

Automatic keypoint generation in learning-by-demonstration trajectories for a surgical robot

Manrique-Córdoba, J.^{a,*}, Poveda-Pérez, M.^a, Martorell-Llobregat, C.^b, Sabater-Navarro, J.M.^a

^a Grupo de Robótica Médica. Instituto de Bioingeniería. Universidad Miguel Hernández de Elche, 03202. Elche, España.

^b Neurosurgery unit. Hospital General de Elche, 03202. Elche, España.;

Resumen

La robótica quirúrgica es una de las áreas más prometedoras de la robótica; sin embargo, la mayoría de los sistemas robóticos aplicados a la cirugía operan bajo un esquema de teleoperación directa. Para mejorar el rendimiento de estos sistemas, sería deseable aumentar el nivel de autonomía de los robots quirúrgicos. El objetivo principal de este artículo es desarrollar una metodología para la generación automática de trayectorias aprendidas mediante la demostración de procedimientos quirúrgicos. Se presenta el detalle de un algoritmo para obtener puntos clave en las trayectorias obtenidas de los cirujanos. Estos puntos clave contienen no solo información geométrica, sino también cinemática y el campo de fuerza ejercido por el cirujano. La codificación presentada permite la implementación de un esquema de aprendizaje de demostración basado en modelos ocultos de Markov, específico para operaciones quirúrgicas. La principal contribución de nuestra propuesta reside en el uso de puntos clave obtenidos de todas las demostraciones y la integración de la información de velocidad y fuerza con la información geométrica. Para ello, se presentan los resultados experimentales obtenidos en varias pruebas realizadas en un sistema especialmente diseñado para obtener las fuerzas y trayectorias realizadas por neurocirujanos en un procedimiento de fresado de hueso mastoideo.

Palabras clave: robótica quirúrgica, aprendizaje por demostración.

Abstract

Surgical robotics is one of the most promising areas of robotics, however most robotic systems applied to surgery operate under a direct teleoperation scheme. In order to improve the performance of these systems, it would be desirable to increase the level of autonomy of surgical robots. The main objective of this paper is to develop a methodology for the automatic generation of trajectories learned by demonstrating surgical procedures. The detail of an algorithm for obtaining keypoints in the trajectories obtained from surgeons is presented. These keypoints contain not only geometric information, but also kinematics and the force field exerted by the surgeon. The coding presented allows the implementation of a demonstration learning scheme based on hidden Markov models particularized for surgical operations. The major contribution of our proposal approach is the use of keypoints obtained from all the demonstrations and the integration of speed and force information together with geometric information. For this, the experimental results obtained in several tests carried out in a setup specially designed to obtain the forces and trajectories carried out by neurosurgeons in a mastoid bone milling procedure are presented.

Keywords: surgical robotics, learning by demonstration.

1. Introduction

Surgical robotics is one of the most successful areas of robotics, with broad acceptance, impressive economic returns, and highly active research and development communities. In 2015, more than 650,000 procedures were performed worldwide (Sridhar, et al., 2017), since then there has been a 25% annual growth in the number of procedures (Mayor et al.,

2022). In Europe, most robotic procedures are performed in the field of urology, while in the US, gynecology and general surgery led the sector (Sridhar, et al., 2017). Existing commercial surgical robots are mostly focused on orthopedic procedures, neurosurgical interventions, and minimally invasive surgical techniques. The rapid spread of this technology has been largely due to the perceived benefits of improved ergonomics, dexterity, safety, and ease of surgery.

*Autor para correspondencia: jmanrique@umh.es

One of the main goals in robotic surgery has been on developing robotic platforms that can work together with the surgeon. Robotic systems have evolved according to the different tasks to be performed and the possible interfaces with the surgeon. Therefore, many studies have focused their efforts on automating certain actions in the surgical environment (Bauzano Nuñez, et al., 2015). This article addresses the idea of improving autonomous behavior in performing simple and repetitive tasks that a surgeon can execute during a surgical procedure, for which the robotic system must be indicated both the trajectory and the forces that it must follow with the goal of completing said task.

