

Jornadas de Automática

Cobertura autónoma con enjambres UAV mediante asignación óptima de tareas

Manuel Domínguez-Montero^a, Víctor Rivera^a, Raul Tapia^{a,*}, José Ramiro Martínez-de Dios^a, Aníbal Ollero^a

^aGRVC Robotics Laboratory, Universidad de Sevilla, Sevilla, España

To cite this article: M. Domínguez-Montero, V. Rivera, R. Tapia, J.R. Martínez-de Dios, A. Ollero. 2025. Autonomous Coverage with UAV Swarms using Optimal Task Assignment. *Jornadas de Automática*, 46. <https://doi.org/10.17979/ja-cea.2025.46.12090>

Resumen

Este trabajo presenta un sistema cooperativo de percepción-planificación-navegación que permite a enjambres de vehículos aéreos no tripulados (UAVs) cubrir de forma autónoma un conjunto de posiciones objetivo representadas mediante marcadores visuales. El sistema de percepción fusiona observaciones distribuidas para estimar con mayor precisión la localización de cada tarea, mientras que el planificador resuelve un problema de programación lineal para obtener la asignación óptima UAV-tarea. Finalmente, un módulo de control off-board multivehículo genera las trayectorias y ejecuta las operaciones de armado, despegue, seguimiento y aterrizaje. La validación experimental se ha realizado en simulación Software-In-The-Loop (SITL) con PX4 y Gazebo. Los resultados obtenidos corroboran que la cooperación de UAVs mejora significativamente la eficiencia y robustez de las misiones de cobertura aérea.

Palabras clave: Sistemas multiagente, Coordinación de sistemas multivehículo, Asignación de tareas y diseño de trabajo, Robótica aérea, Vehículos autónomos

Autonomous Coverage with UAV Swarms using Optimal Task Assignment

Abstract

This work presents a cooperative perception-planning-navigation system that enables swarms of unmanned aerial vehicles (UAVs) to autonomously cover a set of target positions represented by visual markers. The perception system fuses distributed observations to more accurately estimate the location of each task, while the planner solves a linear programming problem to obtain the optimal UAV-task assignment. Finally, a multi-vehicle off-board control module generates the trajectories and executes the arming, takeoff, tracking, and landing operations. Experimental validation was carried out in a Software-In-The-Loop (SITL) simulation using PX4 and Gazebo. The results confirm that UAV cooperation significantly improves the efficiency and robustness of aerial coverage missions.

Keywords: Multi-agent systems, Coordination of multiple vehicle systems, Task allocation-sharing and job design, Aerial robotics, Autonomous vehicles

1. Introducción

El uso de robots aéreos ha demostrado ser de gran utilidad en una amplia variedad de aplicaciones. En particular, los sistemas aéreos no tripulados (*unmanned aerial systems*, UAS) despiertan especial interés para llevar a cabo diversas tareas, que van desde la inspección y el mantenimiento (Tapia et al. (2019); Caballero et al. (2024); Tapia et al. (2024)) hasta la vi-

gilancia (Martínez-de Dios et al. (2020); Gañán et al. (2022)), entre otras. Además, el uso de enjambres de vehículos aéreos (*o swarm aerial robotics*) permite distribuir la carga de trabajo entre múltiples robots, incrementando la cobertura y reduciendo el tiempo total de misión Chung et al. (2018). Los sistemas de percepción y navegación cooperativos propios de los enjambres dotan al sistema de una mayor robustez y capacidad de adaptación frente a fallos individuales o cambios inesperados.

* Autor para correspondencia: raultapia@us.es
Attribution-NonCommercial-ShareAlike 4.0 International (CC BY-NC-SA 4.0)

dos en el entorno Hou et al. (2020). Sin embargo, el diseño e implementación de técnicas para percepción, planificación, navegación o control de enjambres presenta retos de mayor complejidad en comparación con soluciones no cooperativas.

