

Evaluacion de rutas de cobertura.

Evaluacion de rutas de cobertura utilizando un planificador de cobertura.

by AG ariel.guerrero@uc.edu.py 27-07-2024

Referencia:

[1] Torres, Marina, David A. Pelta, José L. Verdegay, and Juan C. Torres. "Coverage Path Planning with Unmanned Aerial Vehicles for 3D Terrain Reconstruction." Expert Systems with Applications 55 (August 2016): 441–51. <https://doi.org/10.1016/j.eswa.2016.02.007>.

[2] Li, Yan, Hai Chen, Meng Joo Er, and Xinmin Wang. "Coverage Path Planning for UAVs Based on Enhanced Exact Cellular Decomposition Method." Mechatronics 21, no. 5 (August 2011): 876–85. <https://doi.org/10.1016/j.mechatronics.2010.10.009>.

[3]"Optimally Survey and Customize Coverage of Region of Interest" disponible en <https://la.mathworks.com/help/uav/ug/optimally-survey-and-customize-coverage-of-region-of-interest-using-coverage-planner.html>

[4]"Execute Coverage Plan Using UAV Mission" disponible en <https://la.mathworks.com/help/uav/ref/uavcoverageplanner.html>

[5] Zhang, C. (2020). Area Coverage with Unmanned Aerial Vehicles Using Reinforcement Learning. https://github.com/zcczhang/UAV_Coverage

[6] Nam, L. H., Huang, L., Li, X. J., & Xu, J. F. (2016, April). An approach for coverage path planning for UAVs. In 2016 IEEE 14th international workshop on advanced motion control (AMC) pp. 411-416). IEEE.

Table of Contents

Referencia:	1
1. Antecedentes	1
2. Defina la region de interes	2
2.1 Crear ejes geografico con un mapa base	2
2.2 Define los vértices del polígono de manera interactiva seleccionando puntos en el mapa	3
2.3 Para definir interactivamente una región de evaluacion	4
3. Dividir la region de interes en subregiones	4
3. Crear el espacio de cobertura y planificar la ruta para la ROI	5
Crear la region de cobertura y planificar la ruta	6
4. Personalizando la region de cobertura	7
4.1 Personalizacion del patron de barrido	7
4.2 Optimizacion del punto de inicio	8
4.3 Replanificacion de la cobertura	8
4.4 Verificar el costo de transicion	8
5. Proximos pasos	8
5.1 Exportar el patrón de navegación compatible con QGC/ArduPilot	8
5.2 Plan de vuelo generado	8
5.3 Simular la ruta de cobertua usando UAV Mission	9

1. Antecedentes

La planificación de cobertura (CPP - Coverture Path Planning) es el proceso de encontrar un plan de ruta óptimo para que un UAV inspeccione toda la región de una determinada área geográfica. Para inspeccionar una región, un UAV típicamente vuela en una dirección y utiliza giros en ángulo recto para alternar la dirección. una buena opción es descomponer el área en subáreas convexas y luego, puedes elegir de manera óptima una estrategia de cobertura que sea mejor para cada subregión y, posteriormente, conectar las rutas para completar la misión.

```
clear variables
close all
clc
```

2. Defina la region de interes

Para comenzar la planificación de cobertura, primero debes definir la región de interés como un polígono o un conjunto de polígonos. Primero, crea el eje geográfico, establece los límites de latitud y longitud, y calcula la media de los límites para obtener el geocentro del área visualizada. Establecer la altura de referencia en 3 metros.