The trajectories that are indicated to the robot can come from a planning carried out from preoperative images, or they can be supported by the information obtained from the performance of the same procedures by experienced surgeons. This information can be obtained through surgical navigation systems, which is extended to systems that allow 3D localization of the different actors involved in the surgical procedure.

Usually, the information acquired from the trajectories can be very dense, obtained at a sampling frequency higher than the communication that can be established with the robot, which complicates making an identical replica of the sampled trajectory. There are different geometry and/or trajectory simplification algorithms, especially focused on cartographic generalization applications, among the best known are the Douglas-Peucker algorithm (Douglas & Peucker, 1973), the Visvalingam–Whyatt algorithm (Visvalingam & Whyatt, 1993), the sleeve-fitting algorithm (Zhao & Saalfeld, 1997) and the Reumann–Witkam algorithm (Rangayyan, et al., 2008), which are developed for two dimensions.

Given that robotic surgery seeks cooperation between the robot and the surgeon, a robotic system that can be adapted to the limitations that this situation presents are collaborative robots, which are designed to share the workspace with humans, unlike conventional industrial robots. Collaborative robots tend to be much lighter in weight, with great mobility, and flexibility, allowing them to be programmed to perform a wide variety of tasks (Sherwani, et al., 2020). The collaborative robots developed by Universal Robots (UR) are commonly used in different activities in the industrial, medical, and educational fields, as well as a wide range of applications (Vivas & Sabater, 2021). In particular, these robots have an integrated teach pendant that allows their programming. This programming can be done by guidance, indicating the waypoints that make up the trajectories to be followed, or by establishing the coordinates of said points.

The work presented aims to obtain resampled trajectories with a significantly lower number of points, potentially defining parametric key points, so that those can be sent to a surgical robot that replicates the trajectories learned from the surgeon without compromising other parameters such as precision, velocity or interaction forces exerted.

The work is organized as follows. Section II summarizes the equipment and the experimentation carried out, presenting the details of the implemented resampling algorithm and explaining the proposed codification. Section III shows the results obtained both with digitally generated virtual trajectories and with real trajectories executed by surgeons and captured by an optical surgical navigation system. A real trajectory for milling in a mastoid reaming procedure These

results are briefly commented in section IV along with the conclusions and future work.

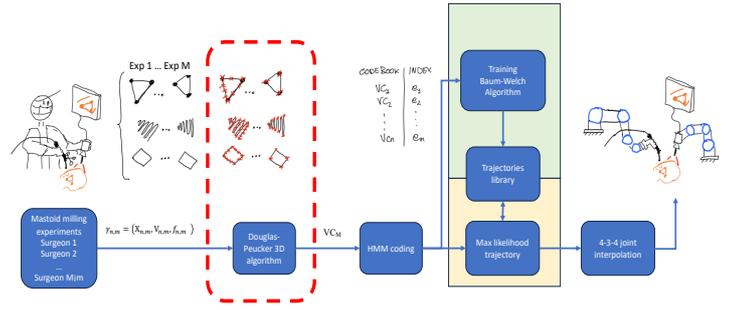


Figure 1- Schematic workflow for learning by demonstration in surgical procedures.

2. Methods

An schematic flowchart representation of the complete process is shown in Fig 1. First, the trajectories of the surgeon's hands are tracked and coded to be learned. The capture of the demonstrated trajectories is carried out with an optical tracking system (Optitrack V120:TRIO) which captures the position of the infrared optical markers attached to tool. The read trajectories are encoded and the kinematic and force information is added. Data Acquisition. The processes and modules needed to obtain the correct interpretation of the maneuvers during a surgical protocol are shown. Contained in the green box are the modules that are used in the offline process for training, while in the yellow box the modules that participate in the online process for the recognition of maneuvers are shown.

