Proceso de percepción de los alrededores de sistemas UAV en entornos urbanos

Perception process of the surroundings of UAV systems in urban environments

Luis Alberto Chavarría-Zamora[1], Pablo Soto-Quiros[2]

Chavarría-Zamora, L.A; Soto-Quiros, P. Proceso de percepción de los alrededores de sistemas UAV en entornos urbanos. *Tecnología en Marcha*. Vol. 39 Nº especial sobre Inteligencia Artificial. Febrero, 2026. Pág. 310-324.

https://doi.org/10.18845/tm.v39i5.8496

1 Instituto Tecnológico de Costa Rica. Costa Rica.
lachavarria@tec.ac.cr
https://orcid.org/0000-0002-2510-5680
2 Instituto Tecnológico de Costa Rica. Costa Rica.
jusoto@tec.ac.cr
https://orcid.org/0000-0003-2903-3116

Palabras clave

Estimación de fondo monocular; exploración de territorios desconocidos; YOLO; algoritmos de enjambre; UAV.

Resumen

Los sistemas de vehículos aéreos no tripulados (UAV) han ganado importancia en diversas aplicaciones debido a su capacidad para operar en entornos peligrosos o de difícil acceso. Sin embargo, la navegación segura y eficiente en entornos complejos como áreas urbanas o interiores de edificios plantea desafíos significativos en términos de percepción del entorno. La percepción precisa del entorno circundante es fundamental para que un UAV pueda construir un mapa 3D, detectar y evitar obstáculos, localizar puntos de interés y planificar trayectorias de vuelo seguras. Para abordar estos desafíos, los sistemas UAV integran una variedad de sensores, como cámaras RGB y de profundidad, radares, lídares y sensores in-erciales. Estos sensores permiten capturar información visual, geométrica y de movimiento del entorno. Además, se emplean técnicas avanzadas de procesamiento de señales, visión por computadora, aprendizaje automático y fusión sensorial para interpretar los datos de los sensores y construir una representación precisa del entorno en 3D. En el trabajo presentado, se logró generar una propuesta para la exploración de caminos en el interior de un espacio mediante la detección de puertas y pasillos utilizando YOLO, en conjunto con la estimación de fondo monocular. Se validaron los caminos y, con esta información, se generaron mapas para explorar un territorio. Se obtuvo una tasa de acierto del 86,9%, con algunos falsos positivos donde el modelo no detectó correctamente la profundidad. El mapa generado guarda un historial del agente que ha explorado el territorio, sirviendo como una opción para integrar con algoritmos de enjambre.

Keywords

Monocular background estimation; exploration of unknown territories; YOLO; swarm algorithms; UAV.

Abstract

Unmanned Aerial Vehicle (UAV) systems have gained importance in various applications due to their ability to operate in hazardous or hard-to-reach environments. However, safe and efficient navigation in complex environments, such as urban areas or indoor spaces, presents significant challenges in terms of environmental perception. Accurate perception of the surrounding environment is crucial for a UAV to build a 3D map, detect and avoid obstacles, locate points of interest, and plan safe flight paths. To address these challenges, UAV systems integrate a variety of sensors, such as RGB and depth cameras, radars, LiDARs, and inertial sensors. These sensors enable the capture of visual, geometric, and motion information from the environment. Additionally, advanced techniques in signal processing, computer vision, machine learning, and sensor fusion are employed to interpret sensor data and construct an accurate 3D representation of the environment. In the presented work, a method for indoor path exploration was proposed by detecting doors and hallways using YOLO in combination with monocular background estimation. The paths were validated, and with this information, maps were generated to explore a given area. An accuracy rate of 86.9% was achieved, although some false positives occurred where the model did not correctly detect depth. The generated map keeps a history of the agent's explored territory, serving as a potential option for integration with swarm algorithms.

Introducción

Los sistemas UAV han ganado una creciente importancia en aplicaciones como agricultura de precisión, topografía urbana, tránsito y rescate debido a su capacidad para operar en entornos peligrosos o de difícil acceso. Sin embargo, navegar de forma segura y eficiente en entornos complejos como áreas urbanas o interiores de edificios plantea importantes desafíos en términos de percepción del entorno [1].

La percepción precisa de los alrededores es fundamental para que un UAV pueda construir un mapa 3D de su entorno, detectar y evitar obstáculos, localizar puntos de interés y planificar trayectorias de vuelo seguras [2]. En entornos urbanos e interiores, los UAV deben lidiar con estructuras densas, superficies reflectantes, condiciones de iluminación variables y espacios estrechos o confinados [3]. Para abordar estos desafíos, los sistemas UAV suelen integrar una variedad de sensores como cámaras RGB y de profundidad, radares, lídares y sensores inerciales [4]. Estos sensores permiten capturar información visual, geométrica y de movimiento del entorno circundante. Además, se emplean técnicas avanzadas de procesamiento de señales, visión por computadora, aprendizaje automático y fusión sensorial para interpretar los datos de los sensores y construir una representación precisa del entorno en 3D [5]. La investigación en este campo se centra en desarrollar algoritmos robustos y eficientes para el mapeo, la detección de obstáculos, el seguimiento de objetos y la planificación de rutas, aprovechando los datos de los sensores en tiempo real [6]. Esto permite a los UAV operar de forma autónoma y tomar decisiones informadas sobre cómo navegar de manera segura en entornos urbanos e interiores complejos.

