

Reducción de puntos de una trayectoria 3D basada en el algoritmo Douglas-Peucker. Aplicación a robótica quirúrgica

Manrique-Cordoba, J. *, Cases-Hurtado, J., Romero-Ante, J.D., De la Casa Lillo, M.A., Sabater-Navarro, J.M.

Instituto de Bioingeniería, Universidad Miguel Hernández de Elche, Avda. de la Universidad s/n, 03202, Elche, España.

To cite this article: Manrique-Cordoba, J., Cases-Hurtado, J., Romero-Ante, J.D., De la Casa, M.A., Sabater-Navarro, J.M., 2023. 3D trajectory subsampling based on the Douglas-Peucker algorithm for surgical robotics applications. XLIV Jornadas de Automática, 65-70. <https://doi.org/10.17979/spudc.9788497498609.065>

Resumen

El avance de la autonomía de robots quirúrgicos está ligado con avances en el aprendizaje de trayectorias hechas por cirujanos expertos al realizar procedimientos sencillos. Un primer paso en este aprendizaje es mejorar la capacidad de obtener trayectorias robóticas validas a partir de las maniobras captadas del cirujano. Las trayectorias manuales son habitualmente muestreadas con un número elevado de puntos que dificulta su inserción en algoritmos de aprendizaje. Este trabajo tiene como objetivo obtener trayectorias remuestreadas con un número significativamente menor de puntos y con la posibilidad de definir puntos clave paramétricos, de forma que puedan usarse para ser enviadas a un robot quirúrgico que replique las trayectorias aprendidas del cirujano sin comprometer otros parámetros como precisión, velocidad o fuerzas de interacción ejercidas. El artículo propone un algoritmo de simplificación basado en el algoritmo de Douglas-Peucker para reducir la cantidad de puntos que componen las trayectorias potencialmente complejas; una vez se tiene la simplificación, se define la velocidad cartesiana de movimiento a partir de la diferencia de los puntos ubicados en el espacio. El trabajo presenta los resultados obtenidos al ejecutar las trayectorias con un robot UR3e y se evalúa la simplificación hecha. Los resultados presentados evidencian que las trayectorias submuestreadas de esta forma pueden ser utilizadas en escenarios quirúrgicos para el aprendizaje de procedimientos autónomos.

Palabras clave: remuestreo de trayectorias, Robótica quirúrgica, Trayectorias 3D.

3D trajectory subsampling based on the Douglas-Peucker algorithm for surgical robotics applications.

Abstract

The advancement of the autonomy of surgical robots is linked to advances in the learning of trajectories made by expert surgeons when performing simple procedures. A first step in this learning is to improve the ability to obtain valid robotic trajectories from the captured maneuvers of the surgeon. Manual trajectories are usually sampled with a high number of points that makes it difficult to insert them into learning algorithms. This work aims to obtain resampled trajectories with a significantly lower number of points and with the possibility of defining parametric keypoints, so that they can be used to 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 article proposes a simplification algorithm based on the Douglas-Peucker algorithm to reduce the number of points that make up potentially complex trajectories; Once the simplification is obtained, the Cartesian speed of movement is defined from the difference of the points located in space. The work presents the results obtained when executing the trajectories with a UR3e robot and the simplification made is evaluated. The results presented show that trajectories subsampled in this way can be used in surgical scenarios for learning autonomous procedures.

Keywords: Trajectory re-sampling, Surgical robotics, 3D trajectories.

1. Introducción

La cirugía robótica ha aumentado exponencialmente en la última década. En 2015, se realizaron más de 650000 procedimientos en todo el mundo (Sridhar, et al., 2017), y desde entonces se mantiene un crecimiento de un 25% anual en el número de procedimientos (Mayor et al., 2022). En Europa, la mayoría de los procedimientos robóticos se realizan en el campo de la urología, mientras que en los EE. UU., la ginecología y la cirugía general lideraron el sector (Sridhar, et al., 2017). Los robots quirúrgicos comerciales existentes están enfocados en su mayoría a procedimientos ortopédicos, intervenciones de neurocirugía y técnicas de cirugía mínimamente invasiva. La rápida difusión de esta tecnología se ha debido en gran medida a los beneficios percibidos de una mejor ergonomía, destreza, seguridad y facilidad de cirugía.

