

Algoritmo para la detección de formas aplicable a la estimación solar

Aguilar-López, J.M.^{a,*}, García, R.A.^a, Camacho, E.F.^a

^aDepartamento de Ingeniería de Sistemas y Automática, Universidad de Sevilla, Camino de los Descubrimientos s/n, 41092 Sevilla, España.

To cite this article: Aguilar-López, J.M., García, R.A., Camacho, E.F. 2020. Two layer algorithm for spatial solar radiation estimation. Revista Iberoamericana de Automática e Informática Industrial 00, 1-5. <https://doi.org/10.4995/riai.2020.7133>

Resumen

En este artículo se presenta un algoritmo híbrido bio-inspirado para la detección de formas aplicado a la estimación solar en plantas solares. Se tiene como objetivo localizar y caracterizar la forma de una nube sobre una planta solar basándose en medidas de niveles bajos de la irradiancia con una pequeña flota de vehículos aéreos no tripulados (UAVs en inglés) equipados con sensores capaces de medir la irradiancia directa normal. El algoritmo híbrido propuesto se inspira y adapta las ideas del algoritmo de optimización de colonia de hormigas (ant colony optimization, ACO) y también usa un algoritmo estándar de cobertura de área, separándose el campo de la planta solar en dos mallados, uno para cada capa del algoritmo, para encontrar el área afectada por la nube. Cuando un UAV localiza la zona de baja irradiancia, los otros van a ayudarle. Dicho equipo delimita el borde de la nube usando conceptos de técnicas de procesamiento de imágenes. Finalmente, se prueba el algoritmo propuesto mediante simulaciones.

Palabras clave: Estimación, Robots móviles, Algoritmo de dos capas

Shape detection algorithm applicable to solar estimation

Abstract

This paper presents a bio-inspired hybrid algorithm for shape detection applicable to solar estimation in solar power plants. The objective is to locate and characterise the shape of a cloud over a solar power plant based on low level irradiance measurement with a small fleet of Unmanned Aerial Vehicles (UAVs) equipped with direct normal irradiance sensors. The hybrid algorithm takes inspiration and adapts ideas of the ant colony optimisation algorithm (ACO) and also uses a standard cover area algorithm, separating the field into two grids, one for each layer of the algorithm, to find the area affected by the cloud. Once the low irradiance zone is located by one of the UAVs, the others go to help it. This team delimits the cloud border using concepts of an image processing technique. Finally, the algorithm is tested by simulations.

Keywords: Estimation, Mobile robots, Two layers algorithm

1. Introducción

En los últimos años, el avance de los vehículos aéreos no tripulados (unmanned aerial vehicle, UAV) o drones, ha incentivado su uso en diferentes aplicaciones, tales como la agricultura (Rokhmana, 2015), las operaciones de salvamento (Silvagni et al., 2017), la prevención de desastres naturales (Yfantis, 2019), o la inspección de grandes áreas con problemas de accesibilidad por tierra (Cesetti et al., 2010). Las principales ventajas de este tipo de vehículos son sus reducidas dimensiones,

maniobrabilidad, velocidad y capacidad de usar una gran variedad de dispositivos adaptados a la aplicación requerida, como cámaras RGB, térmicas o hiperspectrales (Aasen et al., 2015), sensores de temperatura (Sheng et al., 2010), etc. Esta versatilidad los hace adecuados para un gran número de aplicaciones robóticas.

En robótica, una de las líneas de investigación más estudiadas es la conocida como Problema de Cobertura de Áreas o Problema de Cobertura de Camino (Galceran and Carreras, 2013).

*Autor para correspondencia: jaguilarl@us.es

El objetivo es observar un área de interés tan rápido como sea posible, ya sea para tareas de vigilancia o para cualquier otro fin. Este problema es una variación del Problema del Viajante (Travelling Salesman Problem) (Current and Schilling, 1989). Existen numerosas aproximaciones para este problema, de las cuales algunas se basan en el tipo de descomposición del área, como por ejemplo, descomposiciones clásicas en celdas como la de Morse (Acar et al., 2002) o la boustrofedón (Choset and Pignon, 1998); otras se pueden calcular fuera de línea (Oksanen and Visala, 2009) o en línea (Acar and Choset, 2000); y otras tienen en cuenta restricciones energéticas (Choi et al., 2020).

Para mejorar su eficiencia, normalmente los algoritmos para UAVs se desarrollan para sistemas multi-UAVs, es decir, para un equipo de UAVs. Este enfoque necesita algoritmos que controlen individualmente cada uno de ellos además de su formación conjunta (Anderson et al., 2008) (García et al., 2020). La Naturaleza ha sido siempre una fuente de inspiración para la ciencia (Bar-Cohen, 2006), y esta clase de sistemas multiagentes se perfila para el empleo de algoritmos bio-inspirados como la Optimización de Colonia de Hormigas (Ant Colony Optimisation ACO). Dicho algoritmo lo presentó en 1991 Dorigo (Dorigo, 1991) como solución al Problema del Viajante (Johnson and McGeoch, 1997). Es un algoritmo bio-inspirado donde se crean hormigas virtuales que se mueven entre las distintas ciudades, representadas mediante nodos. Las hormigas dejan feromonas en los arcos que conectan los nodos. Al igual que las hormigas reales, las virtuales eligen su próximo nodo de destino basándose en un método probabilístico según la cantidad de feromonas que se ha dejado en el arco que conecta los nodos y la distancia entre ellos. Finalmente, tras cada iteración del algoritmo, una cantidad de las feromonas se evapora en todos los arcos.

El algoritmo que se desarrolla en este trabajo ha sido probado en un área de dimensiones similares a las que tendría una típica planta solar cilindroparabólica. Estas plantas ocupan grandes extensiones de terreno, normalmente 100 ha para plantas de 50 MW (Sánchez et al., 2019). La eficiencia de una planta de este tipo puede verse tremendamente afectada por las sombras que causan las nubes (Ashley et al., 2017). En (Camacho et al., 2012; Camacho and Gallego, 2013) se muestra que es necesario saber la irradiancia directa normal (IDN o DNI en inglés) para conseguir un control eficiente de la planta solar. Debido a que las plantas solares se extienden en grandes áreas (hasta 800 ha), algunas partes del campo solar pueden estar recibiendo mucha menos IDN por culpa del efecto de las nubes. Conocer la IDN en diferentes partes de la planta permite mejorar mucho el control de la misma (Xu et al., 2015). Esta IDN ha sido objeto de estudio en diferentes trabajos para mejorar la eficiencia de una planta, como (Nouri et al., 2018), o (Kuhn et al., 2017), donde se usan cámaras que apuntan al suelo para estimar la IDN. La principal desventaja de este tipo de sistemas es la necesidad de construir torres para la colocación de las cámaras y que, al ser una medida indirecta de la IDN obtenida a través de la reflectividad del suelo, se precisa de un modelo que depende de factores como la temperatura, humedad, la orografía, etc., algunos de los cuales pueden variar a lo largo del año, lo que obligaría a reajustar el modelo. En (Sánchez et al., 2018) se demuestra que con un control de los lazos adecuado mediante válvulas se pueden obtener beneficios en la produc-

ción de energía, que se traducen en beneficios económicos. En el presente trabajo, para llevar a cabo la tarea de medir la IDN en varios puntos de la planta, se propone el uso de una flota de UAVs, equipado cada uno con un sensor liviano, de bajo consumo capaz de medir la IDN (Technologies, 2019).