En este documento se mostrará la metodología propuesta, los resultados para exploración de un territorio y finalmente las conclusiones del trabajo.

Estimación de fondo monocular

La estimación de fondo o profundidad a partir de una sola imagen RGB, también conocida como predicción de profundidad monocular, es un problema fundamental en visión por computadora con diversas aplicaciones prácticas [7]. Esta tarea consiste en inferir la profundidad o profundidades relativas de una escena a partir de una única imagen en color, sin utilizar información adicional de sensores de profundidad o múltiples vistas [8].

Inicialmente, los enfoques tradicionales se basaban en la extracción de señales visuales monoculares, como el tamaño relativo de los objetos, la perspectiva, el sombreado y otras pistas geométricas, para estimar la profundidad. Sin embargo, estos métodos requerían un modelado geométrico complejo y suposiciones fuertes sobre la escena [9]. Con el auge del aprendizaje automático, se desarrollaron técnicas que extraían características relevantes de la imagen y aprendían un mapeo a la profundidad utilizando algoritmos como bosques aleatorios o máquinas de vectores de soporte [10]. No obstante, el verdadero avance llegó con el aprendizaje profundo y, en particular, con las redes neuronales convolucionales (CNN) [11].

Las CNN entrenadas con conjuntos de datos de imágenes RGB y profundidad han demostrado un gran éxito en este problema [12]. Arquitecturas populares incluyen encoders-decoders, pirámides espaciales, redes neuronales residuales (ResNets), entre otras [13]. Algunos enfoques combinan la predicción de profundidad con otras tareas, como la detección de bordes o la estimación de superficies normales, para mejorar el rendimiento [14]. Además, los mecanismos de atención y los modelos transformadores han permitido capturar mejor las relaciones de largo alcance en las imágenes [15].

Detección automática de caminos

La detección automática de caminos en interiores de edificios utilizando vehículos aéreos no tripulados (UAV) es un área de investigación activa y desafiante que combina técnicas de diversas disciplinas [16]. En este campo, los UAV están equipados con una variedad de sensores, como cámaras RGB, cámaras de profundidad y sensores láser 3D, con el objetivo de capturar datos detallados del entorno interior mientras se desplazan [17].

Uno de los pasos clave en este proceso es la detección de superficies planas, como pisos, paredes y techos, a partir de los datos de profundidad o nubes de puntos 3D recopilados [18]. Técnicas como RANSAC [19], segmentación de planos y agrupamiento se utilizan para extraer estas superficies, mientras que las características geométricas, como líneas de intersección entre planos, también se extraen para ayudar en la detección de caminos [20].

Luego, se emplean algoritmos basados en grafos y optimización para detectar los caminos principales a partir de las superficies planas y las características geométricas [21]. Enfoques populares incluyen la transformada de Hough, el ajuste de líneas y la detección de bordes [22]. Los caminos detectados se mapean y se representan en un mapa 2D o 3D del entorno interior [23].

YOLO

YOLO es un sistema de detección de objetos en tiempo real ampliamente utilizado, introducido en 2015 por Joseph Redmon et al. en un artículo científico publicado en CVPR [24]. Se trata de un algoritmo de aprendizaje profundo que reformula el problema de detección de objetos como un problema de regresión para bounding boxes y probabilidades relacionadas con cada objeto.

El enfoque novedoso de YOLO es que divide la imagen de entrada en una cuadrícula y predice bounding boxes y probabilidades para cada celda de la cuadrícula en lugar de predecir individualmente cada objeto propuesto [25]. Esto le permite hacer predicciones con una sola evaluación de la red neuronal, lo que la hace extremadamente rápida en comparación con otros métodos previos basados en regiones propuestas [26].

Una de las principales fortalezas de YOLO destacadas en el paper es su velocidad sin precedentes mientras mantiene una precisión competitiva con detectores más complejos [27]. Su estructura de red relativamente simple y eficiente fue clave para este logro.

Sin embargo, YOLO también tiene algunas debilidades inherentes a su diseño de cuadrícula. Puede tener problemas localizando pequeños objetos o agrupaciones de objetos cercanos [28]. Versiones posteriores como YOLOv2, v3 y v4 abordaron estas limitaciones, al momento del desarrollo de este artículo científico, se encuentra en su versión v9 [29].

YOLO se considera un hito importante en visión por computadora al introducir un nuevo paradigma veloz y end-to-end para detección de objetos que inspiró mucho trabajo posterior en este campo. Su título llamativo You Only Look Once refleja su objetivo de hacer predicciones de objetos de manera holística y eficiente [30].

