

Seguimiento autónomo de personas con un robot aéreo no tripulado

Fabiola Guevara Soriano, Antonio Alfredo Reyes Montero, Abraham Sánchez López

Benemérita Universidad Autónoma de Puebla, Facultad de Ciencias de la Computación,
Puebla, Puebla, México

fabiola.guevara@outlook.com, reyes-fred@hotmail.com,
asanchez@cs.buap.mx

Resumen. El presente trabajo propone combinar dos grandes áreas de la Inteligencia Artificial que son: Visión por computadora y Planificación de Movimientos, para ser incorporadas en el área de robótica aérea a través de robots aéreos no tripulados. Nuestro objetivo principal es el seguimiento de personas utilizando los algoritmos de detección y seguimiento: Histograma de Gradientes Orientados (HOG – Histogram of Oriented Gradient) y Filtro de Partículas, e integrarlo con la planificación de movimiento para el vuelo autónomo del cuadricóptero a través del algoritmo Árboles de Exploración Rápida (RRT - Rapidly-exploring Random Tree) y variantes del mismo. Se presentan los resultados de los algoritmos así como su efectividad y finalmente se muestra el desempeño del AR.Drone 2 con las técnicas propuestas.

Palabras clave: cuadricóptero, HOG, filtro de partículas, RRT, RRT Star, RRT Connect, TRRT, LazyRRT.

Autonomous People Tracking with an Unmanned Aerial Robot

Abstract. This work proposes to combine two main areas of the Artificial Intelligence which are: Computer vision and Motion Planning, to be incorporated in the area of aerial robotics through unmanned aerial robots. Our main objective is the tracking of people using the detection and tracking algorithms: HOG (Histogram of Oriented Gradients) and Particle Filter besides integrate it with motion planning for the autonomous flight of the quadcopter through the algorithm Rapidly-exploring Random Tree (RRT) trees and variants of it. We present the results of the algorithms as well as their effectiveness and finally the performance of the AR.Drone 2 with the proposed techniques.

Key words: quadcopter, HOG, particle filter, RRT, RRT Star, RRT Connect, TRRT, LazyRRT.

1. Introducción

Los vehículos aéreos no tripulados (UAV - Unmanned Aerial Vehicles), se han desarrollado en los últimos años con mayor frecuencia en los ámbitos científico y comercial. La gran popularidad que han tenido los convierte en una de las tendencias de desarrollo más recientes y han sido un área de investigación muy activa.

La aplicación de este tipo de vehículos es cada día mayor en tareas que implican algún tipo de dificultad o riesgo para vehículos convencionales tripulados por personas como: la detección de incendios, la identificación de manchas de petróleo en el mar, el seguimiento del tráfico, la inspección de líneas de tendido eléctrico, etc. [1].

Existe un interés general por el control autónomo de vehículos aéreos no tripulados, ya que en los últimos años se han desarrollado muchos proyectos relacionados con este tema. Es claro que una aplicación de esta naturaleza debe incluir un algoritmo de planificación de movimientos y una estrategia para el seguimiento de la trayectoria generada por este. Por lo tanto a través del algoritmo RRT y sus variantes se hace la planificación de movimientos especializada en robots aéreos no tripulados.

El seguimiento de objetos con UAVs ha creado un gran interés entre la comunidad científica, sobre todo por sus aplicaciones como la de inspección de objetos (grabación) que se mueven, sin la necesidad de controlar el UAV desde una estación remota. Por ello es importante desarrollar estrategias que permitan realizar el seguimiento de personas (tracking) en ambientes exteriores, utilizando los algoritmos de detección de objetos, en este caso el algoritmo HOG y de seguimiento de objetos como lo es el Filtro de Partículas [2].

El objetivo de este trabajo es conjuntar de manera eficiente dos técnicas: un algoritmo para la planificación de movimientos especializado en robots aéreos no tripulados y un algoritmo que permita la detección y seguimiento de personas (tracking) en ambientes exteriores. Esto da pauta a diversas aplicaciones en el mundo real puesto que la autonomía de nuestra propuesta permitiría la toma de decisiones en tiempo real para el seguimiento de cualquier individuo en diversos escenarios.