Este artículo presenta un novedoso algoritmo híbrido para la cobertura de áreas que combina un algoritmo inspirado en las ideas del algoritmo ACO para el problema de planificación de caminos y el algoritmo de boustrofedón para la cobertura del área. El objetivo del algoritmo es detectar zonas con baja irradiancia por efecto de las nubes, y usar las medidas de la IDN en el campo para definir la forma de la sombra de la nube y para la estimación de la irradiancia espacial, en contraposición a lo presentado en otros trabajos recogidos en (Law et al., 2014). La principal contribución de este trabajo respecto a otros, como por ejemplo (Xiong et al., 2019), es su carácter híbrido y su aplicación a la detección de áreas con poca irradiancia en plantas solares.

El resto del artículo se organiza como se indica a continuación. En la Sección 2 se presentan la notación y los conceptos preliminares relacionados con el algoritmo ACO y la descomposición del terreno. En la Sección 3 se describe el problema que se ha considerado. En la Sección 4 se muestra el algoritmo híbrido propuesto, que se pone a prueba por simulación en la Sección 5. Finalmente, las conclusiones se detallan en la Sección 6, donde también se remarcan ciertas consideraciones.

2. Notación y preliminares

2.1. Algoritmo ACO original

Como se ha mencionado en la sección anterior, este algoritmo fue inicialmente propuesto por Dorigo (Dorigo, 1991). A continuación, se describe el algoritmo particularizado para la resolución del Problema del Viajante. Un conjunto de hormigas $B = \{h_1, h_2, \dots, h_i, \dots, h_H\}$ tienen que conectar el hormiguero, que es el nodo de partida, al lugar con la comida, que es el nodo de destino. Ambos nodos pertenecen al conjunto de nodos $V = \{n_1, n_2, \dots, n_i, \dots, n_N\}$. Las hormigas completan su misión viajando a través de los arcos que conectan los nodos, depositando feromonas cuando dejan dichos arcos. Las hormigas eligen de forma probabilística cuál será el siguiente nodo que visitan, según la cantidad de feromonas que haya en el arco de unión entre su nodo y los nodos conectados a él y según la distancia entre los nodos.

Sea d_{ij} la distancia entre el nodo i y el nodo j , se define v_{ij} la visibilidad del nodo i desde el nodo j para el caso del Problema del Viajante como:

$$v_{ij} = \frac{1}{d_{ij}} \quad (1)$$

La probabilidad de que una hormiga h se mueva desde el nodo i al nodo j siendo τ_{ij} la cantidad de feromonas en el arco que conecta el nodo i con el nodo j es:

$$p_{ij}^h = \frac{[\tau_{ij}(k)]^\alpha [v_{ij}]^\beta}{\sum_{g \in J_i^h} [\tau_{ig}(k)]^\alpha [v_{ig}]^\beta}, \quad \forall j \in J_i^h, \alpha \geq 0, \beta \geq 0, \quad (2)$$

donde J_i^h son los nodos no visitados por la hormiga h desde el nodo i . Los parámetros α y β ajustan el peso relativo de la

heurística en el cálculo de la probabilidad. El caso de $\alpha = 0$ es el típico algoritmo codicioso o voraz, donde los nodos más cercanos son los más probables de visitar (Zhang et al., 2000), mientras que en el caso $\beta = 0$ solo se considera el peso de las feromonas.

La evaporación de las feromonas en cada iteración se calcula con Eq. (3):

$$\tau_{ij}(k+1) = (1 - \rho) \tau_{ij}(k) + \sum_{h=1}^H \Delta\tau_{ij}^h(k), \quad (3)$$

siendo ρ la variable que ajusta el grado de evaporación y $\Delta\tau_{ij}^h(k)$ la cantidad de feromonas que deposita la hormiga h en el arco ij en el instante k .

Finalmente, el algoritmo ACO usa una función objetivo que hay que maximizar o minimizar según el tipo de problema. Particularmente, en el Problema del Viajante la función es:

$$\text{mín} \sum_{i=0}^N \sum_{j \neq i, j=0}^N c_{ij} x_{ij}. \quad (4)$$

siendo c_{ij} el coste de viajar por el arco que conecta los nodos i y j , y x_{ij} una variable binaria que indica si se ha viajado por el arco ij .

En resumen, el algoritmo ACO consta de los siguientes pasos:

1. Las hormigas empiezan en nodos aleatorios.
2. Cada hormiga h crea un camino solución P_h entre su nodo de partida y el de destino visitando los otros nodos y eligiendo el arco que los conecta de forma probabilística, según Eq. (2).
3. Se evalúan todos los caminos solución P_h con la función objetivo Eq. (4) y cada hormiga h deja una cantidad de feromonas en los arcos que ha recorrido proporcional a la calidad de su solución.
4. El mejor camino solución encontrado en esta iteración P_h se compara con el mejor camino solución guardado, y si lo supera, lo sustituye.
5. Se evapora la cantidad de feromonas de cada arco ij según Eq. (3).
6. Se empieza de nuevo desde el paso 1.

Para decidir si el último camino solución almacenado es una solución aceptable, se establece una condición de fin que se evalúa después de cada iteración, ya que no hay otra forma de saber si la solución encontrada es el máximo o mínimo absoluto.

2.2. Descomposición del campo

Se divide el terreno en dos capas, una para el algoritmo inspirado en ACO propuesto y otra para el algoritmo de cobertura de área. Usando la terminología de jerarquías, el mallado para el primer algoritmo se denomina malla superior y el mallado para el segundo malla inferior. La Figura 1 muestra ambas mallas: la malla superior se representa en azul y la inferior en rojo. La malla superior del terreno se divide en un número entero de celdas del mismo tamaño, 10×10 en este caso, siendo cada una de 100×100 metros. Esta descomposición se elige tras comprobar que una división mayor en el mallado superior no aprovecharía

las ventajas del algoritmo de cobertura de área, y uno más pequeño no aprovecharía las ventajas del algoritmo inspirado en ACO, incrementando el tiempo de búsqueda final.

De igual forma, la malla inferior también se divide en un número entero de celdas. El detalle de esta descomposición puede observarse en la Figura 1, donde se observa una división de 5×5 celdas, siendo cada una de 20×20 metros. La descomposición de la malla inferior mostrada es diferente a la que se usa en el algoritmo para que se pueda distinguir en la figura, ya que en las simulaciones realizadas cada celda es de 1×1 metros, dando, por tanto, una malla de 1000×1000 celdas.

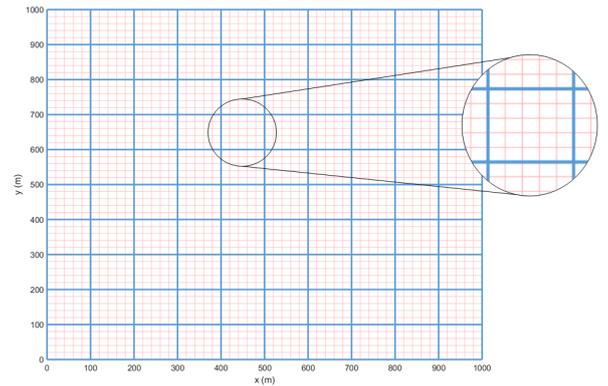


Figura 1: Esquema del mallado de las dos capas usadas en el algoritmo propuesto.

2.3. Sistema multi-UAVs

El modelo de comunicación del sistema multi-UAVs utilizado es el de comunicación centralizada: los UAVs no se comunican directamente entre sí, sino que envían sus medidas y reciben órdenes del ordenador de la estación base, que es el que procesa los datos y calcula las nuevas rutas. La principal desventaja de este tipo de comunicación es la necesidad de un canal seguro de comunicación entre los sistemas que lo componen. La aplicación de esta propuesta está orientada a plantas solares que generalmente tienen una extensión de 1 km^2 , permitiendo las redes Wi-Fi actuales garantizar esta comunicación en estas distancias y a una velocidad adecuada.