En los últimos años, la robótica aérea de enjambres ha alcanzado un alto nivel tecnológico, con aplicaciones en ámbitos tan diversos como la agricultura de precisión, las operaciones de búsqueda y rescate, la logística de entrega, la seguridad pública, etc (Zhou et al. (2020); Javed et al. (2024)). La coordinación autónoma de varios robots proporciona múltiples ventajas frente al uso de una única plataforma, ya que un enjambre puede cubrir áreas más extensas de forma simultánea y cumplir los objetivos con mayor eficiencia Choi and Lee (2023). Además, la operación en enjambre aporta resistencia y robustez: la flota cooperativa logra recopilar datos de manera más precisa o redundante, manteniendo la misión aun ante fallos de agentes individuales Chen et al. (2020). Estos y otros muchos avances reflejan el estado actual de la robótica de enjambres, evidenciando que la cooperación de múltiples vehículos mejora la cobertura y rendimiento de tareas aplicadas en comparación con plataformas aisladas.

Este artículo presenta un sistema de navegación para enjambres de robots aéreos de cualquier dimensión que permite la detección automática de un conjunto de puntos 3D objetivo y resuelve el problema de asignación para que los puntos objetivo sean cubiertos por los agentes del enjambre. El método hace uso del algoritmo de Kuhn-Munkres (Kuhn (1955); Munkres (1957)) para minimizar el coste en un problema de optimización basado en programación lineal entera. El método ha sido implementado y completamente integrado con un simulador específicamente diseñado para su evaluación.

2. Descripción del método

El objetivo es diseñar un sistema cooperativo capaz de detectar, asignar y cubrir de manera autónoma posiciones objetivo (tareas) mediante un enjambre de vehículos aéreos no tripulados (agentes). Cada tarea será codificada mediante un marcador visual colocado en el escenario de la misión y tendrá asociada una posición 3D y una prioridad. El método propuesto en este trabajo se estructura en tres módulos fuertemente acoplados que conforman un esquema de percepción-planificación-navegación:

- **Percepción cooperativa:** A partir de las imágenes captadas por los UAVs, un detector de marcadores distribuye en red la información de las tareas detectadas (cantidad, posición y prioridad). Esta información es utilizada por el sistema planificador.
- **Planificación para asignación:** Con las posiciones objetivo y el estado actual de cada vehículo, el planificador resuelve un problema de asignación que minimiza el tiempo total de la misión. El resultado es un mapa agente-tarea utilizado por el sistema de navegación.
- **Control off-board multi-UAV:** Una vez resuelta la asignación, se generan las trayectorias para que cada UAV alcance su objetivo con seguridad.

Además, el método propuesto es capaz de operar de forma inherente y sin coste computacional adicional con casos de uso en los que el número de vehículos aéreos y el número de tareas a realizar no coincidan. Esto se detalla con mayor profundidad en la Sección 2.2.

2.1. Percepción cooperativa

El sistema de percepción propuesto se basa en la detección cooperativa de marcadores fiduciales ArUco a partir de las imágenes captadas a bordo de los vehículos aéreos. La elección de este tipo de marcadores obedece a varias razones prácticas: (i) cada marcador codifica un ID inequívoco, (ii) proporcionan robustez ante occlusiones parciales y cambios de iluminación (más aún en un enfoque cooperativo) y (iii) el coste computacional de la detección es bajo.

Cada robot aéreo incorpora una cámara que se supone calibrada (parámetros intrínsecos, parámetros de distorsión y matriz de transformación cámara-robot ${}^R\mathbf{T}_C$). El sistema de detección opera de la siguiente forma:

1. Identificación de los marcadores en la imagen usando un algoritmo de detección de ArUcos (Garrido-Jurado et al. (2014)).
2. Estimación de la posición y orientación relativa marcador-cámara ${}^C\mathbf{T}_M$ resolviendo un problema PnP (Lepetit et al. (2009)).
3. Conversión a la referencia del robot: ${}^R\mathbf{T}_M = {}^R\mathbf{T}_C \cdot {}^C\mathbf{T}_M$.
4. Conversión a coordenadas globales empleando la posición y orientación del UAV: ${}^W\mathbf{T}_M = {}^W\mathbf{T}_R \cdot {}^R\mathbf{T}_M$.