El presente artículo primeramente explica de forma sucinta una descripción de los conceptos fundamentales necesarios para la comprensión del tema a desarrollar. Posteriormente presenta las pruebas del funcionamiento de cada uno de los algoritmos desarrollados, cuya finalidad es evaluar su desempeño. A continuación describe el proceso de integración de los algoritmos con el AR.Drone 2.0 para dar a conocer los logros alcanzados en cuanto al seguimiento autónomo de personas en tiempo real. Finalmente exponen las conclusiones a las que se llegaron, tomando en cuenta las restricciones del proyecto. También presenta la perspectiva y discusión de trabajos futuros que pudieran derivar de la continuación del desarrollo del proyecto en este campo de investigación.

2. Vehículos aéreos no tripulados

Los robots aéreos son dispositivos que se encuentran cada vez más presentes en nuestro entorno, gracias a la miniaturización de los componentes electrónicos que los conforman, haciendo posible utilizarlos en situaciones variadas como pueden ser aplicaciones de servicios para: localizar personas, vigilancia, control de incendios, o inspección de zonas de difícil acceso, como es el caso de la inspección de los reactores nucleares dañados en Fukushima en 2011, o la inspección sobre la ciudad fantasma de Chernobyl en 2014, que sufrió las consecuencias del desastre nuclear de 1986, entre otras.

Los UAV son aeronaves capaces de volar sin necesidad de un piloto humano a bordo que las controle. Pueden ser remotamente controlados o volar de forma autónoma basada en planes de vuelo pre-programados mediante el uso de sistemas de control complejos. Además, pueden estar controlados por una estación a bordo que realiza toda la toma de decisiones de forma automática o por operadores que con el uso de un radio control pueden realizar acciones de forma manual como son el despegue y el aterrizaje. La autonomía se relaciona con los algoritmos de control que poseen estos vehículos para responder de manera satisfactoria al encontrarse en eventos inesperados o aleatorios durante sus misiones de vuelo [3].

3. Seguimiento de Objetos

El seguimiento de objetos ha sido desde hace tiempo un tema que ha suscitado gran interés debido a las numerosas aplicaciones en las que se puede emplear. El hecho de poder realizar este tipo de actividades que además no necesitan intervención humana directa, trae consigo muchas ventajas: se realiza de forma mecánica, es decir, que las personas no deben realizar esfuerzo para estar observando por largo tiempo los lugares de interés. Además, de esta forma puede obtenerse la información de movimiento de forma estadística, lo que evita posibles errores humanos que podrían producirse si la supervisión tuviera que llevarla a cabo una persona. Y por último, también consigue alejar a las personas de peligros que puedan acarrear las labores de vigilancia así como permitir el monitoreo de lugares que no sean accesibles a las personas [3].

3.1. Histograma de gradientes orientados (hog)

El algoritmo HOG fue introducido en 2005 por Navneet Dalal y Bill Triggs. La esencia principal de dicho algoritmo es que la forma de un objeto en una imagen puede ser descrita por medio de la distribución de los gradientes.

El objetivo de esta técnica es la extracción de características de una imagen, las cuales identifiquen la silueta de una persona o algún otro objeto. Las características son extraídas teniendo en cuenta los bordes. El proceso de extracción de información de los bordes

presentes en una imagen como se consigue calculando los gradientes y las orientaciones de los píxeles.

La idea principal es que la apariencia y la forma de los objetos pueden ser caracterizados de mejor manera por la distribución de gradientes de intensidad locales, inclusive sin conocimiento del gradiente correspondiente. Esto se hace mediante la división de la imagen en varias regiones pequeñas (celdas), cada una de las cuales tiene un histograma de las direcciones de su gradiente o también llamada orientación de borde de los píxeles de la celda. Las entradas de histograma combinadas forman la representación [4].

3.2. Filtro de partículas

El filtro de partículas fue introducido originalmente en el área de visión computacional como el algoritmo Condensation (CONDitional DENSity PropagATION), que fue presentado por Michael Isard y Andrew Blake para seguir contornos y lo llamaron ICONDENSATION.