Otro de los puntos a tratar cuando se trabaja en un sistema multi-robot como el propuesto es cómo evitar la colisión entre los distintos robots del sistema. La solución general pasa por el uso de sensores junto a una buena comunicación entre los robots o algoritmos de generación de trayectorias que tengan esto en cuenta. En el problema presentado, donde se usan UAVs, se pueden evitar fácilmente las colisiones estableciendo distintas alturas de seguridad para cada uno de los UAVs.

Para poder medir la irradiancia solar, los UAVs disponen, como ya se mencionó en la Sección 1, de un sensor ligero y de bajo consumo, concretamente de 10 gramos de peso y un campo de visión de hasta 120° . Aunque este trabajo es teórico y se prueba en simulación, cuando en futuros trabajos se pruebe experimentalmente, se garantizará que la medida no se vea afectada por una mala orientación del campo de visión del sensor utilizando un Gimbal que lo oriente en los ángulos deseados. La conjunción del gran campo de visión del sensor y del Gimbal permitirá obtener buenos resultados aún con pequeños desplazamientos o vibraciones de los UAVs.

3. Planteamiento del problema

En esta sección se describe el problema propuesto. El objetivo del equipo de UAVs es localizar y caracterizar la forma de una nube C para mejorar la eficiencia de una planta solar situada en un área delimitada A .

El equipo está formado por un conjunto de UAVs denotados como $U = \{u_1, u_2, \dots, u_i, \dots, u_H\}$. Como se verá luego, cada UAV requiere su propia hormiga virtual y, por tanto, el número de hormigas es igual al número de UAVs. El área de interés A es un rectángulo de w por l .

Las siguientes consideraciones se han tenido en cuenta para la solución propuesta, que es un primer acercamiento al problema planteado. Se aproxima la nube C por una elipse con semieje mayor a , semieje menor b , centro O y ángulo de rotación ϕ . Véase la Figura 2. Dado que el objetivo es detectar nubes de tamaño suficiente para que afecten a la eficiencia de la planta solar, se establece un semieje mayor y menor mínimo de 20 metros. Se supone que la nube se mueve tan despacio que puede considerarse estática. La reducción de la irradiancia solar provocada por la nube C se modela como una función lineal, véase la Figura 3. C se divide en tres subregiones con diferente factor de ocultación, como se aprecia en la Figura 4. Estas subregiones también son elípticas y mantienen la misma excentricidad que C pero tienen diferente semieje mayor.

En futuros trabajos muchas de estas restricciones se verán relajadas para abarcar un espectro más amplio de situaciones, como nubes no convexas o en movimiento, que serán comentadas en la Sección 6.

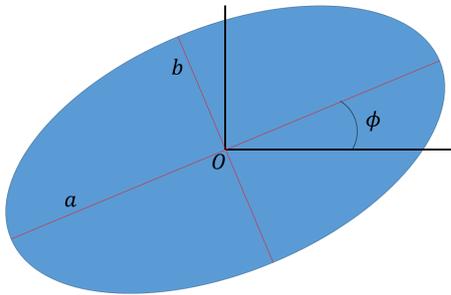


Figura 2: Forma de la nube supuesta.

4. Solución propuesta

El algoritmo híbrido propuesto se descompone en dos fases:

1. **La búsqueda de la nube:** el objetivo de esta fase es encontrar una nube en el área A . Para ello, los UAVs miden continuamente la irradiancia solar, que decaerá bajo la influencia de la sombra de una nube. Esta primera fase se divide, a su vez, en dos etapas:
 - a) *Etapa del algoritmo inspirado en ACO:* cada UAV decide cuál será la siguiente celda de la malla superior a la que se dirigirá (ver Subsección 4.1).
 - b) *Etapa de cobertura de la celda:* cada UAV barre la celda de la malla superior escogida visitando parte de las celdas de la malla inferior contenidas en ella (ver Subsección 4.2).

Si uno de los UAVs toma una medida por debajo de un determinado valor umbral, se considerará que se ha encontrado una nube y concluirá esta fase; en caso contrario, se continúa alternando entre estas dos etapas. Se notificará este hecho a los otros UAVs para que se dirijan a esta posición y se pase a la siguiente fase. En caso de que en el camino a dicha posición detecten otro punto distinto de la nube, se quedarán en esa posición para continuar con la siguiente fase desde ahí.

Este valor umbral no es un valor fijo, y se puede variar durante la ejecución del algoritmo. Es un valor de referencia que se escoge basándose en la IDN del cielo despejado y el error de los sensores. El valor de la IDN se puede conocer tanto por datos meteorológicos de años anteriores como a través de una medida de un sensor que apunte directamente al sol, pudiéndose usar el mismo sensor que montan los UAVs u otro más preciso.

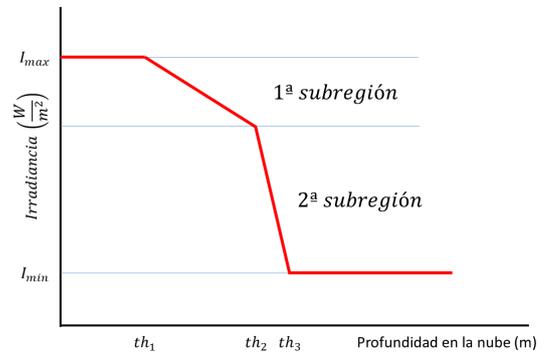


Figura 3: Función de irradiancia supuesta, midiendo *Profundidad de la nube* la distancia desde el borde de la nube hasta su centro, es decir, *profundidad* = 0 es el borde de la nube.

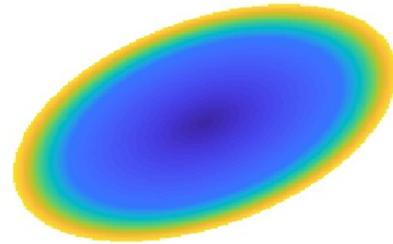


Figura 4: Densidad de la nube supuesta. Hay dos subregiones dentro de la nube: la primera es la mostrada en azul claro y la segunda en azul oscuro. El centro de la nube, por ser la zona más profunda, tiene el valor mínimo de irradiancia I_{min} y es por ello la zona más oscura. El valor máximo de irradiancia I_{max} está representado por el color amarillo anaranjado del borde.

2. **Determinar la forma de la nube:** en esta fase los UAVs encuentran el borde de la nube y lo recorren. Primero calculan el gradiente de irradiancia solar. Después se mueven siguiendo el vector perpendicular a dicho gradiente, es decir, a lo largo de la isolinia de irradiancia solar, corrigiendo el vector de movimiento poco a poco cuando

detectan que se salen de la nube a través de las medidas de irradiancia solar tomadas. Con este movimiento a través de la isolínea se consigue delimitar el contorno de la nube, y con ello, su forma.

Esta fase se divide en tres etapas:

- Medida en espiral cuadrática de la irradiancia solar:* al principio no se puede calcular el gradiente porque solo se ha medido un punto de la nube. Por ello, cada UAV describe una espiral cuadrática (como la espiral de Ulam (Daus, 1932), véase la Figura 7) para tomar suficientes medidas y poder calcular el gradiente (ver Subsección 4.3).
- Cálculo del gradiente de irradiancia solar:* para calcular el gradiente se usan las medidas tomadas en la etapa anterior con el operador Sobel, una técnica de píxeles de procesamiento de imágenes y de visión por computador (ver Subsección 4.4).
- Movimiento a lo largo de la isolínea de irradiancia solar:* los UAVs se mueven siguiendo el vector perpendicular al gradiente calculado, manteniendo un sentido antihorario, aunque hubiera sido indiferente haberlo hecho en sentido horario. En el momento en el que detecten que están fuera de la zona de influencia de la nube, girarán suavemente hacia la izquierda los grados necesarios para volver a encontrar el borde de la nube (ver Subsección 4.5).

