

Jornadas de Automática

Entorno de simulación para misiones de búsqueda y rescate con enjambres de drones

Roldán-Gómez, J.J.^{a,*}, Sánchez-Alocén, A.^a, Domínguez-Sáez, R.^a, González-Mera, E.^b, Moral-Jaraba, D.^b, Nieto-Illescas, A.^b

^a Departamento de Ingeniería Informática, Universidad Autónoma de Madrid, C/ Francisco Tomás y Valiente, n°11, 28049, Madrid, España.

^b Aurea Avionics, C/ Margarita Salas, n° 24, 28919, Leganés, Madrid, España.

To cite this article: Roldán-Gómez, J.J., Sánchez-Alocén, A., Domínguez-Sáez, R., González-Mera, E., Moral-Jaraba, D., Nieto-Illescas, A. 2024. Simulation environment with virtual scenarios for search and rescue missions with drone swarms. *Jornadas de Automática*, 45. <https://doi.org/10.17979/ja-cea.2024.45.10811>

Resumen

Los enjambres de drones son una tecnología con gran potencial en las misiones de búsqueda y rescate, ya que permiten sobrevolar extensas áreas para detectar potenciales víctimas sin requerir un control directo por parte de los operadores. El proyecto “Swarming Edge Computing sobre 5G” (SWECO-5G) ha investigado en tecnologías de computación distribuida y comunicación en redes 5G. En el marco de este proyecto se ha desarrollado un entorno de simulación basado en el motor de videojuegos Unity, que permite simular las dinámicas de múltiples drones y transmitir sus vídeos con altas resoluciones y frecuencias. Este entorno facilita el desarrollo de algoritmos embebidos en los drones, como es el caso de los algoritmos de enjambre para controlar la misión o los modelos de inteligencia artificial para realizar detecciones en los vídeos, al no depender de realizar vuelos con el enjambre para su validación.

Palabras clave: Robótica, robots aéreos, simulación, aprendizaje automático, robótica de campo, navegación de robots.

Simulation environment with virtual scenarios for search and rescue missions with drone swarms

Abstract

Drone swarms are a technology with great potential in search and rescue missions, given that they allow flying over extensive areas to detect potential victims without requiring direct control by the operators. The “Swarming Edge Computing over 5G” (SWECO-5G) project has researched distributed computing and communication technologies in 5G networks. Within the framework of this project, a simulation environment has been developed based on the Unity video game engine, which allows simulating the dynamics of multiple drones and transmitting their videos with high resolutions and frame rates. This environment facilitates the development of algorithms embedded in drones, such as swarm algorithms to control the mission and artificial intelligence models to perform detections in videos, as they do not depend on carrying out flights with the swarm for validation.

Keywords: Robotics technology, flying robots, simulation, machine learning, field robotics, robot navigation.

1. Introducción

Hace tiempo que los drones se aplican en misiones de búsqueda y rescate en escenarios terrestres o marinos. Esto se debe a su capacidad para sobrevolar rápido escenarios extensos donde han ocurrido desastres naturales o provocados, como terremotos y corrimientos de tierras, huracanes y tormentas, inundaciones y riadas, incendios forestales, derrumbes, naufragios, accidentes industriales o atentados terroristas.

El uso de flotas de drones en lugar de drones individuales permite cubrir más área en menos tiempo, algo deseable en un contexto de desastre en el que el tiempo es un factor crucial para salvar vidas. Además, las flotas heterogéneas compuestas por drones con diferentes características y sensores pueden realizar búsquedas más especializadas combinando información obtenida por diferentes fuentes a diferentes alturas (Roldán-Gómez et al., 2020).

Los enjambres de drones suponen un paso más en esta estrategia, buscando una coordinación entre los drones que

*Autor para correspondencia: juan.roldan@uam.es

permita mejorar la eficiencia de la misión y reducir la carga de trabajo de los operadores. Sin embargo, satisfacer esta necesidad de coordinación requiere afrontar una serie de retos, que van desde las comunicaciones entre los robots hasta la toma de decisiones distribuida (García-Aunon et al., 2021).