Por tanto, para cada marcador k detectado por el robot i , existe una posición 3D $\mathbf{p}_k^{(i)}$. Se omite la orientación por sencillez, ya que el objetivo es que los robots cubran un determinado número de posiciones en el espacio. Todos los robots publican sus observaciones $\mathbf{p}_k^{(i)}$ a un proceso central que las fusiona de la siguiente forma:

$$\mathbf{p}_k = \frac{1}{|\mathcal{R}_k|} \sum_{i \in \mathcal{R}_k} \mathbf{p}_k^{(i)}, \quad (1)$$

donde \mathcal{R}_k es el conjunto de los robots (sus índices i) que han detectado el marcador k -ésimo y $|\mathcal{R}_k|$ es la cardinalidad de dicho conjunto. Por tanto, el resultado final del sistema de detección y fusión es un conjunto de tuplas (\mathbf{p}_k, ID_k) representando tareas (posiciones a cubrir por los robots).

Las ventajas de nuestro sistema de detección cooperativo son: (i) cobertura redundante (los marcadores son incorporados aunque sean vistos por un solo robot del enjambre), (ii) precisión mejorada al fusionar múltiples observaciones y (iii) buena escalabilidad (añadir y quitar robots del enjambre no requiere modificar el sistema de percepción). Además es importante destacar que, aunque se utilizan marcadores ArUco para simplificar el problema, se podría emplear cualquier detector capaz de identificar las tareas a realizar. El despliegue de marcadores puede dificultar la misión en ciertos escenarios (por ejemplo, si hay zonas de difícil acceso para colocarlos). En estos casos, el uso de sistemas de percepción alternativos como los basados en redes neuronales artificiales se plantea como una solución factible.

2.2. Planificación para asignación

El módulo de planificación del esquema propuesto en este trabajo tiene como objetivo realizar la asignación óptima entre los vehículos aéreos disponibles y las tareas (posiciones a cubrir) detectadas por el sistema de percepción (Laporte (2007)). Sea $\mathcal{A} = a_1, \dots, a_m$ el conjunto de agentes (vehículos aéreos) y $\mathcal{T} = \tau_1, \dots, \tau_n$ el conjunto de tareas (posiciones 3D objetivo detectadas), definimos $C \in \mathbb{R}^{m \times n}$ como la matriz de costes, donde cada elemento c_{ij} cuantifica el coste que supondría para el agente a_i cubrir la tarea τ_j . Este coste puede incorporar la cantidad de energía almacenada en las baterías de los robots aéreos, el tiempo de vuelo estimado según un modelo dinámico, la distancia a recorrer o la presencia de obstáculos en la ruta (entre muchos otros factores). Por simplicidad, y sin pérdida de generalización, usaremos como coste el tiempo estimado $c_{ij} = t_{ij}$, calculado como:

$$t_{ij} = \frac{\|\mathbf{p}_i - \mathbf{p}_j\|_2}{\bar{v}_i}, \quad (2)$$

donde \mathbf{p}_i es la posición del i -ésimo robot (agente), \mathbf{p}_j es el j -ésimo punto objetivo (tarea) y \bar{v}_i es la celeridad media estimada del i -ésimo robot. Dadas las variables de decisión binarias x_{ij} (con valor 1 si el agente a_i se asigna a la tarea τ_j y 0 en cualquier otro caso), el problema de asignación se formula como un problema de programación lineal entera:

$$\min_{x_{ij}} \sum_{i=1}^m \sum_{j=1}^m c_{ij} x_{ij} \quad (3)$$

$$\text{s.t.} \quad \sum_{j=1}^m x_{ij} = 1, \quad \forall i \in 1, \dots, m, \quad (4)$$

$$\sum_{i=1}^m x_{ij} = 1, \quad \forall j \in 1, \dots, m, \quad (5)$$

$$x_{ij} \in \{0, 1\}, \quad \forall i, j. \quad (6)$$

La función objetivo (3) minimiza el coste total de la asignación y las restricciones (4) y (5) garantizan que cada agente cubra sólo una tarea y cada tarea sea atendida sólo por un agente.