En las siguientes subsecciones se describen cada una de estas fases y etapas.

4.1. Algoritmo inspirado en ACO para cobertura de áreas

El problema abordado en este trabajo no usa el algoritmo ACO como una optimización tal y como se ha visto en la Sección 2, sino que se inspira en sus ideas para resolver la cobertura de área. Para ello se han hecho algunas modificaciones a la adaptación del comportamiento de las hormigas presentadas en el algoritmo original. Cabe señalar que la descomposición en celdas del mallado superior es el equivalente a los nodos del grafo asociado a un problema de optimización resuelto con un algoritmo ACO. En la explicación del algoritmo propuesto se hablará de celdas y nodos indistintamente.

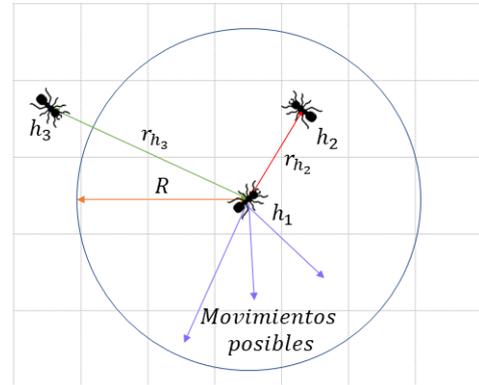
Para empezar, para una cobertura rápida del terreno, las hormigas deben ser repelidas por las feromonas, en lugar de ser atraídas por ellas, haciéndose así más probable que se visiten nodos donde haya menos feromonas. Por tanto, la probabilidad de escoger un nodo j viene dada por:

$$p_{ij}^{\tau} = 1 - \frac{\tau_j}{\sum \tau_g}, \quad (5)$$

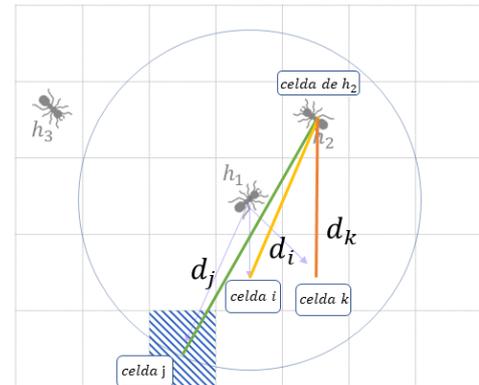
donde $\sum \tau_g$ es la cantidad de feromonas de todos los nodos.

Otro cambio importante es el que se ha realizado en el criterio de la distancia. En el algoritmo ACO original, para el Problema del Viajante, se trata con el cálculo de la visibilidad Eq. (1): el nodo más lejano tiene menos probabilidad de ser escogido. En el caso de la cobertura de área, el objetivo es visitar cada celda al menos una vez, sin importar la distancia que haya que recorrer. Por tanto, en este caso, el criterio de la distancia no se aplica. Alternativamente, para cubrir el área máxima posible, es una ventaja que las hormigas no estén cercanas entre sí, por lo

que el criterio de distancia se sustituye por un criterio de distancia mínima con respecto a otras hormigas. Esto posibilita que las hormigas se repelan entre sí y no visiten a la vez el mismo nodo.



(a) Círculo umbral R: se evita la hormiga h_2 mientras que la hormiga h_3 no es tomada en cuenta ya que se encuentra fuera del círculo R. Las flechas azules muestran algunos movimientos posibles para evitar a la hormiga h_2 .



(b) A efectos de claridad en la representación solo se muestran algunas distancias d , aunque el algoritmo las evalúa todas. Primero se calculan las distancias entre la celda a evitar, la de h_2 , y las candidatas (i, j, k). A continuación, se le asigna a cada celda una probabilidad inversamente proporcional a su distancia d respecto a la celda a evitar. La celda rayada es la más probable para ser escogida puesto que está más lejos de la celda de h_2 .

Figura 5: Criterio de distancia con otras hormigas.

El criterio de distancia con respecto a otras hormigas funciona de manera similar al criterio de las feromonas. Solo se aplica a aquellas hormigas h_j que estén por debajo de una cierta distancia umbral, es decir, las que se encuentran dentro del círculo de radio R . Sea una hormiga h_1 en el centro de un círculo de radio R separada de otras hormigas h_2 y h_3 una distancia r_{h_2} y r_{h_3} respectivamente. En la Figura 5(a) se puede observar que si $r < R$, entonces la hormiga en cuestión se evita, en el ejemplo h_2 , mientras que en otro caso se ignora, en el ejemplo h_3 . Una nueva probabilidad se asigna a cada celda, que será mayor cuanto más lejos esté de la hormiga h_2 :

$$p_{ij}^D = \frac{d_j}{\sum d_l}, \quad (6)$$

siendo d_j la distancia entre la celda candidata, celda j , y la celda ocupada por la hormiga h_2 , como se muestra en la Figura

5(b). Además, d_l es la distancia entre la celda de h_2 y cualquier otra celda candidata por lo que $\sum d_l$ el sumatorio de las distancias entre la celda ya ocupada por h_2 y todas las posibles. En caso de que hubiera más de una hormiga a evitar, se calcularía independientemente cada p_{ij} respecto de cada una de ellas y se aplicaría la ecuación (7):

$$p_{ij}^D = \frac{p_{ij}^{D_1} p_{ij}^{D_2}}{\sum p_{ij}^{D_1} p_{ij}^{D_2}} \quad (7)$$

Finalmente, la probabilidad de que la hormiga h_1 escoja una celda como nuevo destino se obtiene combinando el criterio de las feromonas y el criterio de distancia con otras hormigas, según la ecuación:

$$p_{ij}^{h_1} = \frac{p_{ij}^\tau p_{ij}^D}{\sum p_{ig}^\tau p_{ig}^D} \quad (8)$$

El algoritmo propuesto elige una celda del mallado superior como siguiente posición de h_1 una vez calculadas todas las probabilidades de las celdas candidatas.

4.2. Algoritmo de cobertura de celdas

Una vez escogida la nueva celda del mallado superior, el algoritmo de cobertura crea un camino compuesto de puntos (*waypoints*) para recorrer las celdas del mallado inferior contenidas en ella. Hay que destacar que no se recorren todas las celdas del mallado inferior, ya que solo se quieren detectar nubes mayores de un determinado tamaño mínimo, parámetro que se tiene en cuenta para diseñar el recorrido que barre las celdas. Se diseñaron dos tipos de caminos de cobertura:

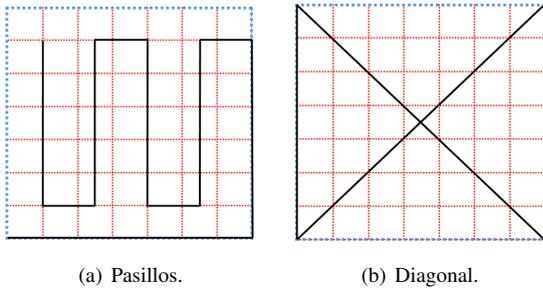


Figura 6: Caminos de cobertura usados en el algoritmo híbrido propuesto.