Una revisión de los casos de uso de enjambres de drones en contexto de emergencias se puede encontrar en (Hoang et al., 2023), donde se analizan las oportunidades y retos de estos sistemas para operaciones de búsqueda y rescate. Algunos de estos trabajos (Ruetten et al. 2020) se centran en los métodos para organizar el enjambre, mientras que otros (Cardona et al., 2021) en los algoritmos de visión para la detección de víctimas.

El proyecto “Swarming Edge Computing sobre 5G” (SWECO-5G) ha investigado en tecnologías de computación distribuida y comunicación en redes 5G para superar estos retos. Este proyecto ha sido desarrollado por Aurea Avionics en colaboración con la Universidad Autónoma de Madrid (UAM), contando con financiación del Programa UNICO 5G I+D 2022 del Ministerio de Economía, Comercio y Empresa del Gobierno de España.

Este artículo describe los principales desarrollos del proyecto SWECO-5G y está organizado de la siguiente manera: la Sección 2 aborda una visión general del proyecto, la Sección 3 se centra en el entorno de simulación, la Sección 4 muestra las pruebas realizadas y los resultados obtenidos y la Sección 5 resume las conclusiones del proyecto.

2. Arquitectura del proyecto

Como se ha mencionado anteriormente, el proyecto SWECO-5G ha buscado desarrollar tecnologías relacionadas con la computación distribuida y la comunicación en redes 5G. Para ello, ha considerado un enjambre de drones que toman vídeos aéreos de un escenario de desastre y emplean modelos de inteligencia artificial para detectar potenciales víctimas que necesiten ser rescatadas.

La idea de este proyecto ha sido pasar de una investigación aplicada y desarrollo de laboratorio disponibles (TRL3-4) a un prototipo validado en un entorno simulado (TRL6). Por tanto, una parte importante ha consistido en desarrollar un entorno de simulación que pueda proporcionar información realista sobre el escenario y los drones para validar los algoritmos y probar las comunicaciones.

El diagrama de la Figura 1 muestra los componentes principales del proyecto SWECO-5G y las comunicaciones entre ellos. Como se puede observar, el entorno de simulación ocupa un lugar central del esquema, generando los vídeos que deben ser procesados a bordo de los drones.

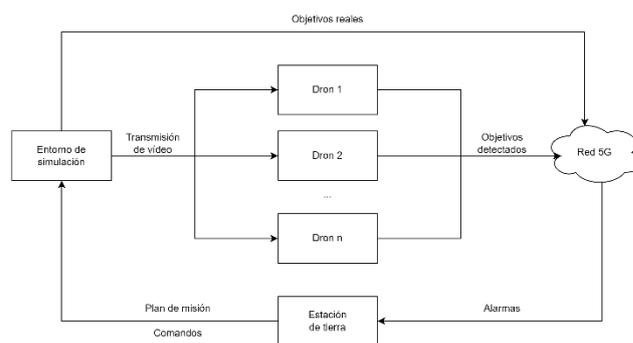


Figura 1: Diagrama simplificado del proyecto SWECO-5G.

Dado que uno de los objetivos del proyecto era transferir fácilmente las tecnologías del entorno de laboratorio a los sistemas reales, el hardware de los drones es independiente y operable tanto en simulaciones como en misiones reales. Cada dron consta de un computador monoplaca (single-board computer, SBC), donde se ejecutan los algoritmos necesarios para controlar el vuelo y las cargas de pago, y un módulo de radio 5G, que le da acceso a una infraestructura de red 5G que le permite comunicarse con el resto de los drones y la estación de tierra.

Esta arquitectura se ha puesto a prueba principalmente con las transmisiones de vídeo. Todos los vídeos son generados en el entorno de simulación, pero cada uno de ellos es enviado al dron correspondiente. En los drones, una red neuronal convolucional se emplea para realizar detecciones de elementos relevantes para la misión, como son el caso de personas, vehículos, etc.