In this paper we focus on the process of extracting information and coding it to make it understandable to Hidden Markov models. Characteristic vectors (VCM) are constructed containing the relevant information of the trajectory (position and velocity) and forces at each instant of time.

2.1. Data Acquisition

Based on the markers' locations with respect to a fixed reference frame, position and orientation (pose) of predefined rigid bodies are inferred over a set of discrete time instants. The measured trajectories from the perception phase are denoted by

$$\Gamma_m = (\gamma_{1,m}, \gamma_{2,m}, \dots, \gamma_{N_m,m})_{m=1}^M \quad m=1, \dots, M. \quad (1)$$

where m is used for indexing the demonstrations, M refers to the total number of demonstrations, and N_m denotes the number of measurements of the demonstration Γ_m .

Each position measurement is a D-dimensional, in this case a 6 D geometric vector containing position and orientation, and the position vector is denoted as:

$$X_{n,m} = (x_{1,m}^{(1)}, x_{1,m}^{(2)}, \dots, x_{1,m}^{(D)}) \quad (2)$$

for $n = 1, 2, \dots, N_m, m = 1, 2, \dots, M$.

The velocity values for each dimension of the recorded trajectories were calculated using an empirically chosen delay value of p sampling periods, i.e

$$v_{n,m}^{(d)} = (x_{n+p,m}^{(d)} - x_{n-p,m}^{(d)}) / (t_{n+p,m} - t_{n-p,m}) \quad (3)$$

for $n = 1, \dots, N_m$ and $d = 1, \dots, D$, where d denotes the dimensionality of the sequences and t denotes the time instants. Then, the vector of the velocities is denoted as:

$$V_{n,m} = (v_{1,m}^{(1)}, v_{1,m}^{(2)}, \dots, v_{1,m}^{(D)}) \quad (4)$$

The force is computed as the variation of force between two instants of time:

$$f_{n,m} = (f_{n+p,m} - f_{n-p,m}) / (t_{n+p,m} - t_{n-p,m}) \quad (5)$$

Being that each demonstration is coded as:

$$\gamma_{n,m} = (X_{n,m}, V_{n,m}, f_{n,m}) \quad (6)$$

2.2. Douglas-Peucker algorithm extended for 3 dimensional position information

First, a modification is presented to adapt the classic Douglas-Peucker algorithm to a 3D space, containing only position information of surgical tools.

In order to generate a parametric library of surgical trajectories of different simple maneuvers, a 3D trajectory subsampling method is required, allowing to reduce the order of the learned trajectory while defining key points that can be used as measurement parameters of the trajectory. In addition, a method to represent the obtained key points in the 3D space and the time associated with said points is required.

As mentioned in the introduction, the Douglas-Peucker algorithm is an algorithm mainly used in 2D spaces for cartography applications. This work relies on the geometric principle of the algorithm to be applied in a 3D space. Thus, the Douglas-Peucker algorithm implemented starts from a matrix R and a tolerance ϵ ; the matrix R , size $n \times 4$, each row condenses the information of the n points of the trajectory, and each point stores in the information of its X, Y, Z coordinates and the time t when that point must be reached.

$$R = [P_1, P_2, \dots, P_n] \quad (7)$$

$$P_n = [X_n, Y_n, Z_n, t_n]^T \quad (8)$$

The algorithm verifies that $n > 2$, if this inequality is not true R is returned as the optimized trajectory, otherwise the distance d that exists between the points P_1 and P_n is obtained:

$$d = \sqrt{(Z_n - Z_1)^2 + (Y_n - Y_1)^2 + (X_n - X_1)^2} \quad (9)$$

Consecutively, it is intended to calculate the perpendicular distance from the line that joins the points P_1 and P_n to the furthest point of the trajectory. With the distance d an evaluation of the relative positions between the points is conducted, if d is less than the minimum computable distance in the software in which the calculation is implemented (in Matlab: $\epsilon = 2.204e^{-16}$), then the distance to the furthest