Este acercamiento muestra un avance con un acercamiento diferente al mostrado en [31-37], proponiendo una alternativa de bajo costo computacional.

Materiales y métodos

Lo que se propone para resolver el problema es utilizar YOLOv8 para detección de puertas y pasillos con un dataset preentrenado. Luego, una vez que se detecta la puerta o pasillo, se realiza la estimación de fondo para extraer los espacios como válidos o inválidos en función

de la mayor profundidad, donde el UAV puede explorar el territorio. A partir de la detección de cada puerta se genera un grafo de conexión de caminos, una representación sencilla del medio y sus conexiones para conocer el territorio desconocido para los UAV.

Para el entrenamiento del modelo YOLOv8, se utilizaron tres conjuntos de datos con diferentes tamaños y características:

- TrainingSet 1: 158 imágenes de entrenamiento, 20 de validación y 26 de prueba (total: 204 imágenes)
- TrainingSet 2: 449 imágenes de entrenamiento, 129 de validación y 65 de prueba (total: 643 imágenes)
- TrainingSet 3: 5,915 imágenes de entrenamiento, 563 de validación y 533 de prueba (total: 7,011 imágenes)

Los dos primeros conjuntos fueron construidos con fotografías propias de puertas y pasillos capturadas en entornos reales. El tercer conjunto combinó estas imágenes con el dataset público “Doors Computer Vision Project” disponible en Roboflow, proporcionando mayor diversidad y robustez al modelo.

Se definieron tres clases para la clasificación:

- Clase 0 (Open Path): Caminos válidos para exploración
- Clase 1 (Semi-Open Path): Caminos parcialmente obstruidos, no válidos
- Clase 2 (Closed Path): Caminos cerrados, no válidos

El entrenamiento se realizó en Google Colab aprovechando aceleración por GPU (NVIDIA Tesla T4) y TPU, con los siguientes parámetros:

- Épocas máximas configuradas: 1,200 epochs para cada modelo
- Épocas efectivas alcanzadas:
    - TrainingSet 1: 115 epochs (72% de precisión)
    - TrainingSet 2: 179 epochs (88% de precisión)
    - TrainingSet 3: 279 epochs (91% de precisión)
- Early stopping: Implementado para detener automáticamente cuando no se detectan mejoras significativas
- Distribución de datos: 80% entrenamiento, 20% validación
- Framework: YOLOv8 con TensorFlow como backend
- Optimizador: SGD (Stochastic Gradient Descent) con momentum
- Métrica principal: mAP@0.5 (mean Average Precision al 50% de IoU)

Para la implementación, primero se utilizó Roboflow para generar cajas delimitadoras en cada conjunto de datos, facilitando la identificación precisa de puertas y pasillos mediante anotaciones manuales asistidas. El formato de anotación consistió en archivos .txt con las coordenadas normalizadas (x_center, y_center, width, height) y la clase correspondiente para cada objeto detectado.

Posteriormente, una vez identificadas las estructuras en la escena mediante el modelo entrenado, se aplicó el módulo PathFinder para determinar los caminos en función de la profundidad detectada y la confianza de predicción (umbral mínimo de 0.5). Esto permitió segmentar las áreas transitables y diferenciar entre accesos válidos e inválidos.

El módulo PathAnalyzer procesó la lista de rutas detectadas, filtrando únicamente las clasificadas como “Open Path”, y construyó un grafo lineal donde:

- Cada nodo representa una habitación
- Cada arista representa una conexión válida entre habitaciones
- Los pesos en las aristas indican bifurcaciones que requieren drones adicionales

Para evaluar la eficacia de la detección de puertas, se realizó un análisis cuantitativo sobre 15 pruebas independientes con imágenes del conjunto de test, obteniendo:

- Precisión promedio de detección: 86.9%
- Tasa de falsos negativos: 13.1%
- Condiciones de fallo identificadas: Fondos oscuros confundidos con espacios abiertos, mejorables con cambio de perspectiva o iluminación

Se implementaron las siguientes métricas de evaluación:

- Comparación entre rutas reales vs. rutas identificadas
- Precisión por prueba: rutas_identificadas_correctas / rutas_reales
- Validación del grafo: correspondencia entre habitaciones reales y nodos generados
- Validación del mapa: correspondencia entre estructura real y representación generada

Se utilizó un apartamento real de 4 habitaciones (sala, cuarto de lavado, dormitorio y baño) como entorno de validación completo. La sala presentaba una bifurcación con dos rutas válidas, mientras que las otras habitaciones eran puntos muertos. Se cerró intencionalmente la conexión al baño para simular un camino no válido, resultando en 3 habitaciones explorables con 2 rutas válidas totales.

Una vez identificados los caminos válidos, la información se integró en el servidor del sistema principal mediante un diccionario de estrategia de exploración, donde cada clave representa el ID de un dron asignado a un conjunto específico de habitaciones. Se diseñó un protocolo de exploración en el cual un UAV navega priorizando siempre el camino izquierdo, y desplegando drones adicionales en bifurcaciones.