Este diseño requiere una capacidad para simular el escenario y los drones en tiempo real, así como para transmitir de forma fluida vídeos con altas resoluciones como hacen los drones reales. En la Sección 3 se describe con más detalle el simulador desarrollado para satisfacer estos requisitos tan exigentes.

3. Entorno de simulación

Como se ha explicado anteriormente, el proyecto SWECO-5G tiene como centro el entorno de simulación, que permite simular el comportamiento de los drones y sus cargas de pago. En esta sección se va a describir dicho entorno de la siguiente manera: la Sección 3.1 analiza el potencial de los motores de videojuegos en la robótica, la Sección 3.2 describe el diseño del entorno de simulación desarrollado en este proyecto, la Sección 3.3 explica las posibles configuraciones para las simulaciones y, por último, la Sección 3.5 aborda con más detalles la generación y transmisión de vídeo.

3.1. Motores de videojuegos en robótica

Los motores de videojuegos son entornos de desarrollo que disponen de un conjunto de recursos que facilitan la creación de videojuegos. Estos entornos se diferencian de otros con propósito más general en que integran los gráficos (modelos y animaciones en 3D o 2D) y las reglas del juego (programadas mediante scripts o bloques). También suelen disponer de ciertos componentes comunes, como físicas, cámaras, luces,

sonidos o animaciones, que hacen más sencillo y rápido el desarrollo de nuevos juegos. Algunos motores de videojuegos que han trascendido su ámbito original y se emplean para desarrollar otras aplicaciones como simuladores, aplicaciones de realidad virtual y aumentada o interfaces son Unity y Unreal.

En el terreno de la robótica, estos motores de videojuegos se han erigido como una alternativa a simuladores tradicionales como Gazebo, Webots o CoppeliaSim (De Melo et al., 2019). En particular, Unity ha mostrado su potencial para la simulación de robots (Konrad, 2019), la recreación de los entornos de las misiones (Roldán-Gómez et al., 2022), la generación de datos sintéticos para visión por computador (Borkman et al., 2021), el entrenamiento de modelos de inteligencia artificial (Juliani et al., 2018) y el desarrollo de interfaces inmersivas (Roldán-Gómez, 2018).

Unity es un entorno flexible para el desarrollo de estas aplicaciones robóticas, ya que permite a los desarrolladores ajustar el nivel de detalle necesario en cuanto a gráficos, físicas... Por ejemplo, un simulador de robots puede emplear modelos cinemáticos y dinámicos muy precisos, mientras que un simulador de misiones puede integrar modelos más sencillos para reducir costes de ejecución. Además, hay multitud de recursos disponibles en su propia tienda (Asset Store) y repositorios externos que facilitan el desarrollo de escenarios, la integración de realidad virtual y aumentada (Meta, 2024) o la aplicación de algoritmos de aprendizaje (MLAgents, 2024).

Por todas estas razones se decidió desarrollar el entorno de simulación del proyecto SWECO-5G utilizando Unity. Este simulador se explica con mayor profundidad en la siguiente sección.

3.2. Diseño del entorno de simulación

El entorno de simulación se desarrolló teniendo en cuenta los siguientes requisitos:

- El escenario virtual reproduce una misión de búsqueda y rescate de un grupo de personas y vehículos.
- La flota tiene de 1 a 5 drones, definidos por su ID, tipo (ala fija o multi-rotor), velocidad máxima y carga de pago.
- Las cargas de pago son una cámara diurna (RGB) y otra nocturna (IR) con movimientos PTZ (pan, tilt, zoom).
- La simulación se configura eligiendo el escenario, el área de búsqueda, el número de objetivos y el número de drones.
- Hay dos modos de simulación: el interno en el que el simulador reproduce la dinámica de los drones y el externo en el que la recibe desde el exterior.
- En el modo interno, el área de búsqueda se divide para asignar una subárea a cada dron y los drones generan una lista de puntos para barrerla.
- En el modo externo, el simulador recibe la telemetría de los drones y representa sus movimientos en el escenario.
- El simulador debe exportar los vídeos tomados por las cámaras de todos los drones.