- **Caso asignación perfecta ($m = n$):** Cuando el número de agentes coincide con el de tareas, el problema se reduce a encontrar una asignación óptima en un grafo bipartito completo. La solución se implementa usando el algoritmo de Kuhn-Munkres (Kuhn (1955); Munkres (1957)).
- **Caso agentes ociosos ($m > n$):** Los agentes no asignados (ociosos) deben ejecutar un comportamiento de *stand-by* (por ejemplo, *hovering* o *landing*). Se implementa introduciendo $m - n$ tareas virtuales de coste 0 y ejecutando la implementación utilizada para el caso de *asignación perfecta*.
- **Caso tareas prioritarias ($m < n$):** Se seleccionan las m tareas más prioritarias. Cada marcador puede tener asociada una prioridad conocida a priori. Por sencillez, y sin pérdida de generalidad, se asume que *IDs* mayores corresponden a mayor prioridad. Una vez seleccionadas, este caso se comporta e implementa como un caso de *asignación perfecta*.

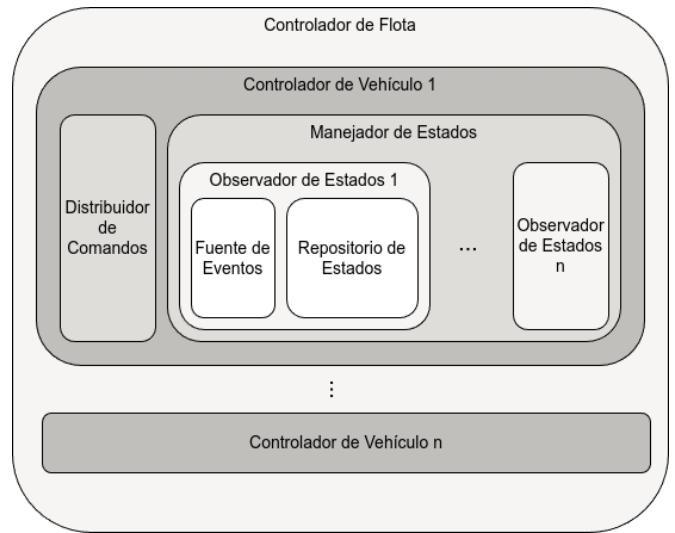


Figura 1: Esquema de la arquitectura software que implementa el control *off-board* multi-UAV.

2.3. Control off-board multi-UAV

El módulo de control *off-board* propuesto está específicamente diseñado para comandar enjambres de UAVs. La arquitectura propuesta pretende desacoplar la lógica de control (dependiente del tipo de controlador de vuelo), facilitando integrar cualquier autopiloto. Además, su diseño modular ofrece la escalabilidad necesaria para adaptarse a distintos tamaños de enjambres y misiones. La clara separación entre los niveles de control y comunicación pretende incrementar la mantenibilidad del sistema. La arquitectura de componentes de software del módulo de control *off-board* se presenta en esquema de bloques de la Figura 1.

- **Controlador de Flota:** Ofrece funcionalidades para las siguientes operaciones: armado del enjambre, despegue, seguimiento de trayectorias, aterrizaje y vuelta a casa (*return-to-home*). El componente **Controlador de Flota** gestiona estas operaciones enviando órdenes a cada **Controlador de Vehículo** (uno por cada UAV en el enjambre).
- **Controlador de Vehículo:** Implementa las funcionalidades de armado, despegue, seguimiento de trayectorias, aterrizaje y vuelta a casa (*return-to-home*) específicos para cada controlador de vuelo. Para ello, genera órdenes a través del **Distribuidor de Comandos**, encargado de enviar los comandos de bajo nivel al autopiloto mediante un esquema cliente-servidor.
- **Manejador de Estados:** Gestiona estados de los UAVs (posición, orientación, estado de armado, etc.). Ofrece la información referente a los estados y la actualiza a partir de los datos proporcionados por el controlador de vuelo. Los datos son recibidos a través de una **Fuente de Eventos** que se adapta a la comunicación empleada por cada UAV (permitiendo atender fuentes de diversas como ROS, ROS2, gRPC, etc.) y, tras ser mapeados y unificados por un **Observador de Estados**, son almacenados en un **Repositorio de Estados**.