El filtro de partículas representa la densidad a posteriori mediante una distribución de partículas en el espacio de estados. Las partículas son estados posibles del proceso, que se pueden representar como puntos en el espacio de estados de dicho proceso. Este enfoque se ha desarrollado de forma independiente en los últimos años en campos como la estadística, la economía o la visión artificial. Los nombres con los que se le ha denominado son: Método Secuencial de Monte Carlo, Algoritmo Condensation, Filtro Bootstrap, Filtro de la Supervivencia del Más Apto, aunque últimamente se está utilizando el término Filtro de Partículas para todos ellos [4].

3.3. Segmentación

La segmentación es un proceso que se encarga de dividir una imagen digital en varias partes. Es utilizada tanto para localizar objetos como para encontrar los límites de estos dentro de una imagen. Concretamente, es un proceso de asignación de una etiqueta a cada píxel de la imagen. Se ha generado una gran cantidad de trabajos que presentan diversas técnicas, modelos y algoritmos. Estas técnicas se dividen en cuatro grandes grupos [4]:

- Técnicas de segmentación basadas en los valores de píxel.
- Técnicas de segmentación basadas en el área.
- Técnicas de segmentación basadas en orillas.
- Técnicas de segmentación basadas en la física.

4. Árboles aleatorios de exploración rápida

Los algoritmos RRT son una técnica desarrollada por Steven M. LaValle y su grupo de colaboradores en la universidad de Illinois, EU. La base de estos métodos es la construcción

incremental de árboles de búsqueda que intentan explorar rápida y uniformemente el espacio de estados, ofreciendo beneficios similares a los obtenidos por otros métodos exitosos de planificación aleatoria. Además, los algoritmos RRT son, particularmente, convenientes para problemas que involucran restricciones diferenciales.

4.1. Rrt

Los algoritmos RRT fueron presentados como una técnica de planificación para búsqueda rápida en espacios de altas dimensiones que tienen tanto restricciones algebraicas (provenientes de los obstáculos) como restricciones diferenciales (originadas por la no-holonomía y la dinámica). La idea clave es dirigir la exploración hacia regiones inexploradas del espacio tomando puntos en el espacio de estados e incrementalmente “jalar” el árbol hacia ellos.

El algoritmo básico de construcción de los RRTs, se muestra en el siguiente algoritmo, en cada iteración se intenta extender el árbol agregando un nuevo vértice en dirección a un estado seleccionado aleatoriamente.

CONSTRUIR_RRT(\mathcal{X}_{ini})

```

1 T.ini(  $\mathcal{X}_{ini}$  );
2 para k=1 a K
     $\mathcal{X}_{aleat} \leftarrow$  ESTADO_ALEATORIO();
3     EXTENDER(T,  $\mathcal{X}_{aleat}$  );
4 regresa T
```

EXTENDER (T,x)

```

1  $\mathcal{X}_{prox} \leftarrow$  VECINO_MAS_PROXIMO(  $\mathcal{X}$  ,T);
2 si NUEVO_ESTADO(  $\mathcal{X}$ ,  $\mathcal{X}_{prox}$ ,  $\mathcal{X}_{nuevo}$ ,  $\mathbf{U}_{nuevo}$  )entonces
3     T.agregaVertice(  $\mathcal{X}_{nuevo}$  );
4     T.agregarArista(  $\mathcal{X}$ ,  $\mathcal{X}_{prox}$ ,  $\mathcal{X}_{nuevo}$ ,  $\mathbf{U}_{nuevo}$  );
5     si  $\mathcal{X}_{nuevo} = \mathcal{X}$  entonces
6         regresa Alcanzado;
7     sino
8         regresa Avanzado;
9 regresa Atrapado;
```

4.2. Variantes rrt

Generalmente consideramos a los RRT como un bloque en la construcción de un planificador eficiente. Por ejemplo, podríamos usar un RRT para escapar de un mínimo local en un planificador de caminos aleatorio con campos de potencial.

