

GUÍA RÁPIDA: RECTIFICACIÓN DE IMÁGENES Y ALGUNAS APLICACIONES

1- Extraer Frames de un Video



a1_ExtrearFrames.m

- Abrir script **a1_ExtrearFrames.m**
- Especifique cuántos frames por segundo desea extraer del video, lo normal son 2 (fpsR = 2) (línea 8)
- Ejecute el script
- Seleccione archivo de video (línea 5)
- Seleccione carpeta donde desee guardar los frames. Idealmente debe ser → demoMovies → demoClip1 (línea 7)
- Espere mientras se extraen y guardan las imágenes.

```
1 % EXTRAER FRAMES DE UN VIDEO
2 - clear
3 - clc
4 - addpath(genpath('.'))
5 - [NombreV DirV]=uigetfile('*.mov', 'Selección de video');
6 - Direccion=[DirV filesep NombreV];
7 - DirGuard = uigetdir([], 'Selección de carpeta para guardar');
8 - fpsR=2; % Razón de extracción de frames, ¿cuántos frames por segundo?
9 - video2images(Direccion, fpsR, DirGuard)
```

2- Determinar Sistema de Referencia Local



a2_SistemaRef.m

- Abrir script **a2_SistemaRef.m**
- Ingrese nombre y coordenadas de los puntos con coordenadas conocidas en UTM (línea 5 – línea 24)
- Especifique origen del sistema de referencia local en UTM (punto de referencia). (línea 30)
- Indique el ángulo de rotación del sistema local con respecto al sistema de coordenadas UTM. (línea 33)

```
1 % SISTEMA DE REFERENCIA LOCAL
2 - clear
3 - clc
4 % Ingresar nombre y coordenadas de los puntos de referencia
5 - GCP_UTM.Name(1,:)= "P1";
6 - GCP_UTM.Name(2,:)= "P2";
7 - GCP_UTM.Name(3,:)= "P3";
15 - GCP_UTM.CoordUTM(1,:)= [256841.603 6289708.571]';
16 - GCP_UTM.CoordUTM(2,:)= [256864.953 6289668.731]';
17 - GCP_UTM.CoordUTM(3,:)= [256906.087 6289667.163]';
18 - GCP_UTM.CoordUTM(4,:)= [256866.15 6289687.263]';
29 %Ingresar punto de referencia [E N]
30 - Re=[256884.448 6289741.483]';
31
32 %Ingresar el angulo de giro en grados (counterclockwise)
33 - theta =150;
```



x-y : Sistema de Coordenadas en UTM
x'-y' : Sistema de Coordenadas en UTM trasladado
x''-y'': Sistema de Coordenadas rotado (Sistema Local)

- Ejecutar el script
- Las coordenadas de los puntos transformadas al sistema de referencia local son guardadas en el archivo refPOINT.mat, el cual se utilizará en etapas siguientes.

3- Configuración INPUTS para análisis UAV



- Abrir script **a3_ConfiguracionUAV.m**
- Se abrirá el script **demoInputFile.m** donde podrá configurar los valores de entrada del análisis.
- Los principales valores que se deben cambiar son los siguientes:

1. Paths, names and time stamp info:

- stationStr = 'Aerielle' (Relacionado con los parámetros intrínsecos y calibración de la cámara, revisar función makeLCPP3.m)
- dateVect → Fecha del primer frame
[aaaa mm dd hh mm ss]

2. Geometry solution Inputs:

- Parámetros extrínsecos de la cámara → [xCam yCam zCam Azimuth Tilt Roll]
- knownFlags: 1 → Si se conoce la variable
0 → No se conoce la variable
- xyCam, zCam, azTilt, roll → si su valor es conocido se debe anotar, si no es conocido, este valor sirve como primer punto de iteración.

3. GCP Info

- gcpList → Anotar el número de los puntos GCP que utilizará en el análisis, son los que se ven en el video
- nRefs → Número de puntos virtuales que utilizará, son los que se distinguen fácilmente en la imagen. Pueden o no ser GCP.