Uno de los principales objetivos en la cirugía robótica se ha enfocado en desarrollar plataformas robotizadas que puedan trabajar en conjunto con el cirujano; los sistemas robóticos han evolucionado de acuerdo con las diferentes tareas a realizar y en las posibles interfaces con el cirujano. Por lo anterior, muchos estudios han orientado sus esfuerzos en automatizar ciertas acciones en el entorno quirúrgico (Bauzano Nuñez, et al., 2015). En este artículo se aborda la idea de avanzar en el comportamiento autónomo en la realización de tareas sencillas y repetitivas que pueda realizar un cirujano durante un procedimiento quirúrgico, para ello se le debe indicar al sistema robótico tanto la trayectoria como el campo de fuerzas que debe seguir con el objetivo de completar dicha tarea.

Las trayectorias que son indicadas al robot pueden proceder de una planificación realizada a partir de imágenes preoperatorias, o bien, se pueden apoyar en la información obtenida a partir de la realización de los mismos procedimientos por cirujanos experimentados. Esta información puede obtenerse a través de sistemas de navegación quirúrgica, la cual se extiende a los sistemas que permiten la localización 3D de los diferentes actores que intervienen en el procedimiento quirúrgico.

En muchas ocasiones la información así adquirida de las trayectorias puede ser muy densa, obtenida a una frecuencia de muestreo superior a la comunicación que se puede establecer con el robot, lo que complica realizar una réplica idéntica de la trayectoria muestreada. Existen diferentes algoritmos de simplificación de geometrías y/o trayectorias, especialmente enfocados a aplicaciones de generalización cartográfica, entre los más conocidos resaltan el algoritmo Douglas-Peucker (Douglas & Paucker, 1973), el algoritmo Visvalingam-Whyatt (Visvalingam & Whyatt, 1993), el algoritmo sleeve-fitting (Zhao & Saalfeld, 1997) y el algoritmo Reumann-Witkam (Rangayyan, et al., 2008), los cuales están desarrollados para espacios 2D.

Dado que la cirugía robótica busca una cooperación entre el robot y el cirujano, una alternativa de sistema robótico que puede adaptarse a las limitaciones que esta situación presenta son los robots colaborativos, que son diseñados para compartir espacio de trabajo con humanos, a diferencia de los robots industriales. Los robots colaborativos suelen ser robots mucho más ligeros en peso, con una gran movilidad, permitiendo que sean desplazados de un lugar a otro con facilidad, y con mucha flexibilidad, permitiendo que sean programados para llevar a

cabo una amplia variedad de tareas (Sherwani, et al., 2020). Los robots colaborativos desarrollados por Universal Robots (UR) son comúnmente usados en diferentes actividades en el ámbito industrial, médico y educacional, así como una amplia gama de aplicaciones (Vivas & Sabater, 2021); en particular, estos robots tienen integrado un teach pendant que permite su programación, esta programación se puede hacer por guiado, indicando los puntos de paso que conforman las trayectorias que debe realizar, o bien estableciendo las coordenadas de dichos puntos.

El trabajo presentado aquí pretende obtener trayectorias remuestreadas con un número significativamente menor de puntos y con la posibilidad de definir puntos clave paramétricos, de forma que puedan usarse para ser enviadas a un robot quirúrgico que replique las trayectorias aprendidas del cirujano sin comprometer otros parámetros como precisión, velocidad o fuerzas de interacción ejercidas.

El trabajo se organiza de la siguiente manera. La sección 2 de materiales y métodos resume los equipos y la experimentación realizada, presentando el detalle del funcionamiento del algoritmo de remuestreo implementado. La sección 3 muestra los resultados obtenidos tanto con trayectorias virtuales generadas digitalmente como con trayectorias reales ejecutadas por cirujanos y captadas por un sistema óptico de navegación quirúrgica. Estos resultados son brevemente comentados en la sección 4 junto con las conclusiones y futuros trabajos.