point will be calculated from point P_1 . Thus, to find the furthest point in the trajectory, a loop is executed from P_2 to P_{n-1} , storing the perpendicular distance between the line joining P_1 and P_n to each point in the trajectory $dp(k)$, for $d < \epsilon$:

$$dp(k) = \sqrt{(Z_k - Z_1)^2 + (Y_k - Y_1)^2 + (X_k - X_1)^2} \quad (10)$$

For $d > \epsilon$:

$$A = \begin{bmatrix} 1 & X_1 & Y_1 & Z_1 \\ 1 & X_n & Y_n & Z_n \\ 1 & X_k & Y_k & Z_k \\ 0 & 1 & 1 & 1 \end{bmatrix} \quad (11)$$

$$dp(k) = |\det(A)| \quad (12)$$

The vector dp stores the distances to each of the points of the trajectory, the distance with the greatest value is selected from it, assuming that the furthest point is P_k , the algorithm evaluates if the distance of this point is greater than the tolerance ϵ defined for this trajectory, if the distance is greater than the tolerance, a recursive call to the algorithm is made with two sub-trajectories, the first from P_1 to P_k and the second from P_k to P_n , from which the reduced trajectory is built; on the other hand, if the maximum distance dp is less than the tolerance then the reduced path will become $[P_1, P_n]$.

2.3. Augmented resampling algorithm for 6D position, velocity and forces

For the implementation of the augmented algorithm, the coding of (6) is used. The augmented trajectories approximate a linear model in state variables, in which the object will be considered to be moving at constant velocity, although subject to acceleration perturbations. Therefore, the linear system is a simple process that can be described with the equation of states (13) and the equation of outputs, where for each demonstration m , the state vector considered $\gamma_{k+1,m}$ is composed of the position, the speed and the increase of force at each instant of time k , $y_{k,m}$ is the measure of the output and u is a known input to the system. The variable w , is the process noise, while z , is the noise in the measure.

$$\begin{aligned} \gamma_{k+1,m} &= D_1 \gamma_{k,m} + D_2 u_{k,m} + w_{k,m} \\ y_{k,m} &= D_3 \gamma_{k,m} + z_{k,m} \end{aligned} \quad (13)$$

2.4. Test protocol

The platform designed for the measurement of forces has great versatility, since it can be adapted to different minimally invasive surgery procedures as can be seen in the different experiments presented in the article. The common set-up of all the experiments has been a force sensor that is placed under the support pieces of the surgical scenario for the different medical procedures, so that there is always the measurement of forces exerted on the surgical scenario. This sensor is from the OnRobot brand and provides precise measurements of force along the 6 dimensions; (3 forces, 3 torques), being able to relate this force with the one that would receive the surrounding tissue to the area of interest in a real surgical intervention. The *OnRobot* sensor is of the high-precision HEX-E/High model.

It should be noted that this experimentation is very versatile since it allows with the same very basic set-up to measure the force exerted by the surgeon. Due to this design, different tests and simulations of minimally invasive surgery can be made, which, although it currently has many advantages for both patients and health personnel, must continue to improve to make it safer and more accurate.

Figure 2 shows an image of the setup showing the placement of the force sensor.

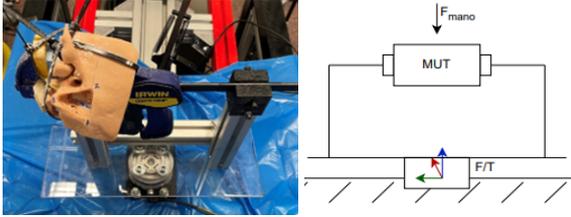


Figure 2. - Setup for recording interaction forces.

3. Results

This section presents the results of the experimentation carried out. Initially, the digitally created trajectories are shown in comparison with their corresponding simplification and the trajectories executed by the robot; Later, the same comparison is shown with real trajectories.

