

Christoph Eyberg*, Lennart Karstensen, Tim Pusch, Johannes Horsch, and Jens Langejürgen

A ROS2-based Testbed Environment for Endovascular Robotic Systems

<https://doi.org/10.1515/cdbme-2022-0023>

Abstract: Developing autonomous endovascular robotic systems requires physical testbeds to test control algorithms. Typically, such testbeds comprise of several hard- and software components along with a way of having these components communicate with each other. Building such a testbed is a multidisciplinary task which can be beyond the scope of expertise for research groups. The goal of this work is to facilitate setting up such testbeds in two ways: First, we propose a testbed architecture that allows to develop tracking, control and instrument manipulation systems separately by utilizing the ROS2 communication protocol. Secondly, we present a reliable yet straightforward to implement tracking algorithm for endovascular instruments that is built using only open-source software packages. The tracking algorithm is evaluated using both video camera and x-ray imaging and is found to meet the requirements for real time control algorithms. Furthermore, we show an example of the proposed modular testbed architecture as it is used in our lab. Both the modular testbed architecture and the open-source tracking algorithm may serve as helpful building blocks for other researchers in the field seeking to evaluate their control algorithms on physical testbeds.

Keywords: endovascular, surgical robotics, testbed, tracking, guidewire,

1 Introduction

Vascular diseases, especially ischemic heart (16%) and cerebrovascular disease (11.2%), are the leading causes of death worldwide [1]. The endovascular intervention is a minimally invasive surgical method to diagnose and treat these diseases. During this type of intervention thin and flexible

instruments, i.e. guidewire and catheter, are navigated through the patient's vascular system to the site of the lesion under medical imaging, where the treatment is performed. Navigating through the vascular system is a complex task that exposes the surgeon to radiation and requires the usage of contrast agent to guide the surgeon through the vascular system.

Current research aims to automate the navigation task in order to enhance patient safety and allow the physicians to focus on the actual treatment while reducing the required dosage of contrast agent and radiation [2–4]. Yet, the development and testing of automated guidewire and catheter navigation in physical testbeds is challenging as it requires solving multiple tasks at once: Receiving feedback about the position of the instruments, e.g. by using the unfiltered medical image or tracking the instrument position, determining an appropriate navigation manoeuvre to move the instruments closer to the target position and executing this manoeuvre.

A frequently used approach to tracking guidewires is to represent the guidewire with a B-spline and then update the position of its control points at every step by optimizing an energy function building both on image features and mechanical plausibility [5–7]. Vandini *et al.* [8] search for image features that possibly represent guidewire segments and combine them to find the guidewire. In recent years convolutional neural networks have been successfully used to extract the position of guidewires from fluoroscopy images [9–11]. For application in control algorithms a sufficiently high tracking frequency and a low tracking induced delay are necessary. Clinicians typically use an image frequency of 4–10Hz while the stated research is able to process images within 50–175ms. These values can be used as a baseline requirement for tracking algorithms in real time control loops for autonomous guidewire navigation. Implementation and runtime optimization of such professional solutions can be a challenging task, however, it might not be necessary during early stages of development of endovascular robotic systems. In these stages facile solutions and interfaces that allow easy replacement of each part of the testbed are required.

The contribution of this paper is twofold: We present an architecture for a testbed using the ROS2 [12] interface which

*Corresponding author: **Christoph Eyberg:** Fraunhofer IPA, Theodor-Kutzer-Ufer 1-3, Mannheim, Germany, e-mail: Christoph.Eyberg@ipa.fraunhofer.de
Lennart Karstensen, Tim Pusch, Johannes Horsch, Jens Langejürgen Fraunhofer IPA, Mannheim, Germany

allows decentralized and wireless communication between the different components of the testbed. This allows to solve each task independently and substitute between different solutions, e.g. switching to a state-of-the-art tracking solution when the stage of animal or clinical trials is reached. Secondly, a tracking algorithm is suggested that is solely based on open source packages and yet meets the stated performance requirements and therefore allows in-vitro development and testing of navigation algorithms and instrument manipulators.