La elección recomendada depende de varios factores, tales como: restricciones diferenciales, el tipo de algoritmo para la detección de colisiones y/o la eficiencia en el cómputo del vecino más próximo. A continuación se mencionan varias alternativas de planificadores que utilizan algún algoritmo RRT, los cuales fueron utilizados para esta investigación:

- RRT Connect [5].
- RRT Star [6].
- TRRT [7].
- LazyRRT [8].
- SyclopRRT [9].

5. Resultados experimentales

5.1. Seguimiento de personas

Para la detección de personas, se utilizó con éxito el algoritmo clasificador HOG. Dicha detección está implementada bajo un sistema basado en entrenamiento y aprendizaje. En este sistema se aprenden los rasgos característicos de cada una de las dos clases, correspondientes a personas y no personas.

Para obtener el algoritmo encargado de la detección se cumplieron los siguientes pasos:

1. Se creó el dataset de imágenes normalizadas positivas y negativas. Para el trabajo desarrollado se integraron dos datasets a fin de obtener mejores resultados en la fase de entrenamiento y aprendizaje de nuestro algoritmo lo que se ve reflejado en un mejor rendimiento y detección de personas, los datasets fueron tomados de INRIA [10] y MIT Pedestrian [11]. Las especificaciones son:

- No. de imágenes Positivas: 4450
- No. de imágenes Negativas: 1963
- Dimensión de las imágenes: 64x128
- Formato de imágenes usados: PNG, JPG, PPM, BMP

2. Se obtuvieron los HOG de las imágenes positivas y negativas donde el algoritmo a seguir se muestra a continuación:

1. Normalización de la imagen.
2. Calcular el gradiente de la imagen en X y Y.
3. Calcular los histogramas de los gradientes.
4. Normalización a través de los bloques.
5. Recolección de datos para procesar en el clasificador.

3. Se entrenó el clasificador utilizando las técnicas de clasificación basadas en límites de decisión. Con este fin se usó la máquina de vectores de soporte (SVM – Super Vector Machine), en particular el modelo lineal. Como resultado, el clasificador aprende el patrón o modelo del objeto deseado.

4. Se realizó el testeo del descriptor.

5. Se realizó el testeo en el proceso de detección.

Los resultados del entrenamiento usando un algoritmo propio en conjunto con la herramienta SVMLight [12], se muestran en la Tabla 1. Cabe mencionar que la eficiencia del clasificador se define, entre otras cosas, tomando en cuenta los valores falsos positivos, falsos negativos, verdaderos positivos y verdaderos negativos, que como se puede apreciar van en función del número de imágenes negativas y positivas usadas para el entrenamiento y aprendizaje, entre otros factores.

Tabla 1. Resultados de las variaciones en el entrenamiento.

	Entrenamiento 1	Entrenamiento 2	Entrenamiento 3
Positivos	4450	5985	6000
Negativos	1963	1750	1943
Tiempo de ejecución (ms)	115.77	161.92	160.01
Evaluaciones del Kernel	288344	395196	394160
Norma x 	1.70197	1.70562	1.73833
Norma w 	10.23471	10.23471	10.23475
Resultados	Positivos verdaderos: 4450 Negativos verdaderos: 1871 Falsos Positivos: 91 Falsos Negativos: 1	Positivos verdaderos: 5981 Negativos verdaderos: 1630 Falsos Positivos: 124 Falsos Negativos: 0	Positivos verdaderos: 5981 Negativos verdaderos: 1842 Falsos Positivos: 120 Falsos Negativos: 0

Los resultados finales de la detección de personas en diferentes espacios y tipos de iluminación se muestran en la Figura 1.



Fig. 1. Pruebas del algoritmo de detección de personas.

5.2. Filtro de partículas

Una vez que se realizó la detección de personas, se aplicó el filtro de partículas para hacer el seguimiento de la persona.

En la Figura 2, se puede observar el filtro de partículas cuando el UAV va siguiendo a una persona que es reconocida por el HOG. En color fucsia se ven las partículas, en color verde se plasma una línea que recopila los estados que han sucedido en el tiempo que se realiza el seguimiento, con una cruz roja se observa el estado actual de la persona y con una cruz blanca se observa el estado actual que el filtro de partículas predice.