- Cuando un dron realice una detección, debe parar su ruta y sobrevolar el objetivo.

El diagrama de la Figura 2 muestra la arquitectura desarrollada para cumplir con estos requisitos. Como se puede apreciar, los drones pueden recibir el plan de misión desde la estación de tierra, creando sus propias rutas para cubrir el área de misión, o directamente recibir las posiciones desde un simulador externo. Por su parte, el entorno virtual se encarga de generar los vídeos teniendo en cuenta las posiciones de los drones y las orientaciones de sus cargas de pago.

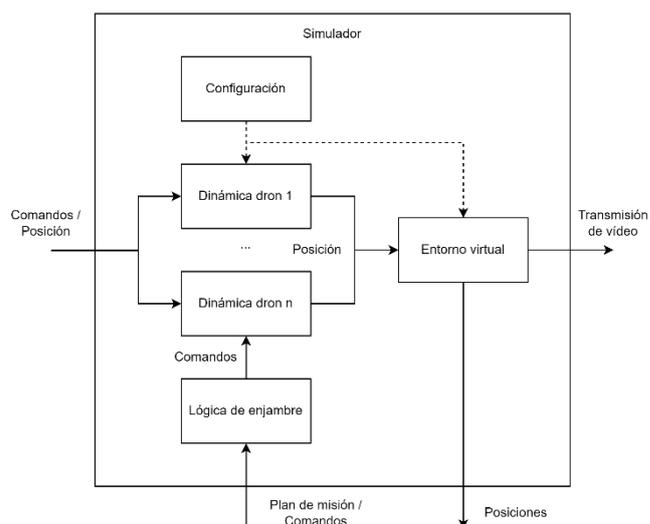


Figura 2: Diagrama simplificado del entorno de simulación.

3.3. Configuración de la simulación

La configuración de la simulación se realiza a través de un conjunto de menús. El primero de ellos se muestra en la Figura 3, que permite configurar la simulación para generar la dinámica de los drones o recibirla desde un programa externo a través un socket UDP. En el primer caso, el simulador esperará el plan de la misión enviado desde la estación de tierra mediante un socket TCP para comenzar su ejecución. En el segundo, también esperará la telemetría generada por el programa externo para comenzar la misión.



Figura 3: Menú principal.

El resto de la configuración de la simulación se puede configurar con los menús de opciones principales (Figura 4) y de los drones (Figura 5). En el primero se puede seleccionar el escenario (por el momento, terrestre o marino), el número de drones, el número de personas buscadas, el número de

iteraciones para la misión y las IPs y puertos para comunicarse con la estación de tierra. En el segundo se ajustan las características de los drones y sus cargas de pago: IPs y puertos para establecer la comunicación, velocidad máxima de los drones, pan, tilt y zoom por defecto para las cámaras, resolución y frecuencia requeridas para los vídeos y sistema empleado para generar las imágenes (CPU o GPU).



Figura 4: Menú de opciones principales.



Figura 5: Menú de opciones de los drones.

Una vez configurada y lanzada la misión, los drones comenzarán a barrer las áreas asignadas y exportar sus vídeos del escenario en tiempo real. Para llevar a cabo esta tarea de forma eficiente han sido necesarios los recursos indicados en la siguiente sección.

3.4. Transmisión de vídeo

Unity contiene multitud de recursos para capturar vídeos de las escenas de los videojuegos. El objeto cámara es muy flexible porque debe funcionar en videojuegos en 2D y 3D tanto en primera como en tercera persona. Para ello permite seleccionar el tipo de proyección (perspectiva u ortográfica), así como parámetros como el campo de visión, la distancia focal, los planos límite cercano y lejano, etc. Basta con situar una cámara en cada dron y ajustar estos parámetros para cumplir ese requisito.

