**Bitácora   
Proyecto Garuda**   
Daniel Ureña López Carnet: 2020425815

* **Jueves 5 septiembre:**  
  Se evaluaron múltiples sistemas de reconocimiento y modelado 3D que utilizan imágenes tomadas a distintos ángulos del objeto o terreno en cuestión, y crea como resultado un objeto 3d en múltiples posibles formatos (como STL, OBJ o GLFT).   
  Dentro de las posibles herramientas evaluadas se encontraron Polycam, entre otros.
* En el caso de Polycam, se hicieron pruebas con una app móvil que poseen, con el fin de probar la eficacia de esta herramienta. Se encontró que se ocupa un mínimo de 20 fotos para que trabaje el programa de forma correcta, además de que se necesita de forma obligatoria una conexión a internet y no posee una API para que se utilice de forma integrada con el software desarrollado en el presente trabajo, únicamente se puede acceder a esta herramienta por medio de página web o app móvil.

A screenshot of a phone

Description automatically generated

El objeto anterior se realizó con múltiples fotografías captadas (mas de 15), se supone que es de una batería posicionada de forma vertical, pero claramente la calidad de captura no es la ideal para este proyecto, debido a la poca claridad de los objetos captados.

* **Jueves 12 septiembre:**

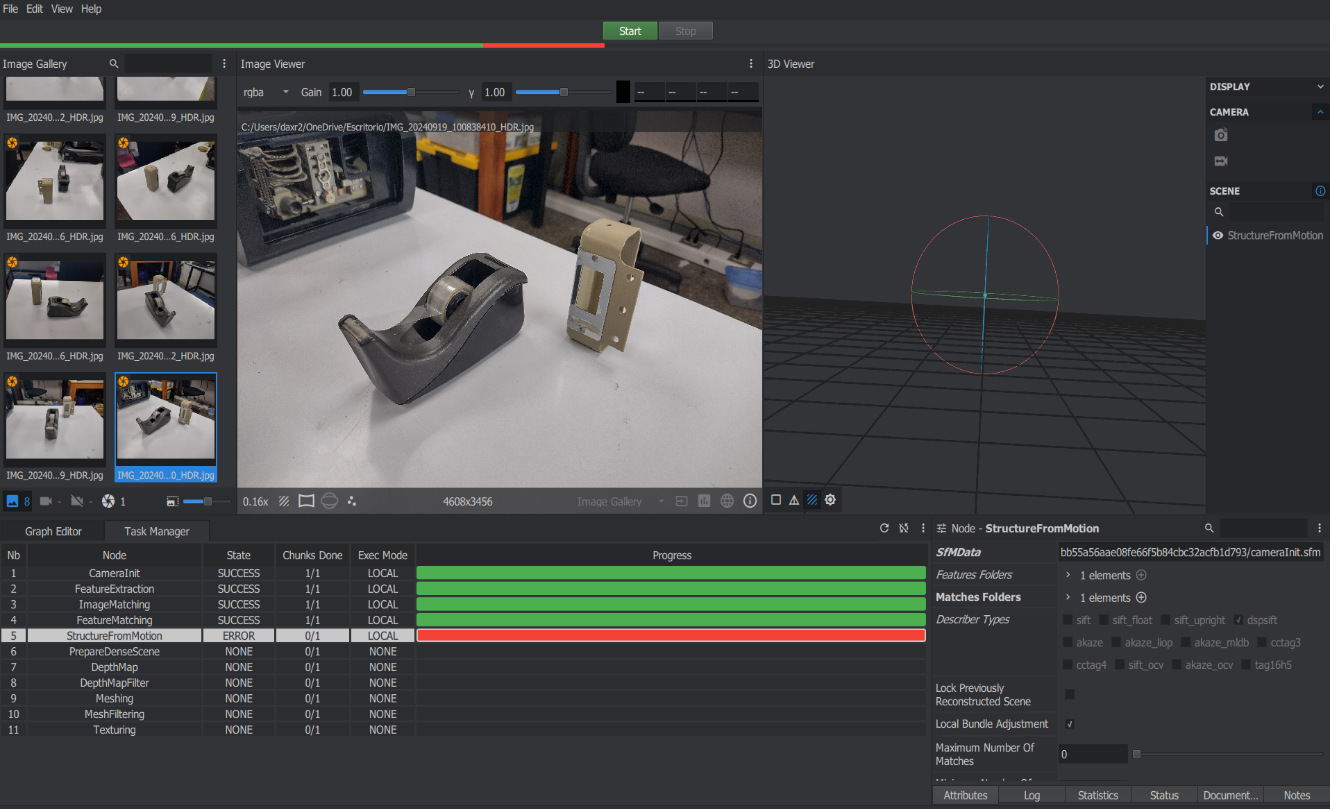
Para el reconocimiento del terreno se evalúan 2 posibles rutas, la inicial que contempla el uso de un software de modelado 3D a partir de fotografías, ahora con la herramienta de Agisoft Metashape, y el de utilizar una herramienta como OpenCV, y software de mapeo GIS, esta ultima parece ser muy capaz de manejar tanto el rastreo de los vehículos como el mapeo del terreno a partir de pocas cámaras (y pocos ángulos).

* Durante el día de hoy se probó una implementación rápida Metashape, probándolo a partir de pocas imágenes, (3 en total) de un hipotético terreno pequeño.  
  Después de implementarlo en Python, se encontró que el módulo para Python de Metashape funciona únicamente si se cuenta con la licencia del software profesional, el cual es de pago, por lo cual se decide descartar el uso de Metashape.