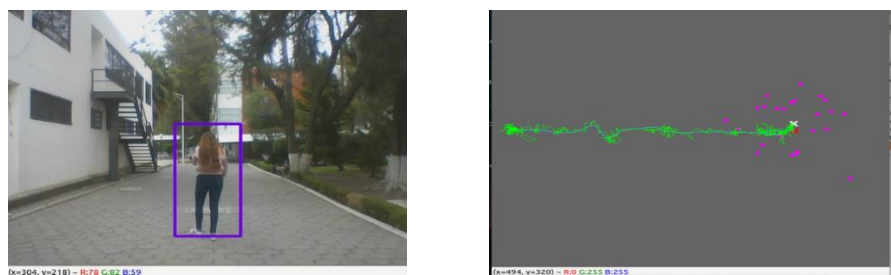


Fig. 2. Pruebas del filtro de partículas (vistas de la cámara y la consola).

5.3. Planificación de movimientos

Se realizaron diversas pruebas a lo largo de la investigación que nos permitieron llegar a un análisis sobre el procesamiento de los diversos algoritmos de planificación de movimientos. Uno de los requerimientos para realizar estas pruebas fue el modelado de 3 escenarios, además del modelado del dron para considerar las restricciones de tamaño que este presenta y de esta forma tener en cuenta la zona deformable y evitar colisiones. Se realizaron 3 escenarios de pruebas distintos, cada uno de estos con ciertas características que podrían dificultar la planificación. Ver los escenarios en la Figura 3.

A través del framework ROS se utilizó la herramienta OMPL app [13] la cual cuenta con un entorno gráfico en donde podemos observar el procesamiento de estos algoritmos y ver la construcción del árbol.

Se realizaron 10 corridas por cada una de las 6 configuraciones de parámetros diferentes del planificador, esto se hizo para cada uno de los 5 algoritmos bajo 3 escenarios diferentes, por lo tanto se tuvieron 900 corridas. En la Figura 4 se muestra un ejemplo de la ejecución del método RRT simple. Esto se simplificó sacando el promedio de tiempo, número de nodos y la eficiencia.

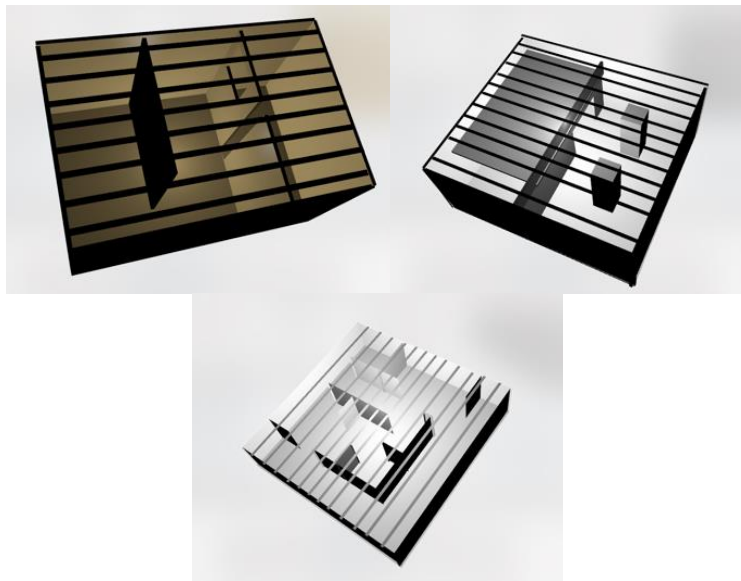


Fig. 3. Escenarios de pruebas 1, 2 y 3.

Características de la computadora para la ejecución.

- Macbook Pro mid 2012,
- 16 GB RAM 1600 MHz DDR3,
- Intel Core i5 - 2.5GHz.