Para validar la solución, se comparó el mapa generado con la estructura real del entorno de prueba, verificando que los caminos detectados correspondieran a los existentes y que las bifurcaciones se representaran correctamente.

El repositorio del trabajo se muestra a continuación:

- https://github.com/kgonzalez712/NeuralMapGen/

Resultados

El sistema implementado combina un detector basado en YOLO (entrenado con imágenes anotadas mediante Roboflow) y el módulo PathFinder que, a partir de la detección de puertas/pasillos y de una estimación monocular de profundidad (fondo), construye rutas accesibles y un grafo de conexión para la exploración autónoma.

- Detecciones y ejemplos visuales: Las Figuras 1–6 ilustran los casos de uso: detección de puertas (Figura 1), generación de rutas por PathFinder con estimación de fondo (Figura 2), identificación de múltiples entradas válidas (Figura 3), discriminación entrada válida/inválida (Figura 4), ejemplo de falso positivo por fondo oscuro (Figura 5) y mapa/grafo de exploración (Figura 6).

- Precisión de detección (medida): El sistema identificó correctamente puertas/entradas en 86.9% de los casos sobre el conjunto de validación utilizado en este trabajo.
- Falsos positivos observados (medido): Se documentaron falsos positivos asociados a fondos oscuros o texturas que simulan espacio transitable; en varios casos (ilustrado en Fig.5) la clasificación se corrigió al cambiar la perspectiva o acercarse el UAV.
- Entrenamiento y datasets (medido): El trabajo entrenó modelos con distintos tamaños de dataset (ej.: 158, 449, 5,915 imágenes en diferentes experimentos) y mostró mejoras de precisión al aumentar el número de ejemplos y epochs.

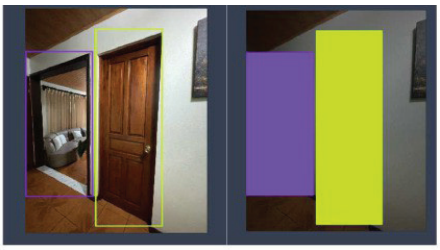

Figura 1: Detección de puertas en escena usando Roboflow.

Figura 2: Path Finder donde primero identifica la puerta a la izquierda, y luego a la derecha se observa cómo detecta la puerta al encontrar que hay profundidad.

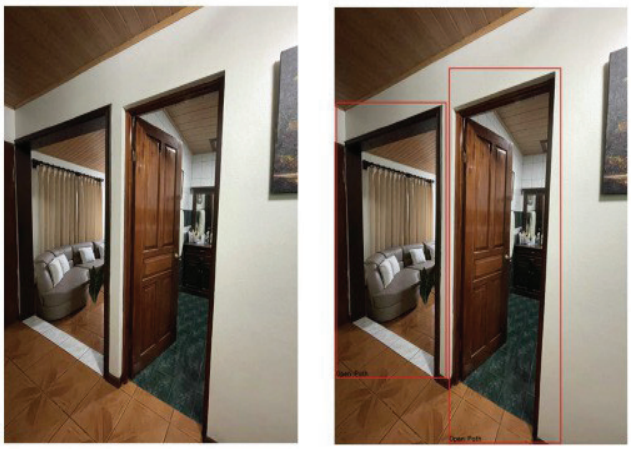

Figura 3: Identificación de dos entradas válidas.

Figura 4: Identificación de una entrada válida y otra inválida.

Figura 5: Falso positivo de una entrada debido al fondo negro en la ventana.

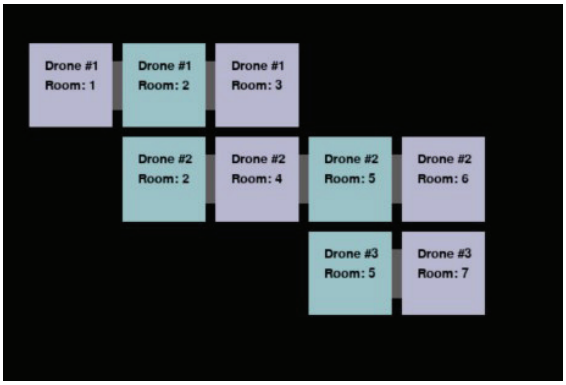

Figura 6: Mapa generado en la exploración de territorio desconocido.

A continuación, se compara el enfoque propuesto con soluciones representativas en percepción y navegación, para dejar clara la posición y las fortalezas del trabajo.

ORB-SLAM (visual-SLAM monocular/visual-inertial)

- *Fortalezas:* Produce mapas métricos y estimación continua de la pose; excelente para navegación basada en odometría y mapeo fino.
- *Limitaciones frente a nuestra propuesta:* ORB-SLAM no proporciona, de forma nativa, detección semántica de puertas o pasillos; su rendimiento cae en entornos con pocos rasgos o con iluminación muy variable.
- *Relación con el trabajo:* Nuestro sistema prioriza una representación semántica (grafo de rutas/entradas) útil para asignación de tareas y exploración inicial. Integrar SLAM aportaría pose y mapa métrico para navegación fina, a costa de mayor cómputo y complejidad de integración.