2. Materiales y métodos

En esta sección se presenta una descripción de los elementos más relevantes en el desarrollo de la investigación; entre ellos se encuentra, la definición del algoritmo implementado, el software y hardware utilizado y el flujo de trabajo.

2.1. Remuestreo de trayectoria 3D por algoritmo Douglas-Peucker

Para poder generar una librería paramétrica de trayectorias quirúrgicas en diferentes maniobras sencillas, se precisa en primer lugar un método de submuestreo de trayectorias en 3D, que permita reducir el orden de la trayectoria aprendida a la vez que definir puntos clave que puedan usarse como parámetros de la trayectoria. Además, se requiere de un método que permita representar los puntos obtenidos en el espacio 3D con el tiempo de paso que tiene asociado cada punto.

Como se menciona en la introducción, el algoritmo de Douglas-Peucker es un algoritmo principalmente utilizado en espacios 2D para aplicaciones de cartografía; en este artículo se parte del principio geométrico del algoritmo para ser aplicado en un espacio 3D. Así, el algoritmo de Douglas-Peucker implementado parte de una matriz R y una tolerancia ϵ ; la matriz R es de tamaño $n \times 4$, donde cada fila condensa la información de los puntos n de la trayectoria, cada punto guarda en las tres primeras columnas la información de sus coordenadas X, Y, Z y su última columna corresponde al tiempo t en el que debe alcanzarse dicho punto.

$$R = [P_1, P_2, \dots, P_n] \quad (1)$$

$$P_n = [X_n, Y_n, Z_n, t_n] \quad (2)$$

Al inicio del algoritmo se verifica que $n > 2$, caso contrario, se retorna R como trayectoria optimizada, luego se obtiene la distancia d que existe entre los puntos P_1 y P_n :

$$d = \sqrt{(Z_n - Z_1)^2 + (Y_n - Y_1)^2 + (X_n - X_1)^2} \quad (3)$$

A continuación, se pretende calcular la distancia perpendicular desde la línea que une los puntos P_1 y P_n hasta el punto más alejado de la trayectoria. Con la distancia d se realiza una evaluación de las posiciones relativas entre los puntos, si d es menor que la mínima distancia computable en el software en el que se implementa el cálculo (en Matlab: $eps = 2.204e^{-16}$), entonces la distancia al punto más alejado se calculará desde el punto P_1 . Así, para encontrar el punto más alejado en la trayectoria se ejecuta un bucle de P_2 hasta P_{n-1} , que almacena la distancia perpendicular entre la línea que une a P_1 y P_n hasta cada punto en la trayectoria $dp(k)$, para $d < eps$:

$$dp(k) = \sqrt{(Z_k - Z_1)^2 + (Y_k - Y_1)^2 + (X_k - X_1)^2} \quad (4)$$

Para $d > eps$:

$$A = \begin{bmatrix} 1 & X_1 & Y_1 & Z_1 \\ 1 & X_n & Y_n & Z_n \\ 1 & X_k & Y_k & Z_k \\ 0 & 1 & 1 & 1 \end{bmatrix} \quad (5)$$

$$dp(k) = |\det(A)| \quad (6)$$

Ahora el vector dp almacena las distancias hasta cada uno de los puntos de la trayectoria, de este se selecciona la distancia de mayor valor, suponiendo que el punto más alejado es P_k , se evalúa si la distancia de este punto es mayor que la tolerancia ε definida para esta trayectoria, en caso de ser mayor a la tolerancia se realiza una llamada recursiva al algoritmo con dos trayectorias, la primera desde P_1 hasta P_k y la segunda desde P_k hasta P_n , a partir del cual se construye la trayectoria reducida; por otra parte, si la distancia dp máxima es menor que la tolerancia entonces la trayectoria reducida pasará a ser $[P_1, P_n]$.