1. **Pasillos:** el UAV se mueve desde un lateral de la celda al otro, se desplaza entonces perpendicularmente una distancia concreta, y vuelve al lateral desde el que empezó. Esto se repite hasta que se cubra completamente la celda del mallado superior, trazando un camino que se asemeja a varios pasillos. Este tipo de movimiento se conoce como boustrofedón, explicado con más detalle en la Subsección 5.2. El ancho del pasillo, o paso, se fija según el tamaño mínimo de nube que se quiere detectar. Como en este caso el mínimo de los semiejes es de 20 metros (ver Sección 3), ese será el ancho del pasillo. Un ejemplo de este tipo de camino se muestra en la Figura 6(a). Este camino tiende a ser generalmente más lento pero también

más exhaustivo que la otra propuesta, el camino diagonal. A pesar de todo, la velocidad depende en gran medida del tamaño de la celda y el paso que se use, siendo más rápido cuanto más pequeña sea la celda o cuanto más ancho sea el paso.

2. **Diagonal:** este camino solo recorre las diagonales de la celda y dos de los lados laterales, por lo que normalmente es más corto que el camino de pasillos, ya que se visitan menos celdas del mallado inferior, aunque depende del ancho del pasillo escogido. No obstante, deja sin visitar una superficie de la celda considerable.

Observación: tras probar ambos tipos de camino se ha descartado el camino tipo diagonal debido a la superficie sin explorar que deja este camino.

Cuando se termina de explorar la celda del mallado superior, se vuelve al algoritmo inspirado en ACO para elegir una nueva celda y cubrirla con un nuevo camino. Este proceso se repite hasta que algún UAV detecte un punto con baja irradiancia, es decir, un punto de una nube, y llame a los demás para empezar la fase que definirá la forma de la nube.

4.3. Medida de la irradiación solar mediante la espiral cuadrática

Cuando se detecta la nube solo se conoce un punto de ésta, insuficiente para calcular el gradiente de la irradiación solar. Para obtener más medidas se usa un camino en forma de espiral cuadrática como la que se ve en la Figura 7. Cuanto mayor sea la espiral, mejor será el cálculo del gradiente, aunque también hará más lento el algoritmo pues al UAV le lleva más tiempo recorrer el camino. Se establece así una relación de compromiso entre el la calidad del cálculo y la velocidad del algoritmo. Para determinar el número de giros utilizados, se han realizado 1000 simulaciones de medidas de espirales de hasta 5 giros con un 3 % de ruido, mostrándose los resultados en la Tabla 1, siendo el error la desviación en grados sexagesimales del vector gradiente obtenido respecto del vector gradiente real, que es el que se obtendría si se conociera la irradiancia exacta o la función del factor de ocultación de la nube.

Tabla 1: Simulaciones de la espiral.

Número de giros	Error (°)	Tiempo (s)
1	43.68	1.05
2	19.15	3.2
3	9.61	6.12
4	6.17	10.31
5	4.13	15.47

A raíz de los resultados, se fijó el número de giros en 3, dado que 10° es una desviación asumible. Este valor de 10° se determinó a través de pruebas de ensayo y error con el algoritmo. En cualquier caso, este número de giros podría determinarse en ejecución en función de la distancia a la planta, dado que si está más lejos de la misma podría emplear más tiempo para obtener más precisión.

4.4. Gradiente de irradiancia solar

El gradiente de irradiancia solar se calcula usando el operador Sobel, un algoritmo de píxeles que se utiliza en el procesamiento de imágenes en el campo de la visión por computador, desarrollado por Irwin Sobel y Gary Feldman en 1968 (Sobel and Feldman, 1968). Es una diferenciación discreta usada principalmente para detectar bordes en las imágenes.

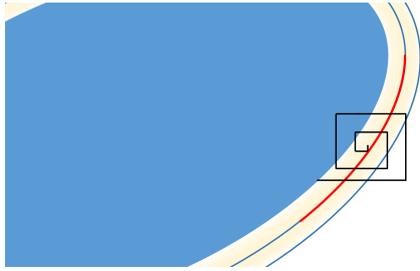


Figura 7: Camino de espiral cuadrática usado para tomar medidas de la nube.

El algoritmo usa dos kernels 3×3 , uno para la dirección horizontal y otro para la vertical. Las ecuaciones Eq. (9a) y Eq. (9b) muestran los kernels que se usan en el algoritmo de Sobel. Posteriormente, Prewitt, Scharr o Roberts Cruz propusieron otros kernels alternativos para obtener resultados diferentes (Savant, 2014). En cualquier caso, en este trabajo solo se han utilizado los originales.

$$K_X = \begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix}, \quad (9a)$$

$$K_Y = \begin{pmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{pmatrix}. \quad (9b)$$

Como se ve en Eq. (9a) y Eq. (9b), la aproximación discreta de las derivadas en los ejes x y y se calcula convolucionando los kernels con el mapa de irradiación solar I . Este mapa se puede considerar como una imagen de tantos píxeles como celdas del mallado inferior se hayan visitado en el camino de espiral cuadrática, siendo así la intensidad de la irradiación en cada celda la intensidad de color del píxel correspondiente en la imagen. Cada píxel en I se puede etiquetar de dos maneras:

- **Evaluado:** uno de los UAVs ha visitado la celda del mallado inferior equivalente y ha medido su valor de irradiancia. Este valor es un *float* positivo de dos decimales.
- **Estado desconocido:** ningún UAV ha visitado la celda del mallado inferior correspondiente. Su valor es -1.

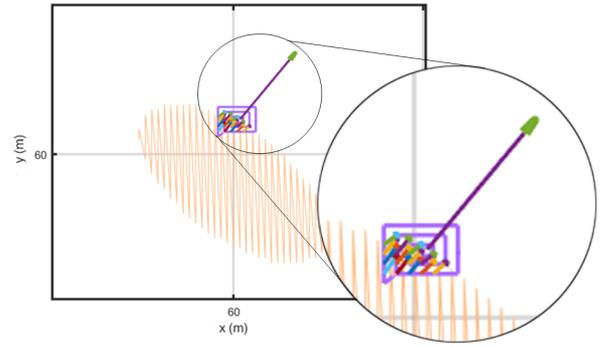


Figura 8: Campo de gradientes y el vector gradiente medio resultante después de la exploración de la nube con el camino de espiral cuadrática. El módulo del vector gradiente medio está amplificado a efectos de una representación clara.

En general, antes de comenzar el barrido en espiral cuadrática, los píxeles de I están marcados con -1 , salvo por aquellas localizaciones que previamente hayan sido visitadas por un UAV durante las fases anteriores del algoritmo. Una vez que acaba el recorrido en espiral, todos los píxeles han sido evaluados y están listos para que se les aplique el operador de Sobel.

El gradiente de irradiancia solar se obtiene con Eq. (10a) y Eq. (10b).

$$I_x = \frac{\delta I}{\delta x} = K_X * I, \quad I_y = \frac{\delta I}{\delta y} = K_Y * I, \quad (10a)$$

$$\nabla I = \begin{bmatrix} I_x \\ I_y \end{bmatrix}. \quad (10b)$$

Usando el operador Sobel sobre todo el mapa I se obtiene un campo de gradientes. Calculando la media de todos los vectores gradiente del campo se obtiene el vector gradiente medio desde la posición inicial en el borde de la nube, como se muestra en la Figura 8. Este vector se usa para determinar la dirección de movimiento del UAV.

4.5. Movimiento siguiendo la isolínea

Teniendo el gradiente resultante, la isolínea de irradiancia solar está en la dirección perpendicular al mismo y es la que sigue el UAV en sentido antihorario para recorrer el borde de la nube.