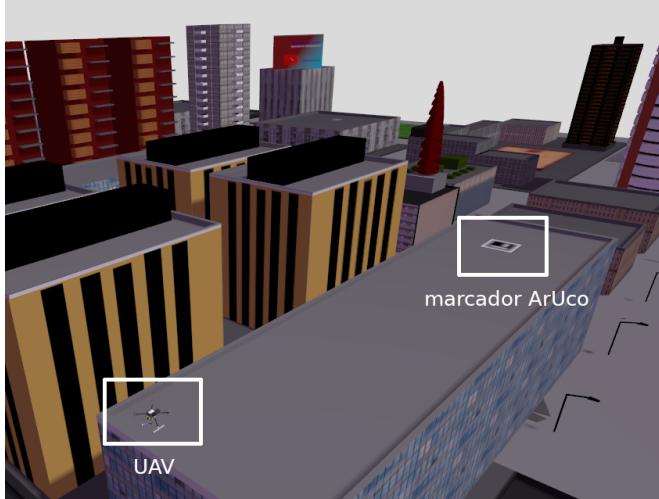


Figura 2: Escenario de Gazebo utilizado para la validación experimental del método.

3. Evaluación experimental

Los entornos operativos de UAVs, inherentemente complejos y dinámicos, presentan desafíos importantes en su diseño e implementación. Las simulaciones proporcionan un entorno controlado, seguro y reproducible para probar y validar algoritmos, estrategias de control y comportamientos de UAVs antes de desplegarlos en situaciones reales. Para la validación experimental del método propuesto se utilizó una simulación *Software In The Loop* (SITL) usando controladores de vuelo basados en el firmware PX4¹ dentro del simulador Gazebo² (ver escenario empleado en la Figura 2). PX4 es un sistema de piloto automático de código abierto, modular y de alta fiabilidad para vehículos no tripulados. El escenario presentaba un entorno urbano de unas 35 hectáreas, los marcadores ArUco fueron situados en los techos de varios edificios con alturas comprendidas entre 10 y 90 m. Los UAVs eran comandados a través de ROS2 Humble. Este método ejecuta el software de control del UAV, incluyendo firmware del autopiloto y lógica de navegación, directamente en un entorno simulado, eliminando la necesidad de hardware físico. PX4 se comunica con Gazebo para intercambiar datos de sensores y comandos de actuadores.

La implementación del método se llevó a cabo usando la librería OpenCV para la detección de los marcadores y la librería SciPy para la resolución del problema de programación lineal. Todos los experimentos fueron ejecutados en un ordenador con procesador Intel i7-13th Gen y 32 GB de RAM. La evaluación experimental del método ha sido dividida en dos partes para mayor detalle: i) validación del sistema de percepción cooperativo descrito en la Sección 2.1 y ii) validación del sistema de planificación para la asignación de tareas presentado en la Sección 2.2. En ambas partes los UAVs utilizan el control *off-board* descrito en la Sección 2.3.

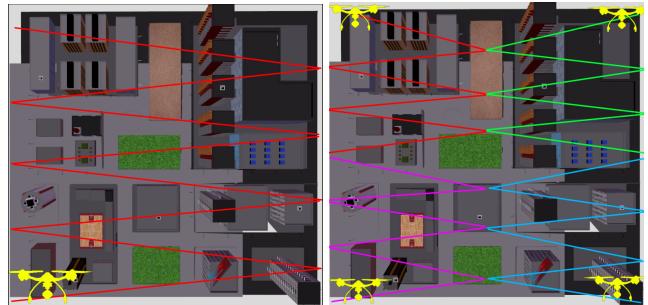


Figura 3: Trayectorias de cobertura planificadas en el escenario de simulación: (izquierda) estrategia sin cooperación; (derecha) estrategia cooperativa.

3.1. Evaluación del sistema de percepción

Para evaluar el impacto del sistema de percepción cooperativa se diseñó un experimento con cuatro multirrotores, cada uno equipado con cámara RGB (resolución 1240×720), y cinco marcadores ArUco (300×300 cm). Para detectar los marcadores, se utilizó el método de la Sección 2.1 durante una misión de cobertura a altura y velocidad constantes implementada con el control *off-board* de la Sección 1. Se compararon dos estrategias de cobertura: i) sin cooperación, donde cada vehículo recorría la malla completa de forma secuencial (véase Figura 3-izquierda) y ii) cooperativa, en la que la misma trayectoria se dividía en cuatro regiones y cada UAV cubría su sector (véase Figura 3-derecha). Se midió el tiempo de misión (desde el despegue simultáneo hasta el final de la cobertura). Los resultados obtenidos se presentan en la Tabla 1, obteniéndose una mejora del 79.68 % con la estrategia cooperativa. El beneficio se atribuye a la cobertura paralela, que elimina recorridos redundantes, y a la fusión de observaciones múltiples, que incrementa la confianza en la localización de cada marcador sin comprometer la robustez del sistema.