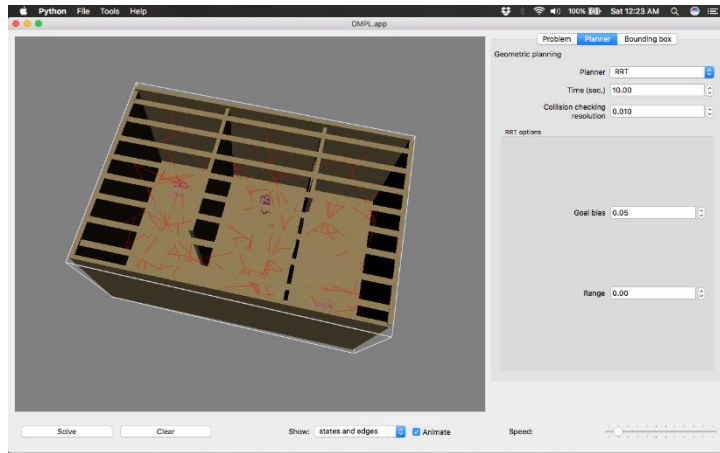


Fig. 4. Ejemplo de la ejecución del método RRT básico en el escenario 1.

La Tabla 2 presenta el porcentaje de eficiencia que tuvo cada algoritmo en cada uno de los escenarios, es decir, el promedio de las 6 corridas.

Tabla 2. Porcentaje de eficiencia de los algoritmos en los 3 diversos escenarios.

	RRT	RRTConnect	RRTstar	TRRT	LazyRRT
Escenario 1	86%	80%	85%	50%	65%
Escenario 2	91.6%	83.3%	76.6%	83.3%	0%
Escenario 3	96.6%	95%	100%	83.3%	81.6

En la Tabla 3 se muestra la ejecución del algoritmo RRT, el cual tuvo el mejor desempeño en el escenario 1.

Tabla 3. Rendimiento del algoritmo RRT en el escenario 1.

Max Time (sec.)	Collision Checking resolution	Goal bias	Range	Tiempo CPU	Estados Creados	Estados Interpolados	Conectado
60	.010	0.05	0	0.161305	157	121	100%
60	0.030	0.05	0	0.189162	1369	123	100%
60	0.030	0.05	1	5.470463	40621	123	100%
60	0.050	0.05	1	4.553411	37647	123	90%
60	0.030	0.02	0	0.072629	148	127	90%
60	0.060	0.02	0	0.094622	309	106	40%

La Tabla 4 muestra la ejecución del algoritmo RRT-Connect que tuvo uno de los mejor desempeños en el escenario 1.

Tabla 4. Rendimiento del algoritmo RRT-Connect en el escenario 1.

Max Time(sec.)	Collision Checking resolution	Range	Tiempo CPU	Estados Creados	Estados Interpolados	Conectado
60	0.010	0	0.0659405	47	121	80%
60	0.010	1	2.9081836	15876	123	100%
60	0.030	0	0.041726	45	102	70%
60	0.080	2	0.9875299	7608	114	30%
60	0.020	2	0.9399893	6663	123	100%
60	0.018	5	0.3602856	2790	122	100%

Es posible observar que los resultados dependen fuertemente de las configuraciones del algoritmo, puesto que estas determinan que tan eficiente es para realizar la planificación. Los parámetros son diferentes en cada uno de los algoritmos, se puede determinar el rango que tendrá el crecimiento de las ramas, la verificación de colisión, el bias que se quiere lograr, etc. Ciertos algoritmos a pesar de que en determinados escenarios tuvieron una respuesta poco favorable en otros tuvieron buen desempeño, nuevamente la dificultad en el escenario es la que nos permite tener resultados diversos, dando pauta a determinar el mejor algoritmo bajo las configuraciones de cada escenario.

5.4. Pruebas en tiempo real

Finalmente se realizó la combinación de las dos técnicas mencionadas al principio para resolver nuestro objetivo principal. El equipo de pruebas fue el cuadricóptero llamado AR.Drone 2. La intervención del dron para la parte del vuelo autónomo se completó usando el framework TUM_Ardrone y bajo un algoritmo propio que sirvió de intérprete para darle el plan de vuelo. Para la parte de detección y seguimiento de personas, se utilizó la cámara principal del Ar.Drone la cuál utiliza los puertos establecidos del UAV para comunicarse mediante la red local de WiFi y hacer streaming del video capturado.