4. Processing Parameters

- rectxy → Grilla de rectificación en X e Y, [xmin dx xmax ymin dy ymax]
- rectz → Nivel vertical para la rectificación (Generalmente es el nivel medio del mar).

```
1 % Demo input file for UAV processing.
2 % The user is responsible for correcting content
3
4 % 1. Paths, names and time stamp info:
5 - inputs.stationStr = 'Aerielle';
6 - inputs.dateVect = [2018 11 13 10 39 00]; %
7 - inputs.dt = 0.5/(24*3600); % delta t (
8 - inputs.frameFn = 'demoClip'; % root o
9 - inputs.gcpFn = [pwd, filesep, 'refPOINT.mat'];
10 - inputs.instsFn = [pwd, filesep, 'demoInstsFile'];
```

13 % 2. Geometry solution Inputs:

```
22 - inputs.knownFlags = [0 0 0 0 0 0];
23 - inputs.xyCam = [-107.4604 265.6772];
24 - inputs.zCam = 80; % based on last data run
25 - inputs.azTilt = [95 60] / 180*pi; % first guess
26 - inputs.roll = 0 / 180*pi;
```

28 % 3. GCP info

```
29 % the length of gcpList and value of nRefs must b
30 - inputs.gcpList = [1 2 3 4 5 6 7 8 9 10 11];
31 - inputs.nRefs = 3; % number of
32 - inputs.zRefs = 2; % assumed z
```

% 4. Processing parameters

```
inputs.doImageProducts = 1;
inputs.showFoundRefPoints = 0;
inputs.showInputImages = 1;
inputs.rectxy = [-50 0.5 400 -300 0.5 300];
inputs.rectz = 0;
```

- Guardar y cerrar **demoInputFile.m**
- También se abrirá automáticamente el archivo **demoInputFile.m**, donde podrá

5. vBar Instruments

Se utiliza una línea discretizada, paralela a la línea de costa, para la obtención de datos rectificados. Para definirlo se utiliza la siguiente estructura:

X = cte
Y = [Ymin Ymax]

6. Runup Lines

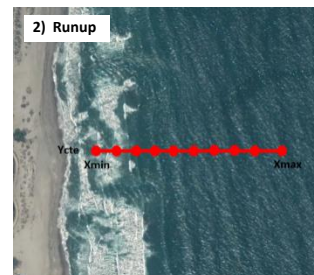
Se utiliza una línea discretizada, transversal a la línea de costa, para la obtención de datos rectificados. Para definirlo se utiliza la siguiente estructura:

X = [Xmin Xmax]
Y = cte

7. cBathy Array

Se utiliza una una region rectangular discretizada (grilla) para la obtención de datos rectificados. Para definirlo se utiliza la siguiente estructura:

X = [xmin xmax]
Y = [ymin ymax]



4- Rectificación de Imágenes



- Abrir script **a4_RectificacionImagenes.m** y ejecutarlo
Obs: Es recomendable ir ejecutando la líneas de código por sección, e ir comprobando que se van obteniendo los resultados esperados.
- Elija la carpeta donde desea guardar los outputs del algoritmo.
- Elija la carpeta que contiene los frames extraídos del video (debe ser demoMovies)
Obs: Antes de ejecutar debe verificar que el archivo **"refPOINT.mat"** se encuentra en la carpeta actual.
- Seleccione los GCP en la imagen, es recomendable tener una foto de referencia que indique claramente donde está ubicado cada punto, tal como se muestra en la siguiente imagen.