3.1 Digitally generated trajectories

Figure 3-a shows the square trajectory digitally created, it is made up of 80 points and presents a smoothing in its corners, because it is intended to be executed at a constant speed. When simplifying, a tolerance of 1 cm has been indicated, thus the smoothing in the corners of the square is removed, generating a toolpath of 5 points. The average absolute error obtained between the digitally created trajectory and the simplification is 0.334 cm, while the average error between the simplified trajectory and the one carried out by the robot is only 0.123 cm, thereby, in the comparison between the digital trajectory and that carried out by the robot, an average absolute error of 0.3814 cm is obtained.

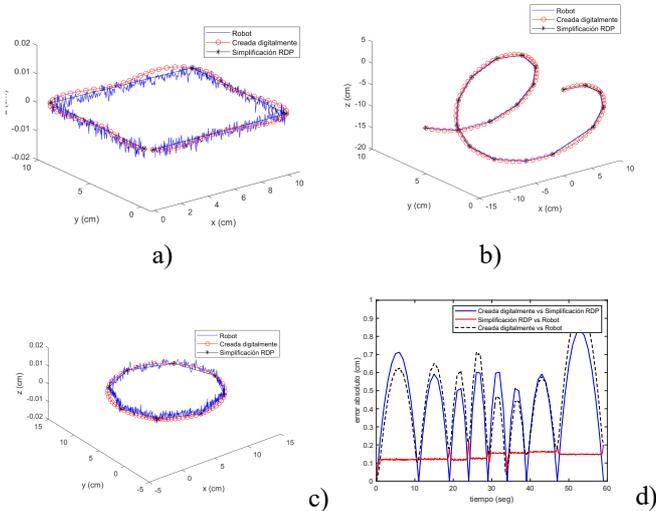


Figure 3.- Digital trajectories resampled.

3.2 Real basic trajectories

Initially, a square trajectory of approximately 10 cm long is drawn (Figure 4), this is done on an elastic surface, so the trajectory shows noise as a result of both the Optitrack measurement and the pulse of the user who executes the trajectory, considering the same tolerance characteristics for simplification, a trajectory reduced to two points is obtained, which is intended to be executed with the robot. From the comparison between the trajectory traced by the user and the simplification, an average absolute error of 0.608 cm is obtained, while the comparison between the simplified path and the one executed by the robot results in an average error of 0.261 cm.

3.3 Real surgical trajectories

This experiment simulates a manual mastoid milling intervention, which is necessary to access the tympanic cavity. In this type of operations it is crucial to measure the force in this type of experiments since when performed with a milling tool a force is being applied that is proportional to the mechanical energy transferred and the heat that is generated, therefore, it must be controlled so as not to perform an unwanted burn to the patient or destroy healthy surrounding tissue.

The set-up consisted of the OnRobot sensor located under a platform that holds a 3D model of the mastoid. Through this platform, the forces are transmitted while milling to the sensor. The design of the experiment can be seen in Figure 2.

The forces recorded by the sensor located on the platform that are presented in this article, are originated by the forces that are exerted in the MUT (*Material Under Test*). The force occurs in all three axes since it depends on the direction of execution of the maneuvers by the surgeons.

Therefore, the force recorded by the F/T sensor is obtained by the sum of several components (equation (14)). These components are the force applied by the surgeon. F_{hand} , the offset of sensor reference values or bias F_b , the forces due to the action of gravity F_g , the characteristic noise of the sensor F_n , forces caused by inertia F_i , and finally, the forces due to the interaction with the external elements or contact forces between the surgical tool and the tissues F_{cont} .

$$F_{sensor} = F_{hand} + F_b + F_g + F_n + F_i + F_{cont} \quad (14)$$

The results obtained in this experiment consist of eight tests of 5 minutes each, since the neurosurgeons have taken 40 minutes to perform the complete milling intervention. Due to this division of the data, it has been possible to observe how the force values decrease as the operation progresses since it goes from a more superficial area in which there is no tissue that can be easily damaged, to a more sensitive area in which there are already blood vessels and nerves.