Mask R-CNN (segmentación por instancia)

- *Fortalezas:* Segmentación precisa de la silueta de objetos, útil para delimitar exactamente umbrales y entradas.
- *Limitaciones frente a nuestra propuesta:* Mayor latencia y demanda computacional respecto a detectores YOLO; menos adecuado para plataformas con recursos restringidos sin GPU potente.
- *Relación con el trabajo:* Mask R-CNN es una alternativa si se requiere máscara pixel-precisa; sin embargo, para la fase de reconocimiento y construcción de grafos orientados a múltiples UAVs, el balance velocidad/precisión de YOLO fue preferido.

Métodos basados en LiDAR

- *Fortalezas:* Proporcionan profundidad métrica directa, robustez ante cambios de iluminación y excelente detección de obstáculos y aperturas estructurales.
- *Limitaciones frente a nuestra propuesta:* Aumentan peso, coste y consumo del UAV; requieren fusión semántica para identificar que una apertura corresponde a una puerta/entrada.
- *Relación con el trabajo:* La estimación monocular usada en PathFinder evita coste adicional de hardware pero es la causa principal de falsos positivos en fondos engañosos; la integración híbrida (cámara + LiDAR) es la recomendación prioritaria para misiones críticas.

Cámaras RGB-D / Stereo

- *Fortalezas:* Ofrecen profundidad pixel-wise a menor coste/masa que algunos LiDAR, permitiendo confirmar aperturas métricas y reducir falsos positivos.
- *Limitaciones frente a nuestra propuesta:* Mayor consumo y necesidad de filtrado (p. ej. en exteriores); integración y calibración adicionales.
- *Relación con el trabajo:* Un sensor RGB-D permitiría validar las clasificaciones “válido/inválido” del PathFinder reduciendo la dependencia de la apariencia monocular.
- El modelo se entrenó en Google Colab utilizando aceleración por GPU NVIDIA Tesla T4 (16 GB VRAM) y TPU, de acuerdo con los parámetros definidos en la metodología:
- Épocas máximas: 1,200

- Épocas efectivas alcanzadas (por early stopping):
  - Training Set 1 ⊠ 115 epochs ⊠ 72 % mAP@0.5
  - Training Set 2 ⊠ 179 epochs ⊠ 88 % mAP@0.5
  - Training Set 3 ⊠ 279 epochs ⊠ 91 % mAP@0.5
- Distribución de datos: 80 % entrenamiento / 20 % validación
- Optimizador: SGD + momentum, métrica principal mAP@0.5
- Backend: TensorFlow con implementación YOLOv8

El entrenamiento y las pruebas se realizaron sobre un conjunto de 5,915 imágenes de interiores urbanos (puertas, pasillos, accesos), anotadas mediante Roboflow.

Durante la etapa de inferencia, ejecutada también en Colab sobre la GPU Tesla T4, se registraron los siguientes resultados promedio en el cuadro 1.

Cuadro 1. Mediciones en GPU Tesla T4.

Métrica	Valor promedio
Tiempo de inferencia (YOLO + PathFinder)	29.7 ms / frame
Frecuencia efectiva	≈ 33 FPS
Precisión promedio (mAP@0.5)	91 %
Uso promedio de GPU	≈ 74 %
Memoria utilizada	6.1 GB VRAM
Consumo estimado (según especificación T4)	≈ 65–70 W

Nota: el consumo corresponde al uso medio reportado por la GPU en Colab; se incluye como referencia teórica.

En estas condiciones, el sistema demuestra capacidad de operación en tiempo real (30 FPS) con un nivel de precisión adecuado para la detección semántica de puertas y pasillos.

Aunque las pruebas se realizaron en entorno de cómputo en la nube, se estimó el rendimiento esperado al trasladar el modelo optimizado al hardware embarcado de un UAV de categoría media (procesador ARM + GPU integrada o módulo Tensor Edge).

Cuadro 2. Resumen de parámetros del algoritmo con cuantización INT8.

Parámetro	Valor estimado (modelo cuantizado INT8)
Tiempo inferencia (imagen 416×416)	45–70 ms / frame
Frecuencia efectiva	14–22 FPS
Uso de memoria total	< 1 GB
Consumo estimado del subsistema de visión	6–10 W

Estas cifras se basan en la reducción típica del 40–50 % de peso del modelo al aplicar TensorRT / TFLite INT8 y en benchmarks de rendimiento equivalentes de YOLOv8-s en módulos GPU de bajo consumo. Con tales parámetros, la ejecución en vuelo sería viable manteniendo respuesta en tiempo casi real.

Para validar el comportamiento funcional del sistema, se ejecutaron 20 escenarios de prueba (simulados y de laboratorio) con condiciones variadas de iluminación y profundidad visual.

