use the same guidewire. The new controller outperforms our previous version in terms of both robustness to vessel motion (Table 1 c) and robustness to variations in vessel dimensions (Table 1, *Liver*). The important performance loss shown by [12] when navigating the liver anatomy, can be explained by the difference in the dimension of the liver geometry, whose entry vessel presents a diameter of $7\,mm$, which almost doubles the diameter of the training anatomies. The high success rate obtained by our controller in the liver's static conditions (Table 1 d)

Table 1: Navigation results summary, in heart and liver.

Heart			
Conditions	Success rate [%]		
	[12]	Our method	Random
Static[a]	100%	90%	15.6%
Dynamic, no centerline update[b]	68%	89%	
Dynamic, centerline update[c]	82%	**97%**	

Liver			
Conditions	Success rate [%]		
	[12]	Our method	Random
Static[d]	36%	89%	15%
Dynamic, no centerline update[e]	50%	24%	
Dynamic, centerline updated with our NN prediction[f]	50%	**93%**	

demonstrates the adaptability of our new training strategy to anatomies presenting various bifurcation shapes and dimensions. However, when navigating a dynamic environment, the performances of the controller critically drop if the centerlines are not updated (see Table 1 e). Our neural network allows computing the new position of the vessels, thus reducing the difference between the real anatomy and the anatomy observed by the controller. In these conditions, our controller demonstrates its ability to navigate various dynamic anatomies both when a synthetic movement is generated (Table 1 c) and when the anatomy moves following real vessel movements (Table 1 f).

5 Conclusion

In this paper, we presented a neural controller, based on a deep reinforcement learning approach, able to navigate a guidewire in complex, unseen, moving anatomies with various dimensions. In addition, we proposed a method for estimating the 3D motion of the anatomy from single-view fluoroscopy images, even without the injection of a contrast agent. The combination of these two contributions makes it possible to automatically perform endovascular navigation in close to real-world conditions, as illustrated in two scenarios: a beating heart and a liver deformed under breathing motion. Our method makes it possible to reach random targets within these anatomies with an average success rate of 95%. To the best of our knowledge, this has never been achieved before.

Although our method already accounts for certain real-world conditions (use of actual anatomies generated from patient data, motion estimation from fluoroscopy), a natural future development of this work will consist in testing the neural controller in a vascular phantom. This will require the use of an FBG-based shape sensing method to reconstruct the shape of the guidewire, and an access to an endovascular robot to apply the action taken by the controller to the device. We have already started working on the shape reconstruction from FBG data, and we will first assess the robustness of our controller in a rigid phantom.

Declarations

Financial Interests: The authors have no relevant financial or non-financial interests to disclose.

References

[1] A Puschel, C Schafmayer, and J Groß. Robot-assisted techniques in vascular and endovascular surgery. *Langenbecks Arch Surg*, 407(5):1789–1795, 2022.

[2] Zhang Jingyu, Lilu Liu, Pingyu Xiang, Qin Fang, Xiuping Nie, Honghai Ma, Jian Hu, Rong Xiong, Yue Wang, and Haojian Lu. Ai co-pilot bronchoscope robot. *Nature Communications*, 15, 01 2024.

[3] Wei Tian, Jian Guo, Shuxiang Guo, and Qiang Fu. A DDPG-based method of autonomous catheter navigation in virtual environment. In *Proc. International Conference on Mechatronics and Automation*, pages 889–893, 2023.

[4] Shuang Wang, Zheng Liu, Xiongpeng Shu, Yongfeng Cao, Ling Zhang, and Le Xie. Study on autonomous delivery of guidewire based on improved yolov5s on vascular model platform. In *2022 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pages 1–6, 2022.

[5] Jihoon Kweon, Kyunghwan Kim, Chaehyuk Lee, Hwi Kwon, Jinwoo Park, Kyoseok Song, Young In Kim, Jeeone Park, Inwook Back, Jae-Hyung Roh, Youngjin Moon, Jaesoon Choi, and Young-Hak Kim. Deep Reinforcement Learning for guidewire navigation in coronary artery phantom. *IEEE Access*, 9:166409–166422, 2021.

[6] Lennart Karstensen, Tobias Behr, Tim Philipp Pusch, Franziska Mathis-Ullrich, and Jan Stallkamp. Autonomous guidewire navigation in a two dimensional vascular phantom. *Current Directions in Biomedical Engineering*, 6(1), 2020.

[7] Victor Miranda, Armando A. Neto, Gustavo M. Freitas, and Leonardo A. Mozelli. Generalization in Deep Reinforcement Learning for robotic navigation by reward shaping. *IEEE Transactions on Industrial Electronics*, pages 1–8, 2023.

[8] Robert Kirk, Amy Zhang, Edward Grefenstette, and Tim Rocktäschel. A survey of zero-shot generalisation in Deep Reinforcement Learning. *Journal of Artificial Intelligence Research*, 76:201–264, 2023.

[9] Lennart Karstensen, Jacqueline Ritter, Johannes Hatzl, Floris Ernst, Jens Langejürgen, Christian Uhl, and Franziska Mathis-Ullrich. Recurrent neural networks for generalization towards the vessel geometry in autonomous endovascular guidewire navigation in the aortic arch. *Int. Journal of Computer Assisted Radiology and Surgery*, 18:1735–1744, 9 2023.