2 Method

2.1 Testbed Architecture

The testbed consists of a transparent phantom of a vascular system, which is either placed inside in an x-ray imaging system (Artis Zeego, Siemens Helthineers) or mounted with a camera, a laptop with a 12-core, 2.6GHz processor and image

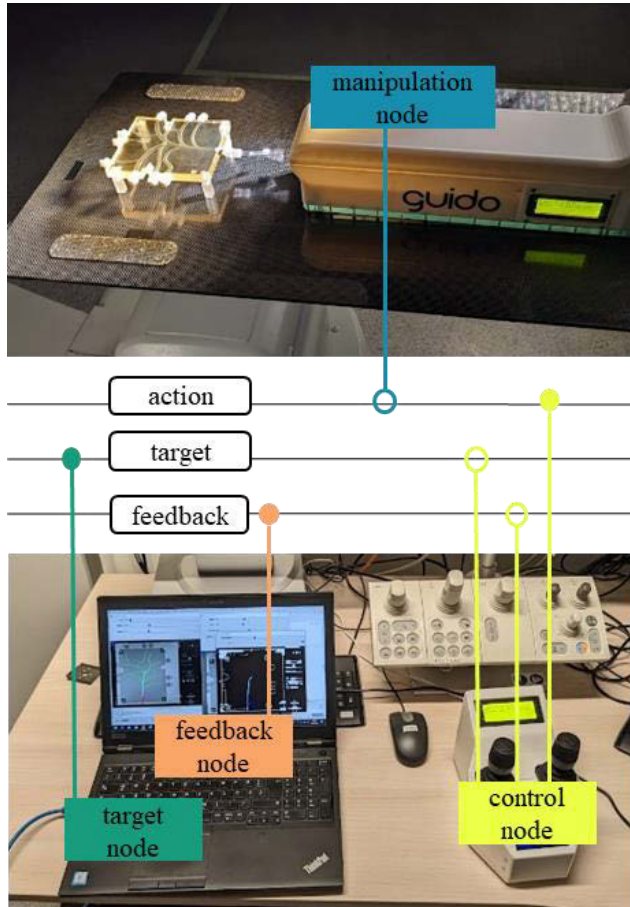


Figure 1: Phantom of vascular system with guidewire manipulator under X-ray imaging (top), ROS topics (center) and guidewire tracking with joystick control pad in control room (bottom). A filled circle represents messages being published to the topic while a ring represents a subscription

processing software, a guidewire manipulator, and a joystick. The communication between the different elements is performed through the ROS2 interface. This also allows us to place all control devices inside the secure control room omitting exposure to any radiation. The setup is shown in Figure 1. The ROS2 architecture is built using the following nodes:

- The feedback node, orange, receives the live webcam or fluoroscopy images and obtains tracking points of the guidewire. Those tracking points are then published to the feedback topic.
- The target node, green, publishes the target position for the guidewire tip to the target topic. The target position can be chosen in the displayed image.
- The control node, yellow, subscribes to both the target and the feedback topic and determines the control signal which is the desired translational and rotational velocity for the guidewire. The control message is published to the action topic for each incoming feedback message.
- The manipulation node, blue, subscribes to the action topic and moves the guidewire accordingly.

This architecture allows to easily substitute solutions for the different nodes e.g. replacing the manually controlled joystick in the control node with an autonomous control algorithm. Analogously, feedback and manipulation node can be replaced by a simulation. Additionally, multiple control nodes with different priorities can be utilized, e.g. to allow manual override via the joystick while navigating autonomously.

2.2 Tracking Algorithm

Our novel guidewire tracking algorithm is placed inside the feedback node to retrieve guidewire tracking points from the image. The tracking algorithm uses the standard image processing functionalities of the publicly available Open CV [13] package. Instead of relying on complex algorithms our approach builds on manual parameter tuning utilizing two facts: First, the endovascular instruments are inserted at a static position which is known during the intervention. Second, endovascular instruments are slender devices, i.e. $l_z \gg l_r$, where l_z is the straight length and l_r is the radius of the guidewire.

Before the tracking loop is started, brightness and contrast of the incoming video stream can be manually adjusted. This allows to obscure the edges of the vascular tree and to enhance the visibility of the guidewire, when using a camera image. Likewise the thresholds for the edge detection can be hand tuned. Furthermore, the insertion area is manually marked (red rectangle in Figure 2) and the coordinate system for the position of the guidewire is specified. In our setup the phantom