Las pruebas se realizaron en un escenario que previamente se modeló a escala lo que nos permitió obtener la planificación de movimientos bajo los diversos algoritmos, en dicho escenario no hubo tránsito de personas durante el proceso, es importante mencionar que la eficiencia en las pruebas y resultados son dependientes de variables como la iluminación, el aire, el tránsito peatonal, etc.

En la Figura 5 se muestran las pruebas realizadas del vuelo autónomo.

La tarea de reconocimiento de personas es pesada computacionalmente, ya que entre mayor es la imagen que se recibe, más tiempo tarda en obtener resultados adecuados. Para imágenes muy pequeñas, del orden de 64×128 píxeles, el reconocimiento de una persona tarda en promedio 40 ms. Para la imagen que se procesa desde el UAV, el tiempo que tarda

en hacer el reconocimiento es de 200ms en promedio, por ello se tomaron medidas que ayudaran a la correcta percepción del seguimiento. La Figura 6 muestra la detección y seguimiento de la persona a través del vuelo autónomo del AR.Drone 2.0.

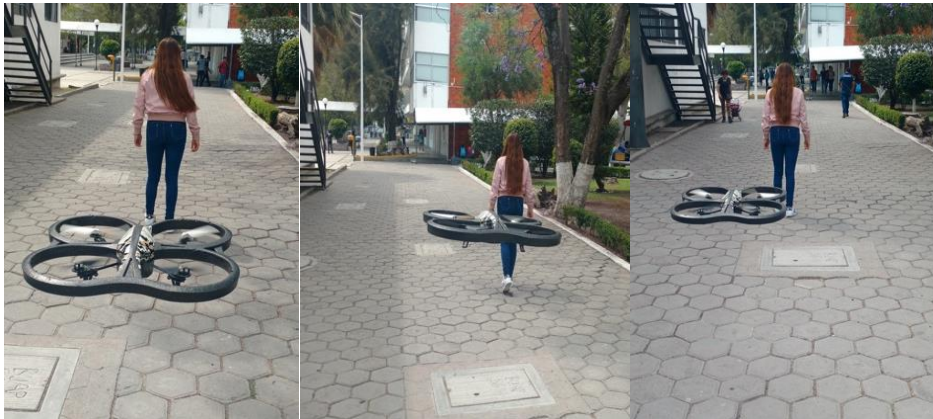


Fig. 5. Vuelo autónomo mediante planificación de movimientos.

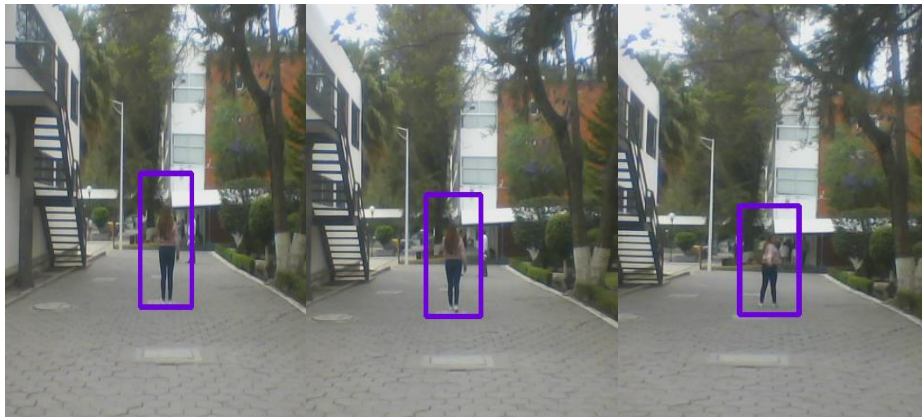


Fig. 6. Seguimiento de personas a través del AR.Drone 2.

6. Conclusiones

El concepto inicial de este artículo fue la elaboración de un sistema capaz de realizar el seguimiento autónomo de personas con un robot aéreo no tripulado. Se ha demostrado que el HOG hace un reconocimiento muy bueno de varias siluetas de personas en la escena, sin embargo, aún falta por acoplar las órdenes que se envían al UAV con el seguimiento del

objeto central de la escena, que si bien con el filtro de partículas fácilmente podemos saber cuál objeto se debe seguir, las órdenes de movimiento aún son un tanto complejas debido a los factores externos muy variables que inherentemente se encuentran con los UAV.