2.2. Robot UR3e

Para el desarrollo de pruebas con un robot se utilizó el robot colaborativo UR3e, fabricado por Universal Robots; entre las características principales de este robot resaltan su tamaño reducido y su efector final que permite un movimiento en 360° , además de su capacidad de cargar hasta 3 Kg, lo que facilita su implementación en entornos quirúrgicos.

Por otra parte, ROS (Robotic Operating System) es un middleware de robótica de código abierto para el desarrollo de software para robots. Es ampliamente utilizado para apoyar plataformas de investigación. Los robots colaborativos UR, cuentan con un driver que permiten su manipulación a través de ROS. En el presente proyecto se implementa una interfaz en ROS que establece la comunicación con el robot y permite que

sea manipulado por comandos de velocidad cartesiana, con la intención de seguir la trayectoria definida en el tiempo estipulado. Adicionalmente, se hace uso del protocolo de intercambio de datos en tiempo real, RTDE por sus siglas en inglés, para obtener las posiciones, tanto articulares como cartesianas del efector final, con el objetivo de realizar la comparación de la trayectoria definida, con la trayectoria ejecutada.

2.4. Control por velocidad

Como se menciona anteriormente, los robots colaborativos pueden ser programados de diferentes formas, debido a que el objetivo de la implementación del presente trabajo se enfoca en que el robot pueda replicar las trayectorias ya definidas, por una parte, se pueden indicar puntos cartesianos al robot, o bien indicar las velocidades cartesianas a las que se debe desplazar el efector final del robot; dado que interesa controlar el tiempo de ejecución de la trayectoria y la velocidad a la que se mueve el efector final, se opta por la segunda alternativa, calculando la velocidad de movimiento a partir de la diferencia de la distancia entre los puntos de la trayectoria y el tiempo al que debe ser ejecutado, con esta información calculada, se indican los datos al robot a través del driver *ur_robot_driver* de ROS, como comando de tipo URScript.

2.5. Optitrack™ V120:TRIO

Para la adquisición de datos de la trayectoria realizada sobre un espacio 3D se hace uso de la herramienta Optitrack V120:TRIO, un sistema óptico-pasivo de bajo costo que permite el registro de movimiento de objetos/personas, midiendo la posición y orientación de estos en el espacio físico, este sistema utiliza marcadores retro-reflectivos sobre el objeto a seguir que son rastreados por un conjunto de tres cámaras infrarrojas. Se utilizó una herramienta con 5 marcadores fijos, permitiendo registrar la posición y orientación de la herramienta al realizar la trayectoria en el espacio.

2.4. Protocolo de pruebas

La experimentación propuesta se divide en dos etapas, en la primera etapa se definen 4 trayectorias virtuales en Matlab®, una línea recta de 10cm en 2D, un cuadrado de 10cm de lado en 2D, un círculo de 12 cm de diámetro en 2D y una espiral de 20 cm de diámetro con 10 cm de recorrido horizontal en 3D; se generan los puntos que describen cada trayectoria a velocidad constante con una frecuencia de 1 Hz, y se repite este proceso para todas las trayectorias a una velocidad variable, acelerando y desacelerando en puntos aleatorios; una vez se tienen definidos los puntos, se realiza una simplificación con el algoritmo de Douglas-Peucker (RDP) en 3D para posteriormente calcular las velocidades que se le indicarán al robot a través de la comunicación ROS establecida. La segunda etapa se centra en repetir el procedimiento, pero en esta ocasión con trayectorias reales realizadas por un cirujano y muestreadas. Se realizan las trayectorias 3D con una herramienta a la que se colocan marcadores, para ser registrada con el Optitrack™, el cual muestrea con una frecuencia de 120

Hz, simplificando los datos con el algoritmo Douglas-Peucker en 3D para que la trayectoria pueda ser replicada por el robot.

3. Resultados

En esta sección se presentan los resultados de la experimentación realizada. Inicialmente se muestran las trayectorias creadas digitalmente en comparación con su simplificación correspondiente y la trayectoria ejecutada por el robot; posteriormente se muestra la misma comparación realizada con las trayectorias obtenidas con el dispositivo Optitrack™.