Durante este movimiento el vector de trayectoria se vuelve rápidamente impreciso ya que la isolínea es elíptica. Este error se corrige continuamente, ya que, cuando el UAV detecta que está fuera de la nube, a través de una medición de la irradiancia por encima del valor umbral, gira su trayectoria suavemente en sentido antihorario para volver a la nube. De esta manera la forma de la nube se delimita paso a paso. Cabe señalar que cuanto más lento se muevan los UAVs menos tardarán en corregir la trayectoria cuando se salgan de la nube, pero también les llevará más tiempo definir la forma completa de la misma.

La Figura 9 muestra el borde que se obtiene de la nube y también cómo los 3 UAVs modifican su trayectoria con el procedimiento explicado anteriormente. Se puede apreciar que las líneas son un poco onduladas debido a la corrección descrita.

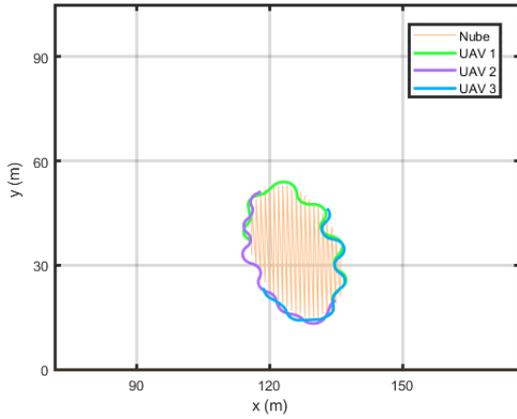
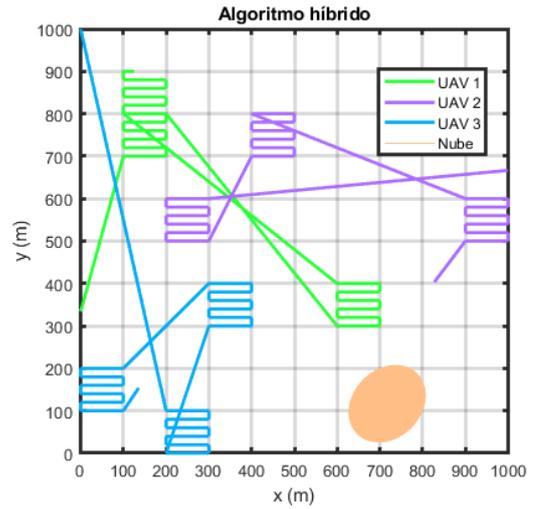
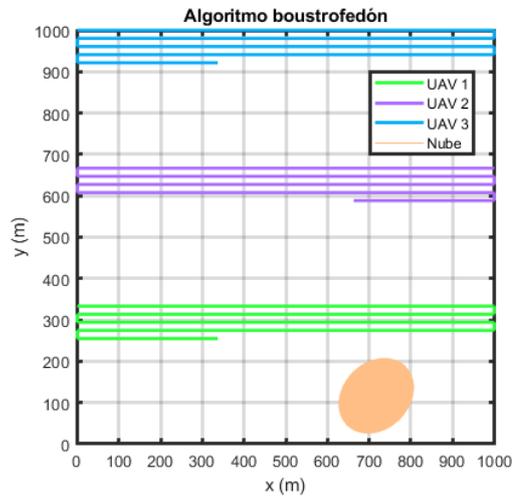


Figura 9: Forma de la nube definida por los 3 UAVs (azul, verde, morado) aplicando el algoritmo propuesto.



(a) Ejemplo de búsqueda de la nube usando el método híbrido propuesto.



(b) Ejemplo de búsqueda de la nube usando el movimiento boustrofedón.

5. Simulaciones

Para probar el método híbrido propuesto se diseñaron una serie de simulaciones. Éstas fueron programadas y ejecutadas en el programa MATLAB versión R2020a. Para una mejor representación gráfica de los resultados las figuras de esta sección muestran un área y una nube que son más pequeñas que las usadas en las simulaciones, pero proporcionales a éstas.

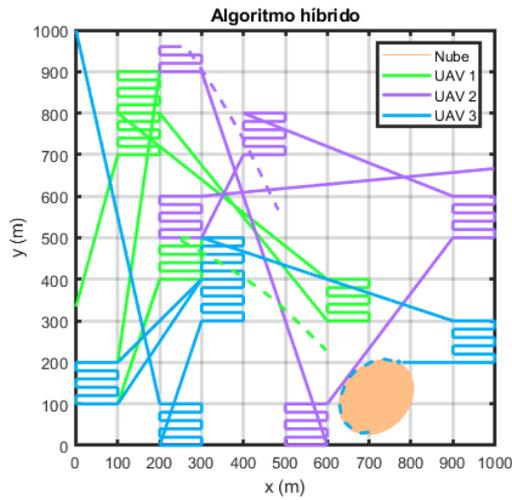
Figura 10: Búsqueda de la nube usando los dos algoritmos.

5.1. Condiciones de las simulaciones

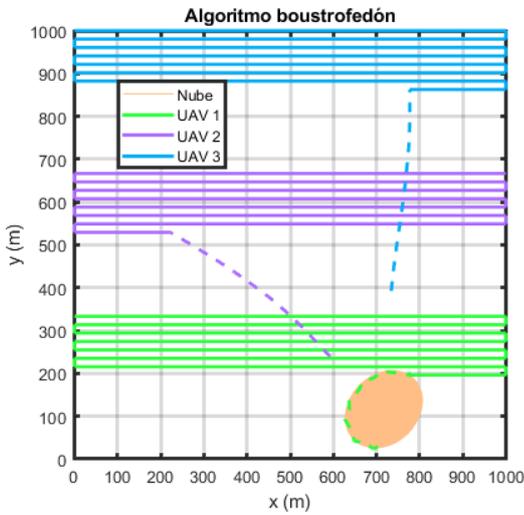
Cada simulación es un escenario de búsqueda de la nube en un área de 1000x1000 metros. La descomposición de la superficie fue de 10x10 celdas para el mallado superior (algoritmo ACO), que es visible en las figuras, y de 1000x1000 celdas para el mallado inferior (movimiento boustrofedón), que no se muestra en las figuras por su pequeño tamaño.

El equipo de UAVs está formado por 3 vehículos, y cada uno de ellos se mueve a 8 m/s sin restricciones de movimiento ni de energía. El valor de la velocidad se basa en la velocidad media de UAVs comerciales como (DJI Technology Inc., 2015). Sus posiciones iniciales, en metros, son: (0, 333.33), (1000, 666.66) y (0, 1000).

La nube se modela como una elipse de 100 metros de semieje mayor y 80 metros de semieje menor. En cada simulación, los parámetros que varían son la posición y el ángulo de giro de la nube, que se generan aleatoriamente cada vez.



(a) Algoritmo híbrido



(b) Algoritmo boustrofedón.

Figura 11: Cuando se encuentra la nube se llama al resto de UAVs. Las líneas continuas representan el camino que se ha seguido antes de encontrar la nube, mientras las discontinuas muestran el camino hacia la nube tras la llamada.

Además, cada escenario se usa para evaluar tanto el algoritmo híbrido como el algoritmo boustrofedón clásico, ya que éste es uno de los más usados en la literatura para abordar el problema de cobertura de área, como se menciona en la Sección 1.

5.2. Algoritmo boustrofedón

Los algoritmos de tipo boustrofedón son los que están inspirados en un tipo de textos bidireccionales, esencialmente textos antiguos, en los que la dirección de la escritura de cada línea es contraria a la dirección de la línea anterior, es decir: la primera línea se escribe de izquierda a derecha, la segunda de derecha a izquierda y así sucesivamente. Este mismo procedimiento se aplica para generar caminos que cubran un área, pues asegura que todas las posiciones del área se visiten dentro de un cierto tiempo usando un camino bastante eficiente sin demasiados giros, como puede verse en la Figura 10(b). Algunos algoritmos

inspirados por este tipo de movimiento se pueden ver en (Combes et al., 2017), (Ntawumenyikizaba et al., 2012), (Jin and Tang, 2010) or (Avellar et al., 2015).