In figure 5, one of the graphs obtained with the force data on the sensor is presented. It is appreciated how there is a wide variability of movements because it is not a controlled movement, but as shown in figure 2 a three-dimensional milling is performed on the surface of the mastoid.

The trajectory of the second interval with a length of 5-minute is plotted in 3D in Figure 6-a). The original trajectory is 6000 sampling points.

In the surrounding figures the application of the proposed algorithm is observed for a tolerance of 0.001m (6-b) in which 267 keypoints are obtained (VCi), for a tolerance of 0.0025m (6-c) in which 121 keypoints are obtained (VCi) and for a tolerance of 0.005m (6-d) in which 53 keypoints (VCi) are obtained. If it is considered that the diameter of the bur used was 0.003m, it can be concluded that the result 6-c allows a representation of the reliable trajectory, in which the forces exerted have been considered.

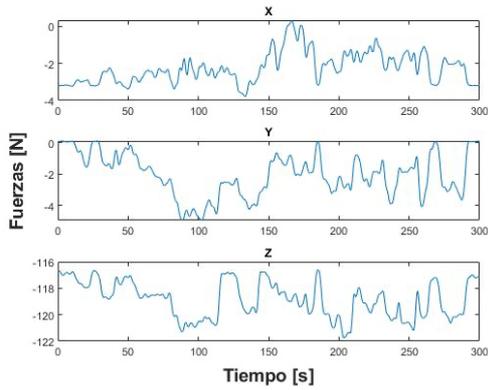


Figure 5. Forces read on the sensor F/T.