	No cooperativo	Cooperativo	Diferencia
Vuelo 1	824.16 s	178.27 s	645.89 s
Vuelo 2	987.77 s	205.72 s	782.05 s
Vuelo 3	956.69 s	178.57 s	778.12 s
Media	922.87 s	187.52 s	735.35 s

Tabla 1: Resultados de evaluación del sistema de percepción de marcadores.

3.2. Evaluación del sistema de planificación

Para evaluar el sistema de planificación se realizaron varios vuelos en simulación con cuatro multirrotores (que iniciaron simultáneamente desde los puntos medios de los cuatro cuadrantes definidos en la Figura 3-derecha tras la etapa de percepción) y un número variable de tareas (marcadores detectados), de modo que se cubrieran los tres casos de interés ($m > n$, $m = n$ y $m < n$; con $m = 4$ agentes y $n = \{3, 4, 5\}$ tareas). En cada vuelo se compararon dos estrategias: asignación aleatoria de agentes a tareas y asignación óptima propuesta en la Sección 2.2. El tiempo total de misión se midió desde el fin de la planificación (comienzo de las tareas) hasta el aterrizaje del último UAV. La Figura 4 muestra los tiempos de misión para los tres casos contemplados ($m = n$ en rojo, $m < n$ en verde y $m > n$ en azul). La asignación óptima redujo el tiempo

¹<https://px4.io/>

²<https://gazebosim.org/>

medio de misión en un 51.86 % respecto a la aleatoria. Estos resultados confirman que el esquema de asignación propuesto mejora de forma sustancial la eficiencia global de la misión.

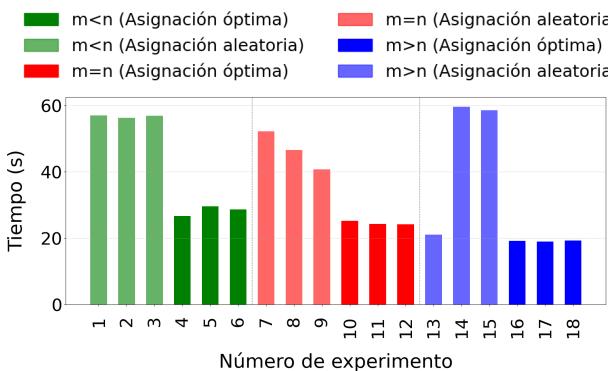


Figura 4: Tiempo total de misión para los tres escenarios de asignación comparando la estrategia aleatoria con la propuesta óptima para los casos $m < n$ (verde), $m = n$ (rojo) y $m > n$ (azul); donde m denota el número de agentes y n denota el número de tareas.

4. Conclusiones

El trabajo presentado en este artículo propone un método de percepción-planificación cooperativo para una asignación de tareas. La navegación de los robots aéreos se realizó usando un novedoso control *off-board* específicamente diseñado para enjambres de UAVs. El método utiliza marcadores visuales para representar tareas y un problema de programación lineal para obtener una asignación óptima. Los sistemas de visión cooperativa, planificación para asignación de tareas y control *off-board* han sido validado experimentalmente con vuelos simulados con *Software-In-The-Loop* (SITL) con PX4 y Gazebo, obteniéndose resultados que muestran cómo los tiempos de misión se reducen significativamente respecto a soluciones no cooperativas. Estos resultados confirman la robustez y viabilidad de la solución para operaciones reales de cobertura aérea autónoma usando enjambres. El trabajo futuro incluye la ejecución en entornos reales y dinámicos, extendiendo el método a enjambres heterogéneos y a un planificador distribuido para eliminar la dependencia de un nodo central. También es parte del trabajo futuro extender la comparativa del método con políticas voraces en las que los UAVs realizan las tareas conforme son detectadas, lo que permitiría evaluar la eficiencia de una asignación óptima.

Agradecimientos

Este trabajo ha sido realizado en el marco del proyecto RAISE (PID2023-149683OB-I00) del Ministerio de Ciencia, Innovación y Universidades del Gobierno de España.