Por defecto, la cámara de Unity captura imágenes visibles y no puede configurarse para generar imágenes térmicas. Para ello hay que configurar el proyecto como SRP (Scriptable Render Pipeline) (SRP, 2024), lo que permite usar scripts de C# para controlar el renderizado de los gráficos, e importar el paquete Post Processing (PPS, 2024), que habilita el procesado de los gráficos después de que la cámara capture la escena pero antes de que esta se visualice en pantalla. Esta combinación permite desarrollar shaders que dependan de variables de la escena (por ejemplo, la temperatura) y aplicarlos a ciertos objetos (un modelo de persona) o materiales usados por varios objetos (un material para todos los humanos). También se pueden usar capas para

elegir qué objetos son visibles y cuáles no por la cámara, ahorrando el tener que asignar un shader dependiente de la temperatura a los objetos de fondo.

La exportación de las imágenes generadas en Unity a otros programas situados en otros equipos tiene cierta dificultad. Aunque hay varios paquetes disponibles para realizar esta tarea en Unity como RenderStreaming (RS, 2024) y WebRTC (WRTC, 2024), su rendimiento no es suficiente cuando se exportan vídeos de varias fuentes (5 drones) con resoluciones altas (1080p) y frecuencias altas (30 Hz).

Estos problemas se pueden evitar empleando FFMpeg (FFmpeg, 2024), una colección de programas libres que permiten grabar, codificar y transmitir vídeo y audio. Este programa permite transmitir los vídeos requeridos con un rendimiento adecuado, pero por sí solo no puede acceder a las cámaras de Unity. Para ello hay que emplear el paquete FFMpegOut (FFmpegOut, 2024), que accede a las imágenes de las cámaras como texturas, las codifica de la manera elegida por el usuario y las envía fuera de Unity a FFMpeg. A pesar de la complejidad de la transmisión, que exige el paso de los vídeos por dos programas intermediarios, el rendimiento se mantiene.

4. Pruebas y resultados

El simulador fue probado con un conjunto de misiones de búsqueda y rescate, en las que se colocaron a varias personas perdidas en ubicaciones aleatorias del mapa y se utilizaron los vídeos de los drones para detectarlas con modelos de inteligencia artificial (ver flujograma de Figura 6).

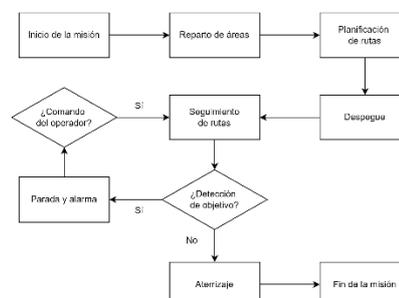


Figura 6: Flujograma de la misión.

La Figura 7 muestra una vista del mapa terrestre donde se puede apreciar el gran nivel de detalle en el relieve y la vegetación, incluyendo modelos muy diversos y realistas de árboles, arbustos o rocas. En esta figura también se puede apreciar una vista cenital en miniatura del escenario, que muestra sus áreas diferenciadas con montañas, valles, ríos o lagos.



Figura 7: Escenario de la misión.

Por su parte, la Figura 8 muestra una imagen tomada por un dron en el simulador y exportada hasta su correspondiente computador, donde una red neuronal ha realizado la detección de una persona. Una vez realizada esta detección, el algoritmo envía una alarma al simulador para detener el dron y permitir a una persona confirmarla o rechazarla.



Figura 8: Detección en el vídeo.

5. Conclusiones

Este trabajo presenta una herramienta de simulación desarrollada para reproducir misiones de búsqueda y rescate con una flota de drones. La generación de vídeos sintéticos de gran realismo y su transmisión con características reales facilita el desarrollo y validación de algoritmos en esta plataforma para después transferirlos a los sistemas reales.

En trabajos futuros se profundizará en aspectos como la lógica de enjambre y el procesamiento de los vídeos. Ahora mismo, la lógica de enjambre es capaz de planificar la misión y distribuir las áreas de forma eficiente, pero no de modificar el plan para adaptarse a los imprevistos como detecciones o averías. Por su parte, el entrenamiento de redes neuronales convolucionales con imágenes sintéticas puede permitir emplear estos modelos en contextos donde no hay suficientes imágenes reales.