El camino de pasillos del método híbrido es un ejemplo de este tipo de algoritmos boustrofedón, como se ve en la Figura 10(a). Es importante subrayar que la implementación del algoritmo boustrofedón utilizada aquí para compararse con el algoritmo híbrido no visita todas las celdas del área por la razón explicada en la Subsección 4.2, esto es, solo las nubes de un determinado tamaño mínimo o superior son de interés. Por esta razón, este algoritmo es, en realidad, como si se usara el camino de pasillos que se usa para cubrir una celda del mallado superior, pero considerando toda el área como una única celda. El ancho de separación entre pasillos es el mismo que se usa en el algoritmo híbrido, es decir, 20 metros.

Además, para evitar solapamientos, cada UAV tiene asignado un tercio del área, empezando su exploración en una esquina, de forma que se optimiza su movimiento y búsqueda en el área asignada. Como en el caso del algoritmo híbrido, los UAVs que aplican este algoritmo están continuamente midiendo la irradiancia para detectar la nube. En la Figura 11 se ve cómo, gracias a esta continua medición, al encontrar uno de los UAVs una medida por debajo del umbral los demás son llamados para continuar con la fase de descripción de la forma de la nube.

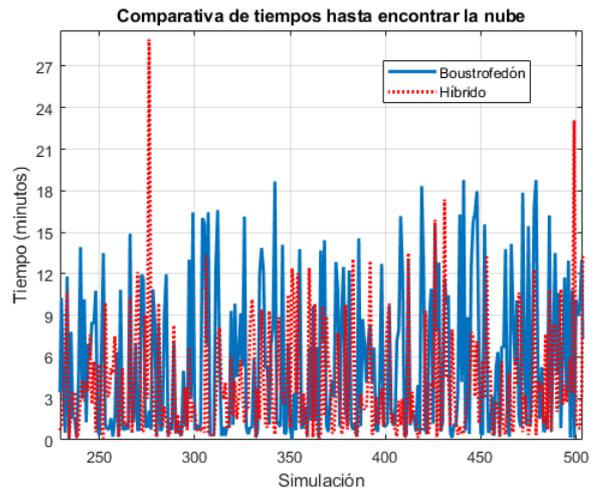


Figura 12: Comparativa de los resultados de las simulaciones de ambos algoritmos.

5.3. Resultados

Los resultados obtenidos tras ejecutar 1000 simulaciones se muestran en la Tabla 2. Cabe señalar que se compara la velocidad en la detección de la nube, y no su posterior caracterización, dado que para esa fase ambos algoritmos emplean el mismo método.

El algoritmo híbrido propuesto obtiene mejores resultados que el boustrofedón en términos generales, aunque con margen de mejora: le lleva un 21 % menos de tiempo de media encontrar la nube. Es importante señalar la existencia de valores atípicos, como se puede ver en la Figura 12. Este tipo de valores empeoran el valor de la media, pero la mediana es bastante más robusta a ellos, razón por la que se recoge en la Tabla 2. Un

Tabla 2: Resultados de las simulaciones.

Algoritmo	Tiempo medio	Mediana del tiempo	Victorias
<i>Boustrofedón</i>	349.9 s	200.1 s	446 (44.6 %)
<i>Híbrido</i>	276.41 s	187.4 s	554 (55.4 %)

ejemplo de la fase del algoritmo en el que los UAVs se dirigen hacia la recién encontrada nube se muestra en la Figura 11.

Otro resultado favorable al algoritmo híbrido es la velocidad media en encontrar la nube en los casos en los que es vencedor. Generalmente, cuando el algoritmo boustrofedón vence al algoritmo híbrido, lo hace un 26.02 % antes que este último. Por el contrario, cuando vence el algoritmo híbrido, detecta la zona de baja irradiancia un 31.56 % más rápido.

6. Conclusiones

En este artículo se ha propuesto un algoritmo nuevo para detectar zonas de baja irradiancia causadas por el efecto de la sombra de una nube. El método presentado adapta las ideas del algoritmo ACO al problema de cobertura de área con grupos multi-UAVs, combinando su carácter probabilístico con un algoritmo determinista y de eficiencia probada como el boustrofedón. El algoritmo híbrido se usa para detectar las zonas de baja irradiancia solar causadas por los efectos de una nube cuya forma es posteriormente caracterizada usando el operador Sobel, una técnica de visión artificial para calcular derivadas discretas. Este método se usa para calcular el gradiente de irradiancia solar y, a través de él, la dirección de la isólinea de irradiancia solar provocada por la nube.

Como primera propuesta de resolución del problema este algoritmo ha probado ser generalmente más rápido que el boustrofedón clásico, aunque solo lo vence en un 10 % de los casos: en el futuro se trabajará para mejorar este porcentaje, dado que los resultados son alentadores. Este método propone usar las medidas de la IDN de las zonas de baja irradiancia solar para hacer una estimación de la irradiancia espacial.

Las líneas futuras de trabajo incluyen probar el algoritmo con nubes móviles o con más de una nube a la vez en el área, aplicando técnicas de trayectorias en formación para controlar el comportamiento de los UAVs del grupo, o discriminar entre las nubes interesantes y las que no lo son, debido, por ejemplo, a su pequeño tamaño. También se estudiarán los cambios oportunos para tratar nubes con formas no elípticas o no convexas en general, incluso modelándolas a través de un polítopo. Igualmente se estudiará el uso de un algoritmo ACO real previo al movimiento para elegir la ruta óptima de cobertura de área de los UAVs. **Dentro del proyecto del que forma parte este trabajo se está estudiando el uso de cámaras, por un lado que apunten al cielo desde distintas localizaciones de la planta solar o la periferia, y por otro lado, con cámaras incorporadas en los UAVs que apunten al suelo para distinguir sombras. Se estudiará como la información obtenida se puede usar para una mejor detección de las nubes al combinarla con las mediciones que obtienen los UAVs.**

Es importante resaltar que en este trabajo solo se han usado 3 UAVs para cubrir un área grande de 1000x1000 metros, de ahí el resultado lógico en términos de coste de tiempo para encontrar la nube. En el futuro se espera reducir este coste de tiempo

ya sea añadiendo más UAVs, delimitando el área de búsqueda o usando información meteorológica como la dirección del viento, entre otras. Finalmente, está planteado aprovechar la posibilidad de adaptación de este algoritmo a otras aplicaciones cambiando los sensores utilizados o el tipo de robots, por ejemplo, acuáticos o terrestres. De esta forma, el mismo algoritmo puede ser empleado para detectar formas en otras situaciones diferentes a la planteada, como manchas de aceite en grandes superficies de agua, zonas afectadas o dañadas por el efecto de plagas en extensiones de cultivo, etc.

Agradecimientos

Este proyecto ha recibido fondos del European Research Council (ERC) en el marco del programa 'European Union's Horizon 2020 and innovation programme' (grant agreement No 789051).