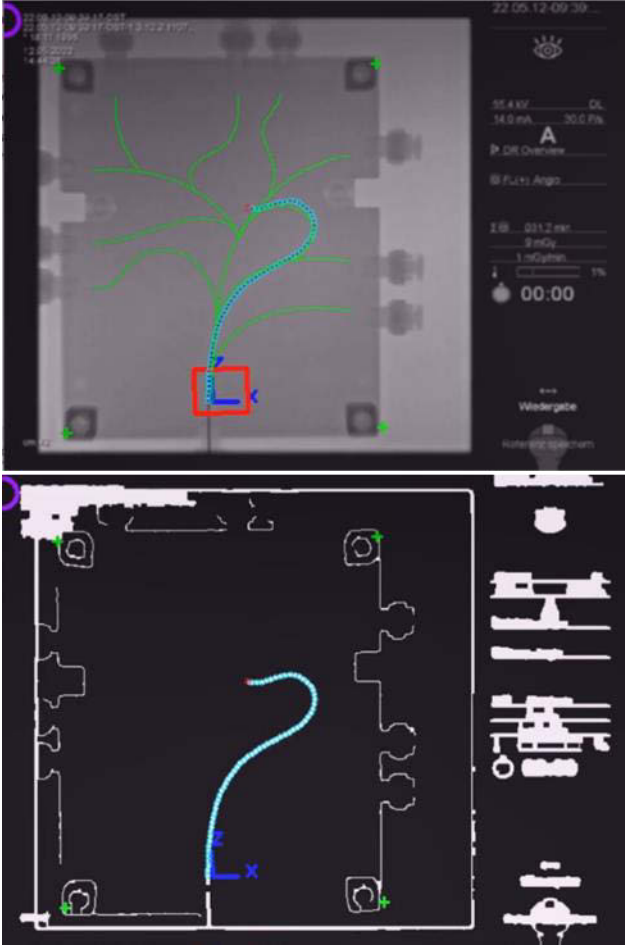


Figure 2: Fluoroscopy image with additionally displayed vessel tree (top) and processed image with tracked guidewire (bottom).

of the vascular system is placed inside the x-z-plane with the guidewire pointing into the direction of the z-axis. This corresponds to the coordinate system of the x-ray imaging system. To track the guidewire, the following steps are performed:

1. Transformation to grayscale image and adjustment of brightness and contrast as specified.
2. Extracting the edges of the guidewire into a binary image using the Canny-Edge-Detector [14].
3. Applying a closing operation (consecutive usage of dilation and erosion filters) to fill up the edge-contours of the slender guidewire to achieve an enclosed contour.
4. Retrieving the shapes and a polygon approximation of all enclosed contours in the binary image.
5. Iterating through the approximation points of all contours until the guidewire is found as the contour with points inside the insertion area.
6. Finding the tip of the guidewire by measuring the distance of both edges starting from the edge of the insertion area in positive coordinate direction. The tip is found as the point where both edges meet under equal distance.

7. Retrieving and publishing tracking points that represent the centreline of the two edges of the guidewire.

Figure 2 displays the successful retrieval of the guidewire from the fluoroscopy image. In evenly illuminated images it was also found possible to substitute steps 2 and 3 by a threshold filter. Setting all binary image values on the lower edge of the insertion area to zero after step 3 can additionally assert that a contour approximation point is placed inside the insertion area. Substituting step 6 by using the angle between the edges of the polygon approximation or adding a correction for guidewire bending during distance calculation were found to be less efficient.

2.3 Experiment Setup

We validate our setup and evaluate the tracking algorithm by manually navigating through all branches of the vascular phantom using the joystick controller, as displayed in Figure 2, resulting in images with a variety of guidewire positions and lengths. During this task the delay of the image processing from retrieving a new image until returning the tracking points and the frequency of incoming tracking signals is measured. Also the number of time steps, where the algorithm is not able to retrieve a guidewire position from the image is counted. The experiment is conducted using both camera and x-ray imaging. Additionally, the experiment is repeated placing the vessel phantom on top of an anatomic phantom in x-ray imaging to assess the capability of the tracking algorithm in a setup closer to the clinical application.

3 Evaluation

The tracking algorithm achieves an average processing delay of less than 50ms under both camera and x-ray imaging which is below the baseline of current research. The achieved control frequency matches or even exceeds the stated requirements of 4-10 Hz. All values are gathered in Table 1.

Table 1: Average values for control frequency and delay, and percentage of time-steps where the guidewire was not found

	Control Frequency	Processing Delay	Guidewire not found
Camera Tracking	17Hz	41ms	0.0%
x-ray Tracking	10Hz	25ms	0.7%

While the guidewire was tracked in the camera image in all time steps, there were a few time steps where the guidewire could not be tracked or the guidewire tip was not found correctly using the fluoroscopy image input. This was due to rapid guidewire movements under relatively low image frequency which resulted in a smeared guidewire. In the experiment an imaging frequency of 30p/s was used. The failure rate increases with a lower image frequency.

The tracking algorithm failed to distinguish the guidewire from the background when the vessel phantom was placed on top of the anatomic phantom, due to equal illumination of guidewire and skeletal structures under x-ray imaging. The tracking algorithm also fails to extract the correct guidewire tip if the guidewire overlaps itself or is kinked by maloperation.