3.1. Trayectorias generadas digitalmente

Para la trayectoria de línea recta creada digitalmente, Figura 1, se compone por 60 puntos dispuestos a lo largo del eje X, cuando se realiza la simplificación de esta trayectoria se obtiene un conjunto de datos de dos puntos, uno inicial y uno final que componen la trayectoria; si bien el robot ejecuta la trayectoria correctamente, se presenta un ruido de alrededor de 0.05 cm.

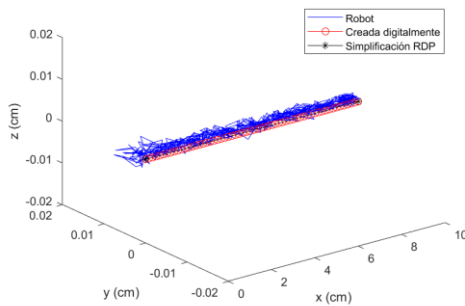


Figura 1: Trayectoria lineal a velocidad constante

El promedio del error absoluto obtenido entre la trayectoria creada digitalmente y su simplificación es de 5.611×10^{-16} cm, mientras que el error obtenido entre la trayectoria creada digitalmente y la trayectoria ejecutada por el robot es de 0.0593 cm.

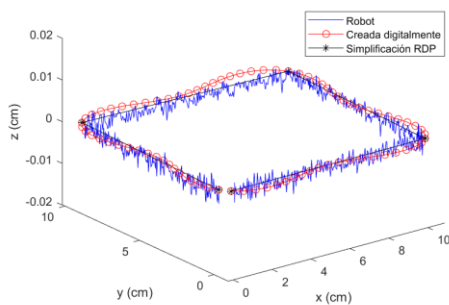


Figura 2: Trayectoria cuadrada a velocidad constante

La Figura 2 muestra la trayectoria cuadrada creada digitalmente, esta se compone por 80 puntos y presenta un suavizado en sus esquinas, debido a que se pretende que sea ejecutada a una velocidad constante, al realizar la simplificación, se ha indicado una tolerancia de 1 cm por lo

que se elimina el suavizado en las esquinas del cuadrado, generando una trayectoria de 5 puntos. El promedio del error absoluto obtenido entre la trayectoria creada digitalmente y la simplificación es de 0.334 cm, mientras que el promedio de error entre la trayectoria simplificada y la realizada por el robot es únicamente de 0.1232 cm, así, en la comparación entre la trayectoria digital y la realizada por el robot se obtiene un promedio de error absoluto de 0.3814 cm.

Si se visualiza el error absoluto a lo largo del tiempo, el error presentado entre la simplificación RDP con respecto a la trayectoria ejecutada por el robot, mostrado en rojo en la Figura 3, permanece relativamente constante sin superar los 0.2 cm, la principal razón para que este error se mantenga es que la velocidad no es alcanzada en el tiempo establecido ya que la herramienta debe acelerar para alcanzar los puntos, al no considerar este retraso en el tiempo de ejecución de la trayectoria se induce un error que se mantiene durante el desarrollo del experimento. Por otra parte, se pueden ver las diferencias entre el la trayectoria original y la simplificación RDP, en azul, donde el error corresponde a la eliminación de los suavizados de la trayectoria digital.

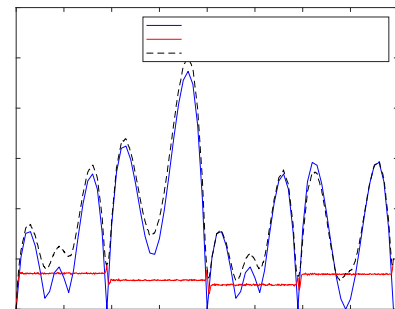


Figura 3: Error absoluto de la trayectoria cuadrada

La Figura 4 muestra la comparativa de la trayectoria circular, de donde la trayectoria creada digitalmente se compone por 80 puntos, que son reducidos a 9 posterior a la simplificación RDP, al igual que la anterior, esta trayectoria cuenta con una tolerancia de 1 cm. El promedio del error absoluto es de 0.3429 cm entre la trayectoria creada digitalmente y la simplificación RDP de la misma, y el error absoluto promedio de la simplificación RDP y la trayectoria ejecutada por el robot es de 0.1502 cm.