Uno de los aspectos más significativos que saltan a la vista es la dificultad de estabilidad en los movimientos del UAV, sobre todo en condiciones de viento fuerte. Además una de las tareas en la que conviene trabajar fuertemente es en el seguimiento del objeto cuando se encuentran muchas personas (siluetas) u objetos del mismo tipo en la escena, sin perder de vista el objetivo principal.

Este tipo de sistemas nos permiten abrir un gran campo de posibilidades para el área de robótica aérea. El algoritmo de planificación utilizado puede ser mejorado realizando diversas pruebas hasta obtener los parámetros óptimos, de igual forma el reconocimiento y seguimiento de personas es dependiente del entrenamiento que reciba, esto nos da pauta a proponer diversos datasets que nos permitan enriquecer el conocimiento de nuestro algoritmo.

Este proyecto actualmente se sigue desarrollando y verificando mediante ciertas variantes para buscar una ejecución más precisa. Existe un amplio trabajo a futuro puesto que, por ejemplo, se puede utilizar un algoritmo RRT en tiempo real que permita generar nuevas trayectorias dependiendo de los obstáculos imprevistos que se puedan presentar, o bien mejorar la estabilidad del vuelo en corrientes fuertes de aire, de igual forma en el seguimiento de personas es posible hacer un mejor entrenamiento para la obtención de características donde se tomen factores claves como, diferentes tipos de iluminación, escenarios, tipos de segmentación, tamaño de los datasets, entre otros, además de mejorar las características del filtro de partículas para un rendimiento más eficiente del seguimiento.

Referencias

1. Garijo, D., López, J. I., Pérez, I.: Control de un vehículo aéreo no tripulado. Universidad Complutense de Madrid, facultad de informática, pp. 3 (2009)
2. Ibáñez, N.: Técnicas de procesamiento de imagen para el seguimiento de objetos desde vehículos aéreos no tripulados. Universidad de Valladolid, pp. 9 (2015)
3. Núñez-Escamilla, R.: Diseño, Construcción, Instrumentación y Control de un Vehículo Aéreo No Tripulado (UAV). Licenciatura, Instituto Politécnico Nacional, pp. 13 (2010)
4. Escobar-Díaz, O.: Seguimiento de objetos con vehículos aéreos no tripulados. Tesis de maestría, FCC-BUAP, pp. 7, 11 y 12 (2015)
5. Kuffner, J., LaValle, S. M.: RRT-connect: An efficient approach to single-query path planning. In: Proc. 2000 IEEE Intl. Conf. on Robotics and Automation, DOI: 10.1109/ROBOT.2000.844730, pp. 995–1001 (2000)
6. Karaman, S., Frazzoli, E.: Sampling-based Algorithms for Optimal Motion Planning. International Journal of Robotics Research, Vol. 30, No. 7 (2011)

7. Devaurs, D., Siméon, T., Cortés, J.: Enhancing the Transition-based RRT to Deal with Complex Cost Spaces. In: IEEE International Conference on Robotics and Automation, DOI: 10.1109/ICRA.2013.6631158, pp. 4120–4125 (2013)
8. Bohlin, R., Kavraki, L. E.: A Randomized Algorithm for Robot Path Planning Based on Lazy Evaluation, in Handbook on Randomized Computing, pp. 221–249 (2001)
9. Plaku, E., Kavraki, L. E., Vardi, M. Y.: Motion Planning with Dynamics by a Synergistic Combination of Layers of Planning. In: IEEE Transactions on Robotics, DOI: 10.1109/TRO.2010.2047820 (2010)
10. Pedestrian Dataset INRIA. <http://pascal.inrialpes.fr/data/human>
11. Pedestrian Dataset MIT. <http://cbcl.mit.edu/software-datasets/PedestrianData.html>
12. SVMLight. <http://svmlight.joachims.org/>
13. OMPL App. <http://ompl.kavrakilab.org/gui.html>