2.1 Crear ejes geografico con un mapa base

`gax = geoaxes(Basemap="satellite");` % Informacion obtenida de google earth Probar

```
fig = figure('Name', 'Mission-Creator');
gax = geoaxes(fig);
% Si esto funciona, intenta agregar el basemap
set(gax, 'Basemap', 'satellite');

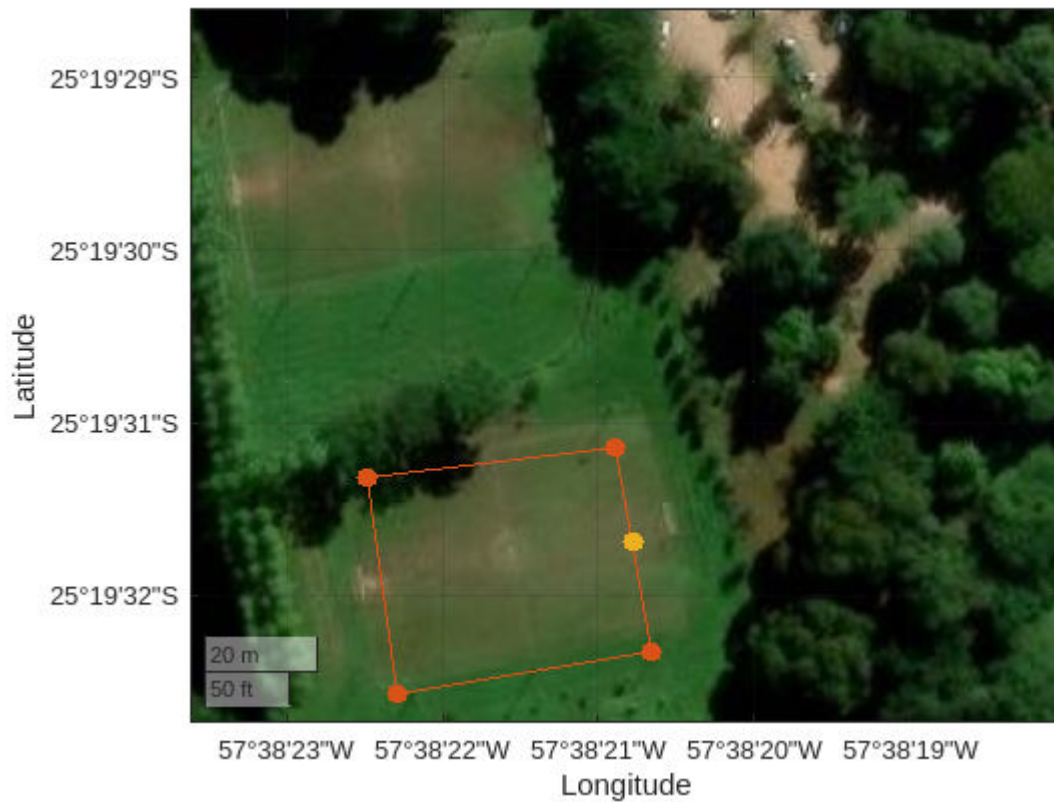
latlimits = [-25.3257611 -25.3246139]; % ROI-01 y ROI-02
lonlimits = [-57.6396222 -57.6386111];
geolimits(latlimits,lonlimits);
geocenter = [mean(latlimits) mean(lonlimits) 0];
refHeight = 3; % Altitud de vuelo del Dron
hold on
```



2.2 Define los vértices del polígono de manera interactiva seleccionando puntos en el mapa

o puedes usar vértices predefinidos almacenados en predefinedROI.mat para definir el polígono, la ubicación de despegue y la ubicación de aterrizaje. Por defecto, se utiliza los puntos predefinidos. Para definir el polígono de manera interactiva, establece interactiveROI en true.

```
interactiveROI = false;
%[landLat, landLon, takeoffLat, takeoffLon, llapoints, xyzpoints] =
ConfigurarRegionInteresFcn(geocenter, 'predefinedROI.mat');
load predefinedROI_V01.mat
GraficarDespegueROIAtterrizaje(gax, takeoffLat, takeoffLon, landLat, landLon, llapoints)
```



2.3 Para definir interactivamente una región de evaluación

Luego, define una ubicación de despegue, un polígono y una ubicación de aterrizaje.

```
if(interactiveROI)
    % Almacenamiento de los límites visuales actuales
    % (latitud, longitud)
    [visLimitsLat,visLimitsLon] = geolimits;

    % Selección interactiva de la ubicación de despegue (Takeoff)
    [takeoffLat,takeoffLon] =
    SeleccionarPuntoDespegueFcn(gax,visLimitsLat,visLimitsLon);

    % Definición interactiva del polígono de la región de interés
    [llapoints,xyzpoints] =
    SeleccionarPoligonoFcn(gax,visLimitsLat,visLimitsLon,geocenter);

    % Selección interactiva de la ubicación de aterrizaje (Land)
    [landLat,landLon] =
    helperLandSelectionFcn(gax,visLimitsLat,visLimitsLon);
end
```

3. Dividir la region de interes en subregiones

Para inspeccionar óptimamente un área compleja, primero hay que descomponer la región en áreas convexas más simples.

```
% Descompone la ROI en areas mas pequeñas y simples en coordenadas ENU
subAreasEN = coverageDecomposition(xyzpoints(:,1:2));
% Creacion de un array de celdas para las nuevas areas.
subAreasLLA = cell(1,numel(subAreasEN));

% Conversion de las coordenadas locales ENU a coordenadas geodesicas.
for i = 1:numel(subAreasEN)
    % Altura de referencia a las coordenadas ENU locales
    altitude = refHeight * ones(size(subAreasEN{i}, 1), 1);
    localENU = [subAreasEN{i}(:,1), subAreasEN{i}(:,2), altitude];

    % Convierte las coordenadas locales ENU a coordenadas geodesicas LLA
    % subArea = enu2lla(localENU, geocenter, "flat");
    subArea = ned2lla(localENU, geocenter, "flat"); % PX4
    % Almacena las coordenadas geodesicas LLA en el array de celdas
    subAreasLLA{i} = subArea(:,1:2);
end
```

3. Crear el espacio de cobertura y planificar la ruta para la ROI

El objetivo es inspeccionar optimamente cada subregion minimizando el numero de giros y la distancia de conexion entre subregiones. La distancia de conexion es la distancia que el UAV debe recorrer para llevar al siguiente poligono a inspeccionar.

```
% Crear el espacio de cobertura
cs = uavCoverageSpace(Polygons=subAreasLLA, ...
    UnitWidth=4, ...
    Sidelap=0, ...
    ReferenceHeight=refHeight, ...
    UseLocalCoordinates=false, ...
    ReferenceLocation=geocenter);

% Mostrar el espacio de cobertura en los ejes geográficos
ax = cs.show(Parent=gax, LineWidth=1.25);
```