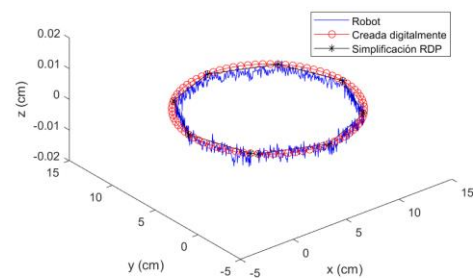


Figura 4: Trayectoria circular a velocidad constante

Por otra parte, en la Figura 5 se puede apreciar la misma trayectoria realizada en el experimento anterior, con la

diferencia de que esta se lleva a cabo a velocidad variable, con una aceleración mayor al inicio y al final del círculo, mientras que en la mitad del camino se baja la velocidad. En este experimento se obtiene un error absoluto promedio de 0.4232 cm entre la trayectoria creada digitalmente y la simplificación de esta, mientras que el error entre la simplificación RDP y la trayectoria ejecutada por el robot es de 0.1377 cm.

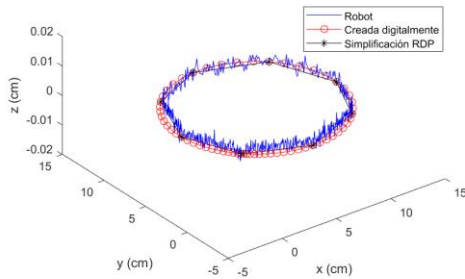


Figura 5: Trayectoria circular a velocidad variable

Si se visualiza el error absoluto en el tiempo (Figura 6), sus valores caen en los momentos en los que se cruzan los puntos de la simplificación, y al igual que se explicó en el experimento del cuadrado, se visualiza un error mantenido en el tiempo de menos de 0.2 cm entre la simplificación RDP y la trayectoria realizada por el robot, que se puede atribuir a la diferencia del tiempo que toma acelerar para llegar a la velocidad deseada.

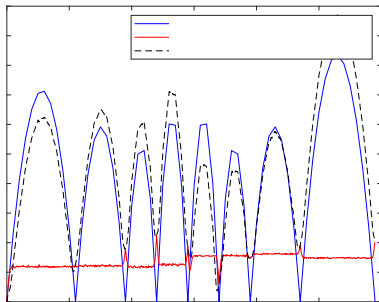


Figura 6: Error absoluto de trayectoria circular a velocidad variable

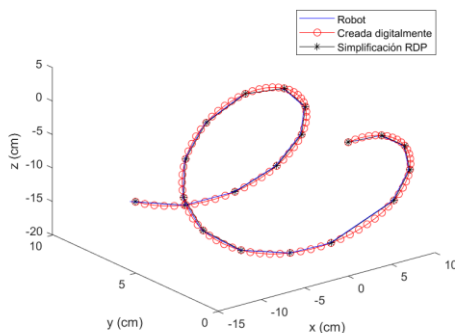


Figura 7: Trayectoria espiral a velocidad constante

En la Figura 7 se muestra la trayectoria realizada en espiral, donde la trayectoria creada digitalmente se compone por un total de 101 puntos, mientras que la simplificación RDP cuenta únicamente con 20 puntos, que al igual que las pruebas anteriores, la simplificación se realiza con una tolerancia de 1

cm. El promedio del error absoluto obtenido entre la trayectoria digital y la trayectoria simplificada es de apenas 0.2354 cm, mientras que los errores obtenidos cuando se comparan estas trayectorias con la ejecutada por el robot superan los 13 cm, aunque los movimientos trazados por el robot se acerquen con bastante precisión a las trayectorias deseadas, se debe tener en cuenta que el error de la trayectoria considera el tiempo de ejecución, así, las acciones de aceleración o desaceleración en lazo abierto del robot pueden ocasionar que se aleje del tiempo de ejecución en que se ha establecido.