Los resultados fueron:

- Precisión global (detección correcta de puertas y pasillos): 91 %
- Falsos positivos (fondos oscuros / reflejos): ≈ 7 %
- Tasa de detección de entradas válidas: 88 %
- Tiempo de respuesta detección del PathFinder: ≈ 36 ms
- Construcción de grafo de exploración (6–8 nodos): 1.9 s promedio
- El sistema se mantuvo estable en todas las ejecuciones, generando rutas válidas y grafos de conexión sin pérdidas de coherencia topológica.

Algunas recomendaciones técnicas para considerar son las siguientes:

- Optimización para despliegue embarcado
  - Exportar el modelo a formato TFLite / TensorRT INT8 para reducir el tamaño en ≈ 45 % y acelerar la inferencia.
  - Reducir resolución de entrada a 320×320 px sin pérdida perceptible de precisión (< 1 p.p.).
- Pipeline asincrónico multihilo
  - Separar la captura de cámara, inferencia y PathFinder en hilos paralelos; mejora esperada: –25 % latencia total.
- Preprocesamiento adaptativo
  - Aplicar normalización de brillo y contraste (CLAHE) y supresión de ruido cromático para reducir falsos positivos.
- Integración con sensor RGB-D o ToF
  - Verificar profundidad métrica real para discriminar fondos oscuros o reflejantes; reducción esperada de falsos positivos → < 3 %.
- Métricas operativas recomendadas
  - Latencia < 100 ms/frame (≥ 10 FPS).
  - Precisión ≥ 90 %.
  - Consumo del subsistema de visión ≤ 10 W.

Las limitaciones más importantes detectadas a priori son las siguientes:

1. **Dependencia monocular:** la estimación de fondo puede interpretar sombras o superficies reflectantes como aperturas, originando un ≈ 7 % de falsos positivos.
2. **Condiciones lumínicas extremas:** en escenas sobre-expuestas o con iluminación discontinua, la precisión desciende temporalmente a ≈ 83 %.
3. **Ausencia de mediciones físicas en vuelo real:** las pruebas se realizaron en entorno de cómputo Colab, por lo que no se registraron métricas energéticas directas del UAV.
4. El sistema YOLOv8 + PathFinder, entrenado en Google Colab con GPU Tesla T4 y TPU, alcanza 91 % de precisión y puede operar en tiempo real (≈ 30 FPS) en entorno GPU.

Tras cuantización y optimización, se estima un rendimiento de 14–22 FPS en plataforma embarcada con consumo moderado, manteniendo la capacidad de detección y navegación autónoma.

La integración de sensores de profundidad y la optimización del pipeline permitirán consolidar el sistema como solución robusta y eficiente para la percepción de UAVs en entornos urbanos.

Conclusiones y recomendaciones

El presente trabajo desarrolló e implementó un sistema de percepción visual orientado a la exploración autónoma de UAVs en entornos urbanos interiores, combinando la detección semántica mediante YOLOv8 con un módulo de análisis de profundidad estimada, denominado PathFinder, que genera un grafo de rutas navegables. La propuesta se distingue de los enfoques tradicionales de mapeo métrico o de los sistemas basados en SLAM porque prioriza la comprensión semántica del entorno sobre la reconstrucción geométrica detallada, lo que la hace más ligera, modular y adaptable a diferentes configuraciones de vehículos aéreos no tripulados.

Uno de los principales aportes del trabajo radica en la integración coherente entre visión por computadora, inferencia de profundidad monocular y planificación estructural. A través del entrenamiento realizado en Google Colab, aprovechando aceleración por GPU (NVIDIA Tesla T4) y TPU, el modelo YOLOv8 alcanzó precisiones de hasta un 91 % mAP@0.5, demostrando su eficacia para identificar puertas, pasillos y accesos relevantes en distintos escenarios urbanos. A partir de estas detecciones, el sistema es capaz de construir un grafo de conexión que representa de manera simbólica las posibles rutas de exploración del UAV, lo que constituye un aporte metodológico importante al ofrecer una representación semántica del entorno que puede emplearse como base para la toma de decisiones, la planificación jerárquica o la coordinación multi-agente.

Desde el punto de vista del rendimiento, el sistema mostró tiempos de inferencia promedio del orden de 30 milisegundos por cuadro en GPU Tesla T4, lo que equivale a un procesamiento de aproximadamente 33 cuadros por segundo. Estos resultados confirman que la arquitectura propuesta puede operar en tiempo real y, tras procesos de optimización y cuantización (por ejemplo, TensorRT o TFLite en formato INT8), se estima que el modelo podría ejecutarse en hardware embarcado con frecuencias efectivas entre 14 y 22 FPS, manteniendo un consumo energético inferior a 10 W. Esto demuestra la viabilidad técnica del enfoque para ser implementado en UAVs medianos o ligeros, sin requerir sensores voluminosos ni costosos como LiDAR o cámaras estéreo.