Referencias

- Aasen, H., Burkart, A., Bolten, A., Bareth, G., 2015. Generating 3d hyperspectral information with lightweight uav snapshot cameras for vegetation monitoring: From camera calibration to quality assurance. *ISPRS Journal of Photogrammetry and Remote Sensing* 108, 245–259.
- Acar, E. U., Choset, H., 2000. Critical point sensing in unknown environments. In: *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No. 00CH37065)*. Vol. 4. IEEE, pp. 3803–3810.
- Acar, E. U., Choset, H., Rizzi, A. A., Atkar, P. N., Hull, D., 2002. Morse decompositions for coverage tasks. *The international journal of robotics research* 21 (4), 331–344.
- Anderson, B. D., Fidan, B., Yu, C., Walle, D., 2008. Uav formation control: Theory and application. In: *Recent advances in learning and control*. Springer, pp. 15–33.
- Ashley, T., Carrizosa, E., Fernández-Cara, E., 2017. Optimisation of aiming strategies in solar power tower plants. *Energy* 137, 285–291.
- Avellar, G. S., Pereira, G. A., Pimenta, L. C., Iscold, P., 2015. Multi-uav routing for area coverage and remote sensing with minimum time. *Sensors* 15 (11), 27783–27803.
- Bar-Cohen, Y., 2006. Biomimetics?using nature to inspire human innovation. *Bioinspiration & biomimetics* 1 (1), P1.
- Camacho, E. F., Gallego, A., 2013. Optimal operation in solar trough plants: A case study. *Solar Energy* 95, 106–117.
- Camacho, E. F., Soria, M. B., Rubio, F. R., Martínez, D., 2012. *Control of Solar Energy Systems*. Springer Science & Business Media.
- Cesetti, A., Frontoni, E., Mancini, A., Zingaretti, P., Longhi, S., 2010. A vision-based guidance system for uav navigation and safe landing using natural landmarks. *Journal of intelligent and robotic systems* 57 (1–4), 233.
- Choi, Y., Choi, Y., Briceno, S., Mavris, D. N., 2020. Energy-constrained multi-uav coverage path planning for an aerial imagery mission using column generation. *Journal of Intelligent & Robotic Systems* 97 (1), 125–139.
- Choset, H., Pignon, P., 1998. Coverage path planning: The boustrophedon cellular decomposition. In: *Field and service robotics*. Springer, pp. 203–209.
- Coomes, M., Chen, W.-H., Liu, C., 2017. Boustrophedon coverage path planning for uav aerial surveys in wind. In: *2017 International Conference on Unmanned Aircraft Systems (ICUAS)*. IEEE, pp. 1563–1571.
- Current, J. R., Schilling, D. A., 1989. The covering salesman problem. *Transportation science* 23 (3), 208–213.
- Daus, P. H., 1932. The march meeting of the southern california section. *The American Mathematical Monthly* 39 (7), 373–374.
DOI: 10.1080/00029890.1932.11987331

- DJI Technology Inc., 2015. Phantom 3 pro user manual. https://dl.djicdn.com/downloads/phantom_3/en/Phantom_3_Professional_User_Manual_V1.6.pdf, accessed: 2021-01-18.
- Dorigo, M., 1991. Ant colony optimization? new optimization techniques in engineering. by Onwubolu, GC, and BV Babu, Springer-Verlag Berlin Heidelberg, 101–117.
- Galceran, E., Carreras, M., 2013. A survey on coverage path planning for robotics. *Robotics and Autonomous systems* 61 (12), 1258–1276.
- García, R., Orihuela, L., Millán, P., Rubio, F., Ortega, M., 2020. Guaranteed estimation and distributed control of vehicle formations. *International Journal of Control*, 1–24.
DOI: 10.1080/00207179.2020.1714074
- Jin, J., Tang, L., 2010. Optimal coverage path planning for arable farming on 2d surfaces. *Transactions of the ASABE* 53 (1), 283–295.
- Johnson, D. S., McGeoch, L. A., 1997. The traveling salesman problem: A case study in local optimization. *Local search in combinatorial optimization* 1 (1), 215–310.
- Kuhn, P., Wilbert, S., Prah, C., Schüler, D., Haase, T., Hirsch, T., Wittmann, M., Ramirez, L., Zarzalejo, L., Meyer, A., et al., 2017. Shadow camera system for the generation of solar irradiance maps. *Solar Energy* 157, 157–170.
- Law, E. W., Prasad, A. A., Kay, M., Taylor, R. A., 2014. Direct normal irradiance forecasting and its application to concentrated solar thermal output forecasting—a review. *Solar Energy* 108, 287–307.
- Nouri, B., Kuhn, P., Wilbert, S., Prah, C., Pitz-Paal, R., Blanc, P., Schmidt, T., Yasser, Z., Santigosa, L. R., Heineman, D., 2018. Nowcasting of dni maps for the solar field based on voxel carving and individual 3d cloud objects from all sky images. In: *AIP Conference Proceedings*. Vol. 2033. AIP Publishing LLC, p. 190011.
- Ntawumenyikizaba, A., Viet, H. H., Chung, T., 2012. An online complete coverage algorithm for cleaning robots based on boustrophedon motions and a* search. In: *2012 8th International Conference on Information Science and Digital Content Technology (ICIDT2012)*. Vol. 2. IEEE, pp. 401–405.
- Oksanen, T., Visala, A., 2009. Coverage path planning algorithms for agricultural field machines. *Journal of field robotics* 26 (8), 651–668.
- Rokhmana, C. A., 2015. The potential of uav-based remote sensing for supporting precision agriculture in indonesia. *Procedia Environmental Sciences* 24 (2015), 245–253.
- Sánchez, A., Gallego, A., Escaño, J., Camacho, E., 2018. Temperature homogenization of a solar trough field for performance improvement. *Solar Energy* 165, 1–9.
- Sánchez, A., Gallego, A., Escaño, J., Camacho, E. F., 2019. Thermal balance of large scale parabolic trough plants: A case study. *Solar Energy* 190, 69–81.
- Savant, S., 2014. A review on edge detection techniques for image segmentation. *International Journal of Computer Science and Information Technologies* 5 (4), 5898–5900.
- Sheng, H., Chao, H., Coopmans, C., Han, J., McKee, M., Chen, Y., 2010. Low-cost uav-based thermal infrared remote sensing: Platform, calibration and applications. In: *Proceedings of 2010 IEEE/ASME International Conference on Mechatronic and Embedded Systems and Applications*. IEEE, pp. 38–43.
- Silvagni, M., Tonoli, A., Zenerino, E., Chiaberge, M., 2017. Multipurpose uav for search and rescue operations in mountain avalanche events. *Geomatics, Natural Hazards and Risk* 8 (1), 18–33.
- Sobel, I., Feldman, G., 1968. A 3x3 isotropic gradient operator for image processing. a talk at the Stanford Artificial Project in, 271–272.
- Technologies, S. M., 2019. Sun Sensor NANO-ISSX/c technical specifications. http://www.solar-mems.com/smt_pdf/NANO_Technical_Specifications.pdf, accessed: 2020-08-18.
- Xiong, C., Chen, D., Lu, D., Zeng, Z., Lian, L., 2019. Path planning of multiple autonomous marine vehicles for adaptive sampling using voronoi-based ant colony optimization. *Robotics and Autonomous Systems* 115, 90–103.
- Xu, L., Wang, Z., Yuan, G., Sun, F., Zhang, X., 2015. Thermal performance of parabolic trough solar collectors under the condition of dramatically varying dni. *Energy Procedia* 69, 218–225.
- Yfantis, E., 2019. A uav with autonomy, pattern recognition for forest fire prevention, and ai for providing advice to firefighters fighting forest fires. In: *2019 IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC)*. IEEE, pp. 0409–0413.
- Zhang, Z., Schwartz, S., Wagner, L., Miller, W., 2000. A greedy algorithm for aligning dna sequences. *Journal of Computational biology* 7 (1-2), 203–214.