3.2. Trayectorias obtenidas con el Optitrack™

Inicialmente se traza una trayectoria lineal de aproximadamente 10 cm de largo (Figura 8), esto se realiza sobre una superficie elástica, por lo que la trayectoria muestra ruido producto tanto de la medida del Optitrack como del pulso del usuario que ejecuta la trayectoria, considerando las mismas características de tolerancia para la simplificación, se obtiene una trayectoria reducida a dos puntos, la cual se pretende ejecutar con el robot; de la comparación entre la trayectoria trazada por el usuario y la simplificación, se obtiene un promedio de error absoluto de 0.6081 cm, mientras que la comparación entre la trayectoria simplificada y la ejecutada por el robot se obtiene un promedio de error de 0.2619 cm.

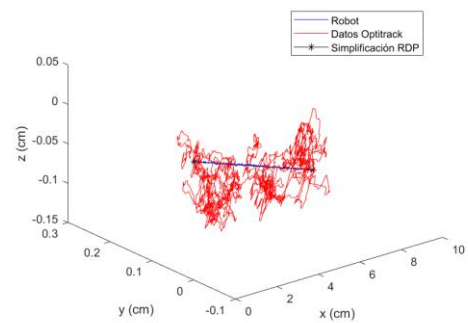


Figura 8: Trayectoria lineal realizada por usuario

La siguiente trayectoria realizada fue un cuadrado, del cual se tomaron 5373 puntos, que al realizar la simplificación se redujeron a 5, en la Figura 9 se observa el suavizado que representa reducir en dicha magnitud la cantidad de puntos, de este experimento se obtuvo un promedio de error absoluto de 0.6294 cm entre los datos obtenidos del Optitrack con respecto a la simplificación, y un promedio de error absoluto de 0.2848 cm entre la trayectoria simplificada con respecto a la ejecutada por el robot.

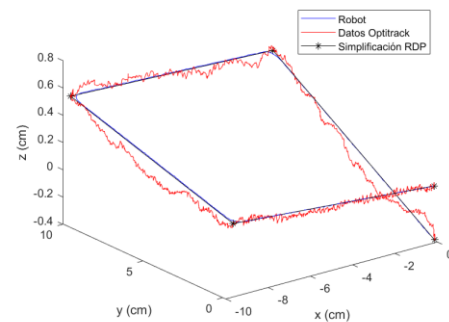


Figura 9: Trayectoria cuadrada realizada por usuario (tolerancia 1 cm)

Asimismo, se decide repetir el experimento anterior, definiendo una tolerancia 10 veces menor a la establecida, con la intención de comparar el desempeño tanto de la simplificación de la trayectoria como la ejecución de esta realizada por el robot. La Figura 10 muestra el experimento con la trayectoria simplificada a menor tolerancia, donde esta cuenta con un total de 33 puntos, de donde se obtiene un promedio error absoluto de 0.2033 cm entre la trayectoria realizada por el usuario con respecto a la trayectoria simplificada y un error de 0.2776 cm entre la trayectoria simplificada y la trayectoria ejecutada por el robot.

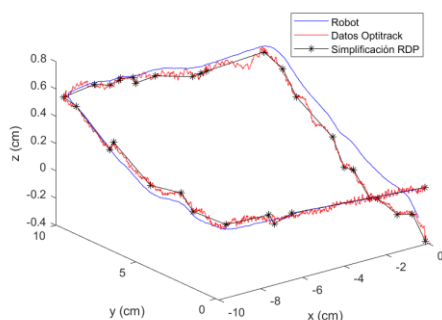


Figura 10: Trayectoria cuadrada realizada por usuario (tolerancia 0.1 cm)

4. Discusión y conclusiones

El presente trabajo pretende brindar una alternativa para la reducción del número de puntos en trayectorias 3D a partir de demostraciones, simplificando una cantidad muy densa de datos por una trayectoria sencilla fácilmente ejecutable, con la que se pierda la menor cantidad de información posible. Se implementa el algoritmo de Douglas-Peucker, con una modificación para que pueda ser utilizado en espacios 3D, y se define un set-up donde a partir de los puntos establecidos se calcula la velocidad y dirección en la que se debe desplazar el efector final del robot UR3e.