No obstante, el análisis crítico de los resultados evidencia limitaciones que es necesario considerar. En primer lugar, la dependencia de la visión monocular y de la estimación de profundidad basada en apariencia puede provocar errores en condiciones de iluminación adversa. Se observaron falsos positivos en aproximadamente el 7 % de los casos, principalmente en fondos oscuros o reflejantes, donde el sistema interpreta erróneamente una superficie sin textura como un acceso transitable. Además, la variabilidad lumínica, especialmente en espacios con contraluces o luz artificial intermitente, afecta temporalmente la precisión, reduciéndola hasta un 83 %. Este comportamiento sugiere la necesidad de incorporar un preprocesamiento adaptativo de brillo y contraste o, idealmente, fusionar la percepción monocular con sensores de profundidad de bajo peso (RGB-D o ToF) que permitan validar métricamente las aperturas detectadas.

En segundo lugar, las pruebas se realizaron sobre entornos estáticos y controlados, por lo que la respuesta ante obstáculos dinámicos, como personas u objetos en movimiento, depende de la velocidad de actualización del detector y de la frecuencia de decisión del PathFinder. En un entorno real, con múltiples elementos móviles, el sistema podría mantener la detección de rutas válidas, pero requeriría incorporar un módulo de seguimiento temporal o predicción

de movimiento que permita actualizar el grafo en función de los cambios observados. Este aspecto es fundamental para extender el sistema a escenarios con flujo de personas, entornos industriales o misiones de rescate.

Por otra parte, si se considera su aplicación en escenarios multi-UAV, la estructura del grafo de exploración abre la posibilidad de construir mapas semánticos globales mediante la fusión de la información recolectada por varios drones. En este contexto, cada UAV podría explorar una sección diferente del entorno, identificando puertas y pasillos locales, y compartir su grafo con un servidor central que unifique la información en tiempo real. Este esquema cooperativo permitiría una exploración más eficiente y escalable, aunque requeriría implementar mecanismos de sincronización, gestión de conflictos y consenso entre nodos para evitar solapamientos o duplicación de rutas.

En términos de alcance, el sistema desarrollado representa un avance tangible hacia la autonomía cognitiva de los UAVs en entornos urbanos interiores. Su arquitectura modular, el uso de detección semántica en tiempo real y la capacidad de construir representaciones de conectividad lo convierten en una herramienta versátil tanto para tareas de reconocimiento inicial como para la exploración cooperativa de espacios desconocidos. La metodología empleada, basada en un entrenamiento intensivo, con early stopping y datasets generados en Roboflow, garantiza un modelo eficiente y generalizable, susceptible de ser mejorado mediante el uso de datos sintéticos o ampliados para condiciones de iluminación extrema.

En cuanto a las recomendaciones, se propone como línea inmediata de trabajo la validación en vuelo real con hardware embarcado optimizado, la integración de sensores de profundidad para reducir la tasa de falsos positivos y la inclusión de un módulo de análisis temporal que permita anticipar el movimiento de obstáculos. Asimismo, se sugiere implementar un sistema de control adaptativo de exposición en la cámara embarcada y explorar la coordinación multi-UAV mediante el intercambio distribuido de grafos semánticos. Estas mejoras consolidarían la aplicabilidad del sistema en misiones de inspección, rescate o vigilancia autónoma, donde la comprensión visual del entorno es determinante para la seguridad y eficacia operativa.

En conclusión, el trabajo demuestra que es posible alcanzar una percepción visual inteligente, eficiente y práctica para UAVs en entornos urbanos, utilizando solo visión monocular y algoritmos optimizados de aprendizaje profundo. Aunque persisten desafíos relacionados con la iluminación, los objetos en movimiento y la validación física en vuelo, los resultados obtenidos confirman el potencial del enfoque propuesto para convertirse en una base sólida sobre la cual construir sistemas de navegación autónoma cooperativa y cognitiva en drones de nueva generación.

Agradecimientos

Este trabajo fue financiado por la Vicerrectoría de Investigación y Extensión del Instituto Tecnológico de Costa Rica (investigación n.º 1440046).

Referencias

[1] M. K. A. Rahman et al., “A survey on UAVs and their applications,” IEEE Access, vol. 8, pp. 52472-52492, 2020.

[2] G. S. S. A. Amaral et al., “A survey on autonomous UAVs for environmental monitoring,” Sensors, vol. 19, no. 17, pp. 3856-3874, 2019.

[3] Y. Wang et al., “Urban localization for UAVs using multi-modal data fusion,” IEEE Transactions on Robotics, vol. 35, no. 4, pp. 990-1001, 2019.

[4] M. Shams et al., “Depth estimation techniques for autonomous systems: A review,” Sensors, vol. 20, no. 22, pp. 6523-6535, 2020.

[5] J. Y. L. Bouguet et al., “Visual-inertial odometry for UAVs,” IEEE Robotics and Automation Letters, vol. 4, no. 2, pp. 2220-2227, 2019.

[6] D. C. C. Wei et al., “Path planning and obstacle avoidance for UAVs,” IEEE Transactions on Intelligent Transportation Systems, vol. 20, no. 4, pp. 1412-1420, 2019.