4 Discussion

A testbed for endovascular robotic systems was presented. The utilization of the ROS2 communication protocol divides the navigation task into the subtasks of receiving a feedback about the instruments position, finding an appropriate manoeuvre and executing this manoeuvre. This allows researchers to focus on each individual task during development while having the ability to easily replace different solutions for all other components.

Furthermore, a tracking algorithm for guidewires was presented that can be implemented straightforwardly using open source software packages. It meets the requirements for application in navigation control loops and reliably tracks the guidewire in two-dimensional vascular phantoms.

The testbed allows researchers to evaluate control algorithms or robots for endovascular instruments by replacing the respective ROS2 node with their solution. Algorithms that successfully navigate through the presented physical testbed are promising candidates for testing in phantoms with higher complexity or even animals. The tracking algorithm can then easily be replaced by a professional solution.

Author Statement

Research funding: This project is funded by the Ministry of Economics, Labour and Tourism Baden-Württemberg within the framework of the Forum Gesundheitsstandort Baden-Württemberg.

Conflict of interest: Authors state no conflict of interest.

References

- [1] World Health Organization (2020) Global Health Estimates 2019 Summary Tables: Deaths by cause, age and sex, by world bank income group, 2000-2019. https://www.who.int/docs/default-source/gho-documents/global-health-estimates/ghe2019_cod_global_2000_20194e572f53-509f-4578-b01e-6370c65d9fc5.xlsx?sfvrsn=eaf8ca5_7. Accessed 16 May 2022
- [2] Karstensen L, Ritter J, Hatzl J et al. (2022) Learning-based autonomous vascular guidewire navigation without human demonstration in the venous system of a porcine liver. *International Journal of Computer Assisted Radiology and Surgery*
- [3] Schegg P, Dequidt J, Coevoet E et al. (2022 - 2022) Automated Planning for Robotic Guidewire Navigation in the Coronary Arteries. In: 2022 IEEE 5th International Conference on Soft Robotics (RoboSoft). IEEE, pp 239–246
- [4] Kweon J, Kim K, Lee C et al. (2021) Deep Reinforcement Learning for Guidewire Navigation in Coronary Artery Phantom. *IEEE Access* 9:166409–166422. <https://doi.org/10.1109/ACCESS.2021.3135277>
- [5] Baert SAM, Viergever MA, Niessen WJ (2003) Guide-wire tracking during endovascular interventions. *IEEE Trans Med Imaging* 22:965–972. <https://doi.org/10.1109/TMI.2003.815904>
- [6] Slabaugh G, Kong K, Unal G et al. (2007) Variational Guidewire Tracking Using Phase Congruency. In: Springer, Berlin, Heidelberg, pp 612–619
- [7] Chang P-L, Rolls A, Praetere H de et al. (2016) Robust Catheter and Guidewire Tracking Using B-Spline Tube Model and Pixel-Wise Posteriors. *IEEE Robot Autom Lett* 1:303–308. <https://doi.org/10.1109/LRA.2016.2517821>
- [8] Vandini A, Glocker B, Hamady M et al. (2017) Robust guidewire tracking under large deformations combining segment-like features (SEGlets). *Medical Image Analysis* 38:150–164. <https://doi.org/10.1016/j.media.2017.02.001>
- [9] Zhou Y-J, Xie X-L, Bian G-B et al. (2019) Fully Automatic Dual-Guidewire Segmentation for Coronary Bifurcation Lesion. In: 2019 International Joint Conference on Neural Networks (IJCNN). IEEE, Piscataway, NJ, pp 1–6
- [10] Gherardini M, Mazomenos E, Menciassi A et al. (2020) Catheter segmentation in X-ray fluoroscopy using synthetic data and transfer learning with light U-nets. *Computer Methods and Programs in Biomedicine* 192:105420. <https://doi.org/10.1016/j.cmpb.2020.105420>
- [11] Wagner MG, Laeseke P, Speidel MA (2019) Deep learning based guidewire segmentation in x-ray images. In: Gilat-Schmidt T, Chen G-H, Bosmans H (eds) *Medical Imaging 2019: Physics of Medical Imaging*: 17-20 February 2019, San Diego, California, United States. SPIE, Bellingham, Washington, USA, p 150
- [12] Macenski S, Foote T, Gerkey B et al. (2022) Robot Operating System 2: Design, architecture, and uses in the wild. *Sci Robot* 7:eabm6074. <https://doi.org/10.1126/scirobotics.abm6074>
- [13] Bradski G (2000) The OpenCV Library. In: Dr. Dobb's Journal of Software Tools
- [14] Canny J (1986) A Computational Approach to Edge Detection. *IEEE Trans Pattern Anal Mach Intell PAMI* 8:679–698. <https://doi.org/10.1109/TPAMI.1986.4767851>