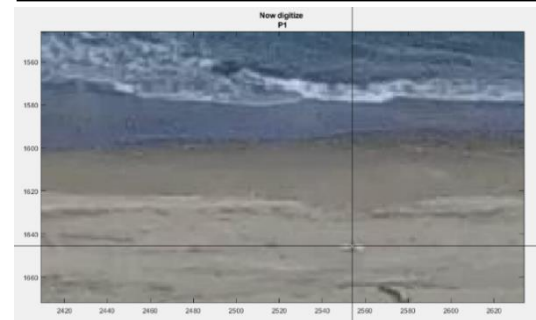


- Elija los gcp virtuales, estos tienen que ser puntos idealmente blancos o muy claros, con la idea de que se diferencien bastante de su entorno, suelen servir los techos de las casas.
Obs: No es necesario que se conozcan las coordenadas reales de estos puntos. Tampoco hay problema si existen puntos que cumplen tanto como GCP como GCP virtuales.
- Una vez elegido el recuadro que encierra al GCP virtual, se debe fijar un umbral de intensidad. Debe asegurarse que este umbral sea tal, que la forma (límite) que define el GCP virtual no se altere durante el análisis de todos los frames.
Obs El valor de este umbral está entre los 180 y 220 normalmente.
- Siga ejecutando las secciones
- Espere mientras se rectifican las imágenes.
- Al finalizar se guardarán los resultados en la carpeta que usted definió.

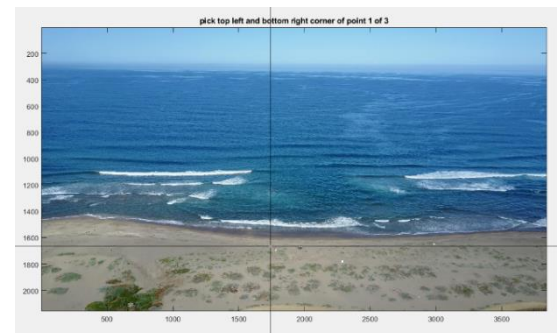
1- Haga Click para hacer zoom



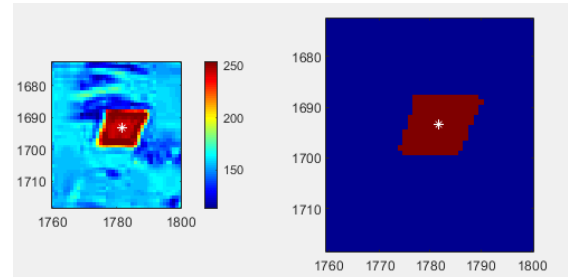
2- Cuando tenga un identificado claramente el GCP, presione "enter" y digitalice.



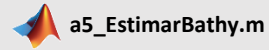
3- Seleccione la esquina superior izquierda e inferior derecha, formando un recuadro que encierre al GCP virtual



3- Ingrese umbral de intensidad para filtrar objeto



5- Estimación de la Batimetría con cBathy



- Abrir script **a5_EstimarBathy.m** y vaya ejecutando la secciones:
- Elija el archivo cBathy a partir del cual desee estimar la batimetría, este archivo corresponde a uno de los OUTPUTS del proceso anterior (rectificación de imágenes).
- Posteriormente le corresponde abrir el archivo **argus02a**, donde puede editar los parámetros de entrada del análisis:

- Los principales elementos a cambiar son las primeras cinco entradas:

stationStr: Nombre del presente archivo matlab

dxm / dxy : Deltas deseados de la matriz de muestreo

xyMinMax: [xMin xMax yMin y Max]
Especifica la extensión espacial de la grilla de análisis.

tideFunction: Corresponde a la función proporcionada por el usuario, utilizada para encontrar la corrección de elevación de la marea. Esto puede requerir manipulación del usuario para sistemas que no son Argus. (Esteban: yo no cambié esto).

Obs: Los demás parámetros corresponden a inputs que un usuario avanzado puede utilizar. Sólo si se comprende lo que se está está haciendo se puede cambiar.

- Ejecutar algoritmo cBathyDemo
- La estimación de la profundidad estará contenida en la estructura:

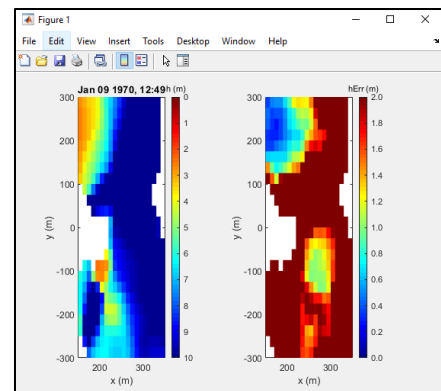
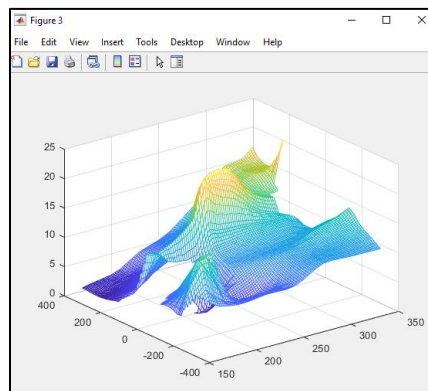
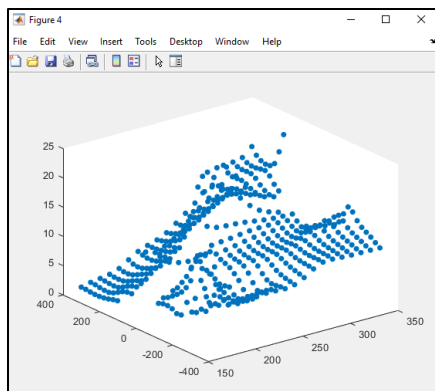
bathy.fCombined.h

- Finalmente puede graficar los resultados asociados.

```

1  %%% Site-specific Inputs
2  - params.stationStr = 'argus02a';
3  - params.dxm = 10;
4  - params.dym = 25;
5  - params.xyMinMax = [150 350 -300 300]
6
7  - params.tideFunction = 'cBathyTide';
8
9  %%%%%%%%% Power user settings from h
10 - params.MINDEPTH = 0.25;
11 - params.minValsForBathyEst = 4;
12
13 - params.QTOL = 0.5;
14 - params.minLam = 10;
15 - params.Lx = 3*params.dxm;
16 - params.Ly = 3*params.dym;
17 - params.kappa0 = 2;
18 - params.DECIMATE = 1;
19 - params.maxNFix = 80;
20
21 % f-domain etc.
22 - params.fB = [1/18: 1/50: 1/4];
23 - params.nKeep = 4;
24
25 % debugging options
26 - params.debug.production = 1;
27 - params.debug.DOPlotStackAndPhaseMaps
28 - params.debug.DOSHOWPROGRESS = 1;
29 - params.debug.DOPlotPhaseTile = 1;
30 - params.debug.TRANSECTX = 200;
31 - params.debug.TRANSECTY = 900;

```



Rutinas para convertir frames rectificados a video

6- Rectificación de imágenes (Guardado en JPG)



a6_RectImágenesJPG.m

- Abrir script **a6_RectImágenesJPG.m** y ejecutarlo.
- La rutina es muy parecida a **a4_RectificacionImágenes.m**, se tiene que seguir los mismos pasos iniciales.
- Una vez terminado el proceso con éxito, las imágenes rectificadas quedarán guardadas en la carpeta especificada.

7- Convertir imágenes a video



a7_FramesToVideo.m

- Abrir script **a7_FramesToVideo.m**.
- Defina la cantidad de frames por segundo que desea en el video, esto determinará la duración de este. ([Línea 8](#))
- Si desea agregar más opciones de configuración del video, revise la documentación de **VideoWriter** que pone a disposición matlab
- Ejecute el script.
- El video se guardará en la carpeta actual.

```
1 % CONVERTIR FRAMES A VIDEO
2
3 - clear
4 - DirGuard = uigetdir([], 'Seleccione carpeta donde se usará el video');
5 - clipFns = dir(DirGuard);
6 - Nf=length(clipFns)-2 % Examinar la estructura clipFns para ver los nombres de los archivos
7 - movie_obj = VideoWriter('TestMovie.avi');
8 - movie_obj.FrameRate=15; % Especifique con cuantos frames por segundo desea
9 - open(movie_obj)
10 - h = waitbar(0, 'Procesando...');
11 - for K = 1 : Nf
12 -     this_image = imread([DirGuard '\FrameRec2_' int2str(K) '.jpg']);
13 -     writeVideo(movie_obj, this_image);
14 -     waitbar(K/Nf, h);
15 - end
16 - close(movie_obj);
17 - waitbar(1, h, 'El proceso ha finalizado correctamente')
```