Agradecimientos

El proyecto SWECO-5G (Swarming Edge Computing sobre 5G) se realiza en el marco del Programa ÚNICO I+D 6G de 2022, del Ministerio de Asuntos Económicos y Transformación Digital, con número de expediente TSI-064200-2022-5 y está financiado por la Unión Europea – NextGenerationEU.

Referencias

- Borkman, S. et al., 2021. Unity perception: Generate synthetic data for computer vision. arXiv preprint arXiv:2107.04259.
- Cardona, G. A. et al., 2021. Visual victim detection and quadrotor-swarm coordination control in search and rescue environment. *International Journal of Electrical and Computer Engineering*, 11(3), 2079.
- De Melo, M. S. P. et al., 2019. Analysis and comparison of robotics 3d simulators. In 2019 21st Symposium on Virtual and Augmented Reality (SVR) (pp. 242-251). IEEE.
- FFmpeg, 2024. FFMPEG Project. <https://ffmpeg.org/> (último acceso: 17/05/2024)
- FFmpegOut, 2024. FFMpegOut Project. <https://github.com/keijiro/FFmpegOut> (último acceso: 17/05/2024)
- García-Aunon, P., et al., 2021. Practical applications using multi-UAV systems and aerial robotic swarms. *REVISTA IBEROAMERICANA DE AUTOMÁTICA E INFORMÁTICA INDUSTRIAL*, 18(3), 230-241.
- Hoang, M. T. O. et al., 2023. Drone swarms to support search and rescue operations: Opportunities and challenges. *Cultural Robotics: Social Robots and Their Emergent Cultural Ecologies*, 163-176.
- Juliani, A. et al., 2018. Unity: A general platform for intelligent agents. arXiv preprint arXiv:1809.02627.
- Konrad, A., 2019. Simulation of mobile robots with unity and ros: A case-study and a comparison with gazebo.
- Meta, 2024. Meta XR All-in-One SDK <https://assetstore.unity.com/packages/tools/integration/meta-xr-all-in-one-sdk-269657> (último acceso: 17/05/2024)
- MLAgents, 2024. Unity Machine Learning Agents. <https://unity.com/products/machine-learning-agents> (último acceso: 17/05/2024) y <https://github.com/Unity-Technologies/ml-agents> (último acceso: 17/05/2024)
- PPS, 2024. Unity Post Processing Stack. <https://docs.unity3d.com/Packages/com.unity.postprocessing@3.2/manual/index.html> (último acceso: 17/05/2024)
- Roldán Gómez, J. J., 2018. Adaptive and immersive interfaces to improve situational awareness in multi-robot missions, Doctoral dissertation, Universidad Politécnica de Madrid.
- Roldan-Gomez, J. J., et al., 2020. A review on multi-robot systems: current challenges for operators and new developments of interfaces. *Revista Iberoamericana de Automática e Informática Industrial*, 17(3), 294-305.
- Roldán-Gómez, J. J. et al., 2022. SwarmCity project: monitoring traffic, pedestrians, climate, and pollution with an aerial robotic swarm: Data collection and fusion in a smart city, and its representation using virtual reality. *Personal and Ubiquitous Computing*, 26(4), 1151-1167.
- RS, 2024. Unity Render Streaming. <https://docs.unity3d.com/Packages/com.unity.renderstreaming@3.1/manual/index.html> (último acceso: 17/05/2024)
- Ruetten, L. et al., 2020. Area-optimized UAV swarm network for search and rescue operations. In 2020 10th annual computing and communication workshop and conference (CCWC) (pp. 0613-0618). IEEE.
- SRP, 2024. Unity Scriptable Render Pipeline. <https://unity.com/srp> (último acceso: 17/05/2024)
- WRTC, 2024. WebRTC for Unity. <https://docs.unity3d.com/Packages/com.unity.webrtc@3.0/manual/index.html> (último acceso: 17/05/2024)