Los resultados obtenidos muestran que la simplificación propuesta basada en el algoritmo Douglas-Peucker introduce un error directamente relacionado con la tolerancia que se establece en la simplificación, así, pudiendo variar la cantidad de datos que representan la trayectoria y por ende su error; por otra parte, el error obtenido entre la simplificación RDP y la trayectoria ejecutada por el robot es menor, el cual está relacionado con el ruido de los datos leídos del robot y la precisión que tiene este al ejecutar los movimientos; en esta propuesta se tiene en cuenta el tiempo de ejecución de cada trayectoria, obteniendo resultados diferentes para el mismo recorrido a diferentes velocidades (Figuras 4 y 5). Teniendo en cuenta que en la implementación al robot se realiza un montaje a lazo abierto, no existe realimentación que facilite la reducción del error, es por ello por lo que en los fragmentos de recorrido en los que no existen puntos definidos el error tiende a aumentar, como se puede apreciar en las Figuras 3 y 6. Asimismo, el error presentado entre la trayectoria ejecutada por el robot y la trayectoria original, bien la generada digitalmente o la obtenida a través del Optitrack™, recae principalmente en las diferencias que se generan por la simplificación de la trayectoria.

Es de resaltar que en este artículo se expone un set-up de experimentación, donde se obtienen datos haciendo uso de

diferentes métodos, los cuales pueden ser simplificados e incluidos en la ejecución del robot, siendo conveniente para un espacio de trabajo de 3 dimensiones; pudiendo ser implementada esta metodología en entornos donde las aplicaciones requieren movimientos repetitivos o movimientos que son pre-establecidos.

Esta propuesta pretende abrir un camino a la ejecución de trayectorias quirúrgicas paramétricas simples o estandarizadas, en escenarios que necesiten que un instrumento se desplace de un punto A hacia un punto B por un recorrido establecido, independiente de la forma en la que se defina el recorrido. Es de considerar que en la presente investigación la implementación se realiza en lazo abierto, sin tener ningún tipo de realimentación de la posición del efector final del robot con respecto a la trayectoria deseada, si bien los resultados obtenidos son buenos, con errores absolutos bajos, como trabajo futuro se plantea la implementación de un lazo cerrado que permita un mejor control tanto de la posición del efector final durante el recorrido, como la velocidad y aceleración que debe mantener para minimizar el error en mayor medida. Asimismo, se plantea la posibilidad de que la trayectoria pueda ser adaptativa, considerando que entornos quirúrgicos se pueden presentar movimientos de los elementos presentes en el desarrollo de la trayectoria, es interesante proporcionar una alternativa que actualice la información de la trayectoria constantemente.

Agradecimientos

Este trabajo ha sido financiado parcialmente por la Agencia Nacional de Investigación con el proyecto RACE (ref. PID2019-111023RB-C32).

Referencias

- 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.
- Douglas, D. H. & Paucker, 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.
- 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/tre.834>
- 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.
- 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.
- 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.
- Toker, A., 2014. Robotic thoracic surgery: from the perspectives of European chest surgeons. *Journal of Thoracic Disease*, 6(Suppl 2), p. S211.
- Visvalingam, M. & Whyatt, J. D., 1993. Line generalisation by repeated elimination of points. *The cartographic journal*, 30(1), pp. 46-51.
- 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.
- 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.
- Zhao, Z. & Saalfeld, A., 1997. Linear-time sleeve-fitting polyline simplification algorithms. *Proceedings of AutoCarto*, Volumen 13, pp. 214-223