Crear la region de cobertura y planificar la ruta

```
% Crear el planificador de cobertura
cp = uavCoveragePlanner(cs);
% Planificar la ruta de cobertura
[wpts,solnInfo] = plan(cp,[takeoffLat takeoffLon 0],[landLat landLon 0]);

% Mostrar la ruta de cobertura en el espacio de ruta de cobertura
% en el espacio de cobertura
path1 = geoplots(gax, wpts(:,1), wpts(:,2), 'LineWidth', 1.5);
hold off;
```




```
% Verificar el costo de transicion en metros
% Este valor representa la distancia total que el UAV debe recorrer entre
los poligonos,
% incluyendo la distancia desde la ubicaci3n de despegue al primer poligono
% y desde el 3ltimo poligono hasta la ubicaci3n de aterrizaje
```

```
connectionDistancePath1 = solnInfo.TransitionCost
```

4. Personalizando la region de cobertura

4.1 Personalizacion del patron de barrido

Aunque el planificador determina automaticamente el patron de barrido en cada subregion, se puede ajustar manualmente para mejorar la cobertura dependiendo del tipo de aplicacion.

```
% cs.setCoveragePattern(1,SweepAngle=90); %

% En el ejemplo de referencia [1], el angulo de barrido de los poligonos 2
% y 3 se establece en 90 grados para alinearse mejor con la orientacion
% deseada
% cs.setCoveragePattern(2,SweepAngle=90);
% cs.setCoveragePattern(3,SweepAngle=90);
```

4.2 Optimizacion del punto de inicio

En el ejemplo de referencia [1], el poligono 3 parece ser mejor punto de inicio para la mision de cobertura, ya que esto minimizara la distancia que el Dron necesita viajar fuera de la mision de cobertura, optimizando el uso de bateria y tiempo.

```
%cp.SolverParameters.StartingArea = 3; % Solo 1 region
```

4.3 Replanificacion de la cobertura

Despues de ajustar el angulo de barrido y el area de inicio, se vuelve aplanificar la ruta de cobertura con estos nuevos parametros. Finalmente, se elimina la ruta anterior y se traza la nueva ruta de cobertura en el grafico para visualizar el nuevo plan de vuelo.

```
% Replanificar la ruta de cobertura
% [wpts,custSoln] = plan(cp,[takeoffLat takeoffLon 0],[landLat landLon 0]);
% landLat = takeoffLat; % Retorna al punto de despegue
% landLon = takeoffLon;
% [wpts,custSoln] = plan(cp,[takeoffLat takeoffLon 0],[landLat landLon 0]);

% Remover la ruta anterior y actualizar a la nueva ruta
% delete(path1);
% hold on
% h2 = geoplot(gax,wpts(:,1),wpts(:,2),LineWidth=1.5);
% hold off
```

4.4 Verificar el costo de transicion

```
% connectionDistancePath2 = custSoln.TransitionCost
```

5. Proximos pasos

5.1 Exportar el patrón de navegación compatible con QGC/ArduPilot

Archivo de waypoints en el formato QGroundControl Waypoint List (QGC WPL) para planificar una misión de un UAV (dron).

```
exportWaypointsPlan(cp,solnInfo,"misionLED.waypoints");
% exportWaypointsPlan(cp,custSoln,"customCoverage.waypoints"); %ver [1]
```

5.2 Plan de vuelo generado

```
readAndPrintWaypoints('misionLED.waypoints');
```

Index	Command	Latitud	Longitud	Altitud
----	-----	-----	-----	-----
0	16	-25.325470	-57.639105	0.00
1	22	-25.325470	-57.639105	3.00
2	16	-25.325629	-57.639081	3.00
3	16	-25.325695	-57.639523	3.00

4	16	-25.325659	-57.639529	3.00
5	16	-25.325593	-57.639088	3.00
6	16	-25.325558	-57.639095	3.00
7	16	-25.325623	-57.639534	3.00
8	16	-25.325587	-57.639539	3.00
9	16	-25.325522	-57.639102	3.00
10	16	-25.325487	-57.639109	3.00
11	16	-25.325552	-57.639545	3.00
12	16	-25.325516	-57.639550	3.00
13	16	-25.325451	-57.639116	3.00
14	16	-25.325416	-57.639123	3.00
15	16	-25.325480	-57.639555	3.00
16	16	-25.325444	-57.639561	3.00
17	16	-25.325380	-57.639130	3.00
18	16	-25.325344	-57.639137	3.00
19	16	-25.325408	-57.639566	3.00
20	16	-25.325373	-57.639572	3.00
21	16	-25.325348	-57.639403	3.00
21	21	-25.325470	-57.639105	0.00

5.3 Simular la ruta de cobertura usando UAV Mission

Crear una mision y simular la mision

```
% mission =
uavMission(PlanFile="customCoverage.waypoints",Speed=5,InitialYaw=90);
% SimularUAVMission(mission,geocenter)
```