[10] Wenqiang Chi, Jindong Liu, Hedyeh Rafii-Tari, Celia Riga, Colin Bicknell, and Guang Zhong Yang. Learning-based endovascular navigation through the use of non-rigid registration for collaborative robotic catheterization. *International Journal of Computer Assisted Radiology and Surgery*, 13:855–864, 6 2018.

[11] Wenqiang Chi, Giulio Dagnino, Trevor M. Y. Kwok, Anh Nguyen, Dennis Kundrat, Mohamed E. M. K. Abdelaziz, Celia Riga, Colin Bicknell, and Guang-Zhong Yang. Collaborative robot-assisted endovascular catheterization with generative adversarial imitation learning. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 2414–2420, 2020.

[12] Valentina Scarponi, Michel Duprez, Florent Nageotte, and Stéphane Cotin. A zero-shot reinforcement learning strategy for autonomous guidewire navigation. *International Journal of Computer Assisted Radiology and Surgery*, 19:1185–1192, 2024.

[13] S. Singh, N. Ajayi, L. Lazarus, and K. S. Satyapal. Anatomic study of the morphology of the right and left coronary arteries. *Folia Morphologica*, 76(4):668 – 674, 2017.

[14] Richard Bellman. A Markovian decision process. *Journal of Mathematics and Mechanics*, 6(5):679–684, 1957.

[15] Tuomas Haarnoja, Aurick Zhou, Kristian Hartikainen, G. Tucker, Sehoon Ha, Jie Tan, Vikash Kumar, Henry Zhu, Abhishek Gupta, P. Abbeel, and Sergey Levine. Soft Actor-Critic algorithms and applications, 2018.

[16] Francois Faure, Christian Duriez, Hervé Delingette, Jeremie Allard, Benjamin Gilles, Stéphanie Marchesseau, Hugo Talbot, Hadrien Courtecuisse, Guillaume Bousquet, Igor Peterlik, and Stéphane Cotin. *SOFA: A Multi-Model Framework for Interactive Physical Simulation*, volume 11, pages 283–321. Springer, 06 2012.

[17] Ibrahim Bitar, Stéphane Grange, Panagiotis Kotronis, and Nathan Benkemoun. A review on various formulations of displacement based multi-fiber straight timoshenko beam finite elements. In *Proc. CIGOS*, 2015.

[18] Franck Jourdan, Pierre Alart, and Michel Jean. A gauss-seidel like algorithm to solve frictional contact problems. *Computer Methods in Applied Mechanics and Engineering*, 155(1):31–47, 1998.

[19] Omar Al-Ahmad, Mouloud Ourak, Jan Van Roosbroeck, Johan Vlekken, and Emmanuel Vander Poorten. Improved fbg-based shape sensing methods for vascular catheterization treatment. *IEEE Robotics and Automation Letters*, 5(3):4687–4694, 2020.

[20] Ran Wei, Fugen Zhou, Bo Liu, Xiangzhi Bai, Dongshan Fu, Bin Liang, and Qiuwen Wu. Real-time tumor localization with single x-ray projection at arbitrary gantry angles using a convolutional neural network (cnn). *Physics in Medicine & Biology*, 65(6), 2020.

[21] Megumi Nakao, Mitsuhiro Nakamura, and Tetsuya Matsuda. Image-to-graph convolutional network for 2d/3d deformable model registration of low-contrast organs. *IEEE Transactions on Medical Imaging*, 41(12):3747–3761, 2022.

[22] Hua-Chieh Shao, Yunxiang Li, Jing Wang, Steve Jiang, and You Zhang. Real-time liver motion estimation via deep learning-based angle-agnostic x-ray imaging. *Medical Physics*, 2023.

[23] Francois Lecomte, Valentina Scarponi, Pablo A Alvarez, Juan M Verde, Jean-Louis Dillenseger, Eric Vibert, and Stéphane Cotin. Enhancing fluoroscopy-guided interventions: a neural network to predict vessel deformation without contrast agents. In *Hamlyn Symposium on Medical Robotics*, pages 75–76, 2023.

[24] Alain Trouve, Mirza Faisal Beg, Michael I Miller, and Laurent Younes. Computing Large Deformation Metric Mappings via Geodesic Flows of Diffeomorphisms. *International Journal of Computer Vision*, 61(2):139–157, 2005.

[25] Mathias Unberath, Jan Nico Zaech, Sing Chun Lee, Bastian Bier, Javad Fotouhi, Mehran Armand, and Nassir Navab. DeepDRR – A Catalyst for Machine Learning in Fluoroscopy-Guided Procedures. *Lecture Notes in Computer Science*, 11073 LNCS:98–106, September 2018.

[26] Antonin Raffin, Ashley Hill, Adam Gleave, Anssi Kanervisto, Maximilian Ernestus, and Noah Dormann. Stable-baselines3: Reliable reinforcement learning implementations. *Journal of Machine Learning Research*, 22(268):1–8, 2021.

[27] P. Ghibes, C. Artzner, S. Partovi, F. Hagen, S. Nadalin, and G. Grozinger. Endovascular treatment of symptomatic hepatic venous outflow obstruction post major liver resection. *BMC Gastroenterol*, 23(1), 2023.

[28] Yosef Golowa and Jacob Cynamon. *Endovascular Treatment of Portal Hypertension*, chapter 85, pages 1095–1106. John Wiley & Sons, Ltd, 2012.