Referencias

- Caballero, A., Tapia, R., Ollero, A., 2024. Combining route planning and visual servoing for offshore wind-turbine inspection using uavs. In: Iberian Robotics Conference. pp. 1–6.
DOI: 10.1109/ROBOT61475.2024.10796917
- Chen, M., Wang, H., Chang, C.-Y., Wei, X., 2020. SIDR: A swarm intelligence-based damage-resilient mechanism for UAV swarm networks. IEEE Access 8, 77089–77105.
DOI: 10.1109/ACCESS.2020.2989614
- Choi, U., Lee, S., 2023. Bandwidth-aware coverage path planning for swarm of uavs with aerial base station. In: International Conference on Unmanned Aircraft Systems. pp. 360–365.
DOI: 10.1109/ICUAS57906.2023.10156244
- Chung, S.-J., Paranjape, A. A., Dames, P., Shen, S., Kumar, V., 2018. A survey on aerial swarm robotics. IEEE Transactions on Robotics 34 (4), 837–855.
DOI: 10.1109/TRO.2018.2857475
- Gañán, F. J., Sanchez-Díaz, J. A., Tapia, R., Martínez-de Dios, J. R., Ollero, A., 2022. Efficient event-based intrusion monitoring using probabilistic distributions. In: IEEE International Symposium on Safety, Security, and Rescue Robotics. pp. 211–216.
DOI: 10.1109/SSRR56537.2022.10018655
- Garrido-Jurado, S., Muñoz-Salinas, R., Madrid-Cuevas, F. J., Marín-Jiménez, M. J., 2014. Automatic generation and detection of highly reliable fiducial markers under occlusion. Pattern Recognition 47 (6), 2280–2292.
DOI: 10.1016/j.patcog.2014.01.005
- Hou, K., Yang, Y., Yang, X., Lai, J., 2020. Cooperative control and communication of intelligent swarms: A survey. Control Theory and Technology 18 (2), 114–134.
DOI: 10.1007/s11768-020-9195-1
- Javed, S., Hassan, A., Ahmad, R., Ahmed, W., Ahmed, R., Saadat, A., Guizani, M., 2024. State-of-the-art and future research challenges in UAV swarms. IEEE Internet of Things Journal 11 (11), 19023–19045.
DOI: 10.1109/JIOT.2024.3364230
- Kuhn, H. W., 1955. The hungarian method for the assignment problem. Naval Research Logistics Quarterly 2 (1-2), 83–97.
DOI: 10.1002/nav.3800020109
- Laporte, G., 2007. What you should know about the vehicle routing problem. Naval Research Logistics 54 (8), 811–819.
DOI: 10.1002/nav.20261
- Lepetit, V., Moreno-Noguer, F., Fua, P., Feb. 2009. EPnP: An accurate $O(n)$ solution to the PnP problem. International Journal of Computer Vision 81 (2), 155–166.
DOI: 10.1007/s11263-008-0152-6
- Martínez-de Dios, J. R., Gómez Eguíluz, A., Rodríguez-Gómez, J. P., Tapia, R., Ollero, A., 2020. Towards UAS surveillance using event cameras. In: IEEE International Symposium on Safety, Security, and Rescue Robotics. pp. 71–76.
DOI: 10.1109/SSRR50563.2020.9292606
- Munkres, J., 1957. Algorithms for the assignment and transportation problems. Journal of the Society for Industrial and Applied Mathematics 5 (1), 32–38.
DOI: 10.1137/0105003
- Tapia, R., Gil-Castilla, M., Martínez-de Dios, J. R., Ollero, A., Nov. 2024. Autonomous guidance of an aerial robot using 6-DoF visual control and behavior trees. In: Iberian Robotics Conference. pp. 1–6.
DOI: 10.1109/ROBOT61475.2024.10797354
- Tapia, R., Martínez-de Dios, J. R., Ollero, A., 2019. Efficient mosaicking for linear infrastructure inspection using aerial robots. In: Jornadas de Automática.
DOI: 10.17979/spudc.9788497497169.802
- Zhou, Y., Rao, B., Wang, W., 2020. UAV swarm intelligence: Recent advances and future trends. IEEE Access 8, 183856–183878.
DOI: 10.1109/ACCESS.2020.3028865