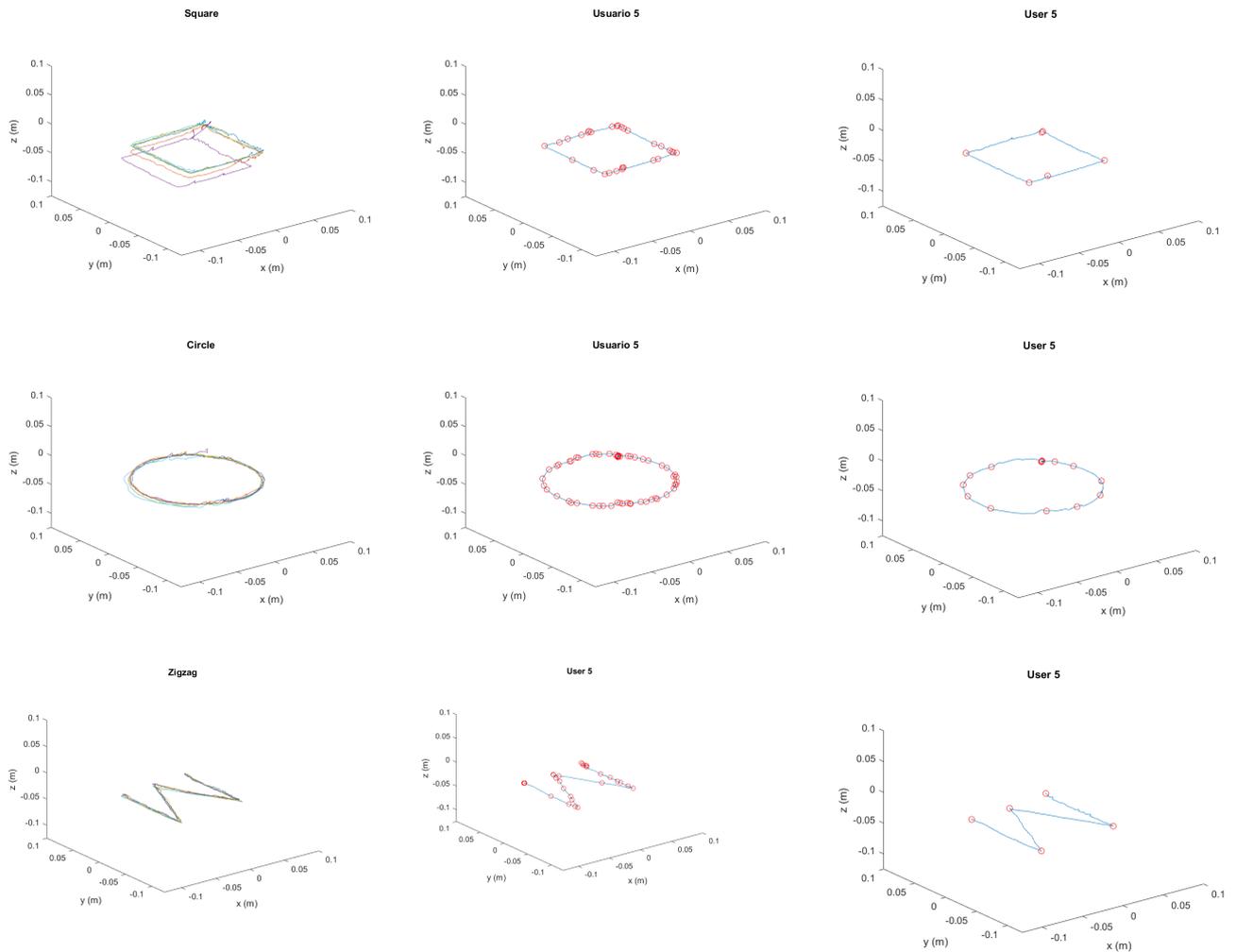


Figure 4. Real basic trajectories.

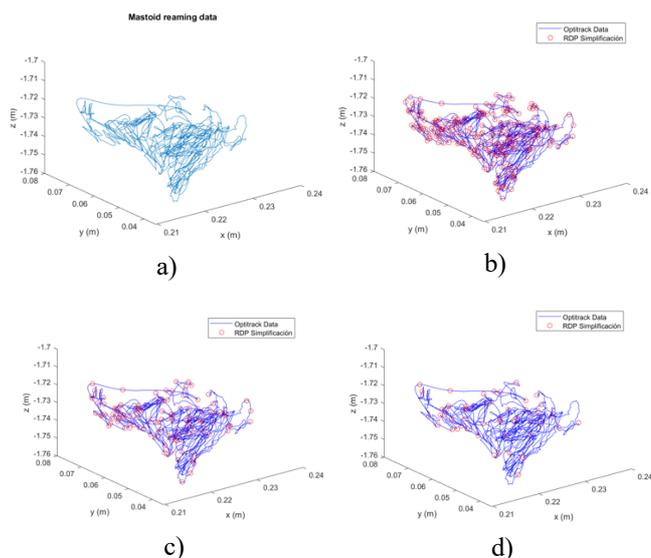


Figure 6. Mastoid milling trajectories. Second interval. Tolerance of 0.001m (6-b) in which 267 keypoints are obtained Tolerance of 0.0025m (6-c) in which 121 keypoints are obtained . Tolerance of 0.005m (6-d) in which 53 keypoints are obtained

4. Discussion and Conclusions

The present work aims to provide an alternative for reducing the number of points in 3D trajectories based on demonstrations, simplifying a very dense amount of data with a simple, easily executable trajectory, where the least amount of information possible is lost. The Douglas-Peucker algorithm is implemented, with modifications so that it can be used in 3D spaces, and kinematic and force information is integrated prior to define the keypoints for learning stages.