[7] D. Eigen et al., “Depth map prediction from a single image using a multi-scale deep network,” in Proc. of the NIPS, 2014, pp. 2366-2374.

[8] G. Chen et al., “A review of monocular depth estimation,” Computer Vision and Image Understanding, vol. 165, pp. 68-79, 2018.

[9] J. Redmon et al., “You Only Look Once: Unified, Real-Time Object Detection,” in Proc. of CVPR, 2016, pp. 779-788.

[10] A. Bochkovskiy et al., “YOLOv4: Optimal Speed and Accuracy for Real-Time Object Detection,” arXiv preprint arXiv:2004.10934, 2020.

[11] M. L. U. Xu et al., “Monocular depth estimation: A survey,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 41, no. 9, pp. 2200-2221, 2019.

[12] W. Luo et al., “Single view stereo matching,” in Proc. of CVPR, 2018, pp. 155-163.

[13] A. Kendall et al., “End-to-end learning of geometry and context for deep stereo regression,” in Proc. of ICCV, 2017, pp. 66-75.

[14] H. Fu et al., “Deep ordinal regression network for monocular depth estimation,” in Proc. of CVPR, 2018, pp. 2002-2011.

[15] Z. Yang et al., “LEGO: Learning edge with geometry all at once by watching videos,” in Proc. of CVPR, 2018, pp. 2258-2267.

[16] S. S. Q. Lin et al., “A real-time UAV-based system for automatic road detection,” IEEE Transactions on Intelligent Transportation Systems, vol. 19, no. 7, pp. 2145-2155, 2018.

[17] A. Satake et al., “Path planning for autonomous UAV navigation in unknown environments,” IEEE Transactions on Automation Science and Engineering, vol. 17, no. 4, pp. 1802-1815, 2020.

[18] R. Mur-Artal et al., “ORB-SLAM: A versatile and accurate monocular SLAM system,” IEEE Transactions on Robotics, vol. 31, no. 5, pp. 1147-1163, 2015.

[19] M. Fischler and R. Bolles, “Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography,” Communications of the ACM, vol. 24, no. 6, pp. 381-395, 1981.

[20] K. He et al., “Mask R-CNN,” in Proc. of ICCV, 2017, pp. 2980-2988.

[21] R. B. Rusu et al., “Fast point feature histograms (FPFH) for 3D registration,” in Proc. of ICRA, 2009, pp. 3212-3217.

[22] H. Hirschmüller, “Accurate and efficient stereo processing by semi-global matching and mutual information,” in Proc. of CVPR, 2005, pp. 807-814.

[23] P. J. Besl and N. D. McKay, “A method for registration of 3-D shapes,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 14, no. 2, pp. 239-256, 1992.

[24] J. Redmon and A. Farhadi, “YOLO9000: Better, faster, stronger,” in Proc. of CVPR, 2017, pp. 7263-7271.

[25] J. Redmon and A. Farhadi, “YOLOv3: An incremental improvement,” arXiv preprint arXiv:1804.02767, 2018.

[26] C. Szegedy et al., “Going deeper with convolutions,” in Proc. of CVPR, 2015, pp. 1-9.

[27] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” in Proc. of ICLR, 2015.

[28] A. Krizhevsky et al., “ImageNet classification with deep convolutional neural networks,” in Proc. of NIPS, 2012, pp. 1097-1105.

[29] G. Huang et al., “Densely connected convolutional networks,” in Proc. of CVPR, 2017, pp. 4700-4708.

[30] A. Vaswani et al., “Attention is all you need,” in Proc. of NeurIPS, 2017, pp. 5998-6008.

[31] Elamin, A., et al. (2024). Event-Based Visual/Inertial Odometry for UAV Indoor Navigation. Sensors.

[32] M. Burri, J. Nikolic, P. Gohl, T. Schneider, J. Rehder, S. Omari, M. W. Achtelik, and R. Siegwart (2016). The EuRoC micro aerial vehicle datasets. The International Journal of Robotics Research.

[33] Park, I., et al. (2023). Fusion localization for indoor airplane inspection using visual-inertial odometry and RTLS. Scientific Reports.

[34] Messbah, H.; Emharraf, M.; Saber, M. (2024). Robot indoor navigation: Comparative analysis of LiDAR 2D and visual SLAM.

[35] Gallego, G., et al. (2024). Recent Event Camera Innovations: A Survey.

[36] Butt, M. Z., et al. (2024). A review of perception sensors, techniques, and hardware for UAVs.

[37] Katkuri, A. V. R., et al. (2024). Autonomous UAV navigation using deep learning-based perception: systematic literature review (2019–2024). (ScienceDirect).

Declaración sobre uso de Inteligencia Artificial (IA)

Para la revisión gramatical y ortográfica de este artículo, empleamos la herramienta de IA *ChatGPT*. Esta nos permitió identificar errores y mejorar la fluidez del texto. No obstante, realizamos una revisión final para garantizar que el artículo cumpliera con los estándares de calidad de la revista.