The results obtained show that the proposed simplification based on the Douglas-Peucker algorithm introduces an error directly related to the tolerance established in the simplification, thus, being able to vary the amount of data that represents the trajectory and therefore its error. On the other hand, the error obtained between the RDP simplification and the trajectory executed by the robot is lower, which is related to the noise of the data read from the robot and the precision it has when executing the movements. In this proposal, the execution time of each trajectory is taken into account, obtaining different results for the same route at different speeds.

This proposal aims to open a path to the execution of simple or standardized parametric surgical trajectories, in scenarios that require an instrument to move from a point A to point B along an established path, regardless of the way in which the path is defined. As future work, the implementation of the

HMM of figure 1 is proposed that allows learning by demonstration algorithms. Likewise, we suggest the possibility that the trajectory can be adaptive, considering that surgical environments may present movements of the elements present in the development of the trajectory, it is interesting to provide an alternative that constantly updates the information of the trajectory.

Acknowledgements

This research was funded by Spanish Government— Agencia Estatal de Investigación (AEI) through the project PID2022-138206OB-C32 and by the Generalitat Valenciana through the project CIPROM/2022/16.

References

- [1] Bauzano Nuñez, E., Estebanez Campos, B., García Morales, I. & Muñoz Martínez, V. F., 2015. Planning Automatic Surgical Tasks for a Robot Assistant. *Motion and Operation Planning of Robotic Systems: Background and Practical Approaches*, pp. 193-220. Springer. <https://doi.org/10.1007/978-3-319-14705-5>
- [2] Douglas, D. H. & Peucker, T. K., 1973. Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. *Cartographica: the international journal for geographic information and geovisualization*, 10(2), pp. 112-122. <https://doi.org/10.3138/FM57-6770-U75U-7727>
- [3] Mayor, N., Coppola, A.S. and Challacombe, B. (2022), Past, present and future of surgical robotics. *Trends Urology & Men Health*, 13: 7-10. <https://doi.org/10.1002/true.834>
- [4] Rangayyan, R. M., Guliato, D., de Carvalho, J. D. & Santiago, S. A., 2008. Polygonal approximation of contours based on the turning angle function. *Journal of Electronic Imaging*, 17(2), pp. 023016-023016. <https://doi.org/10.1117/1.2920413>
- [5] Sherwani, F., Asad, M. M. & Ibrahim, B. S. K., 2020. Collaborative robots and industrial revolution 4.0 (ir 4.0). 2020 International Conference on Emerging Trends in Smart Technologies (ICETST), pp. 1-5. DOI: 10.1109/ICETST49965.2020.9080724
- [6] Sridhar, A. N., Briggs, T. P., Kelly, J. D. & Nathan, S., 2017. Training in robotic surgery—an overview. *Current urology reports*, Volumen 18, pp. 1-8. <https://doi.org/10.1007/s11934-017-0710-y>
- [7] Toker, A., 2014. Robotic thoracic surgery: from the perspectives of European chest surgeons. *Journal of Thoracic Disease*, 6(Suppl 2), p. S211. doi: 10.3978/j.issn.2072-1439.2014.05.05.
- [8] Visvalingam, M. & Whyatt, J. D., 1993. Line generalisation by repeated elimination of points. *The cartographic journal*, 30(1), pp. 46-51. <https://doi.org/10.1179/000870493786962263>
- [9] Vivas, A. & Sabater, J. M., 2021. Ur5 robot manipulation using matlab/simulink and ROS. En: *IEEE International Conference on Mechatronics and Automation (ICMA)*. s.l.:IEEE, pp. 338-343. doi: 10.1109/ICMA52036.2021.9512650
- [10] Weinstein, G. S., O'Malley Jr, B. W., Desai, S. C. & Quon, H., 2009. Transoral robotic surgery: does the ends justify the means?. *Current opinion in otolaryngology & head and neck surgery*, 17(2), pp. 126-131.
- [11] Zhao, Z. & Saalfeld, A., 1997. Linear-time sleeve-fitting polyline simplification algorithms. *Proceedings of AutoCarto*, Volumen 13, pp. 214-222