A screenshot of a computer program

Description automatically generated

Por otro lado, también se intentó utilizar el programa Meshroom, el cual también puede ser integrado a programas de lenguajes como Python, y es de código abierto, sin embargo, este necesita un mínimo de 8 imágenes para entregar un modelo decente que se pueda utilizar, sin embargo, el software no tiene un límite de imágenes para procesar. Al intentar procesar una cantidad de 3 imágenes, el sistema no es capaz de generar un modelo 3d. También se probó con 4, y con 8, y el sistema tampoco fue capaz de generar un modelo 3D. Ya que incluso 8 cámaras exceden el numero ideal de cámaras que se utilizarían en el proyecto, y el hecho de que múltiples herramientas tienen limitaciones similares con pocas imágenes, se descarta el uso de algún software modelador de objetos 3D para mapear el área de trabajo. Se enfocarán esfuerzos ahora en el uso de OpenCV.

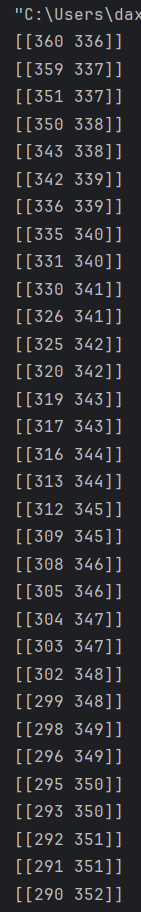


* **Jueves 19 septiembre:**

Se continua con el desarrollo del sistema del reconocimiento del terreno con OpenCV, iniciando con el reconocimiento de la delimitación del terreno, el cual será por medio de líneas amarillas al menos de manera inicial. Se inicia con el reconocimiento de las líneas en las imágenes que provendrían de las imágenes captadas por las cámaras.  
A screenshot of a computer

Description automatically generated

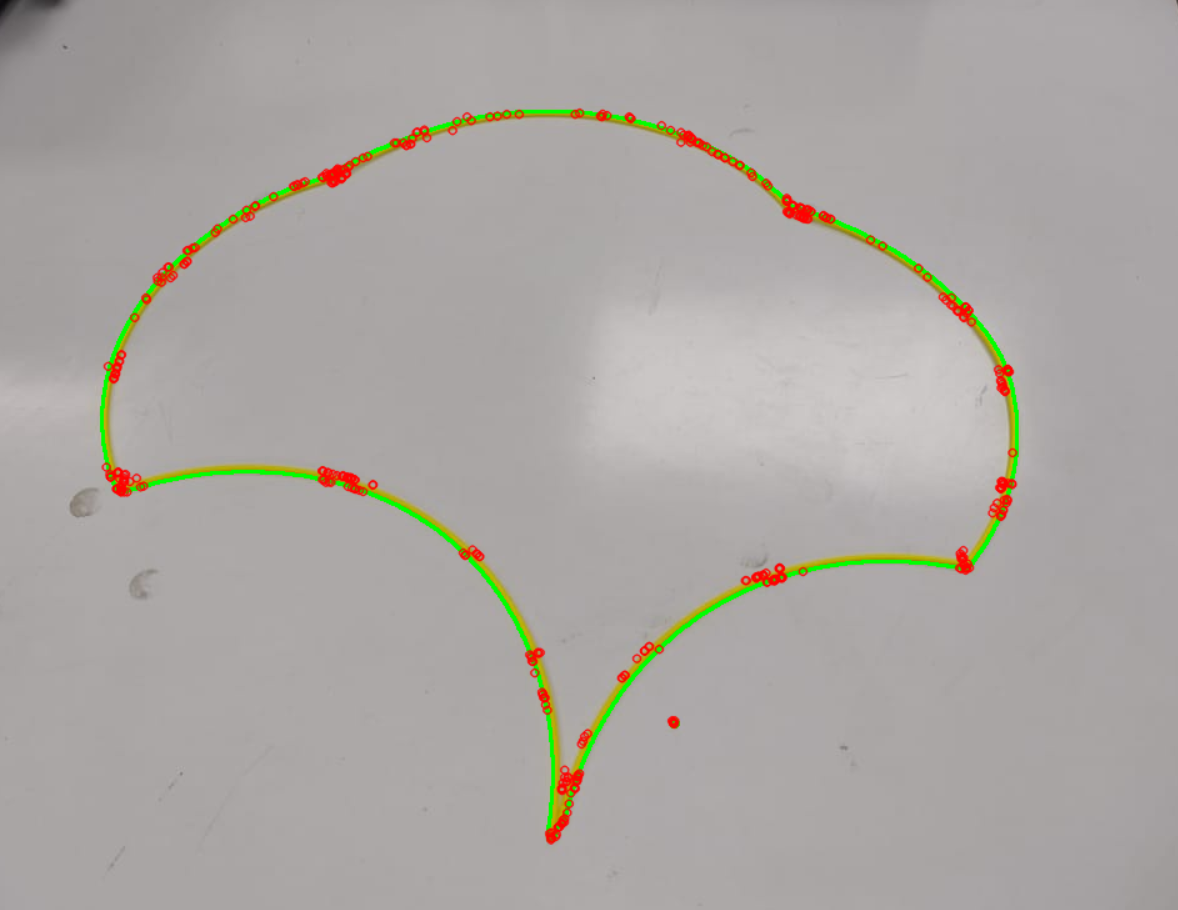
Además, se implementa la función de detectar los puntos y crear coordenadas en la imagen misma.   

Con esta implementación falta la triangulación del terreno con base a múltiples imágenes, de modo de que se pueda dibujar el terreno en una imagen 2D con base a múltiples imágenes.

* **Jueves 26 septiembre:**

Se continuo con el desarrollo del programa que detectaría y recrearía desde una vista superior el delineamiento del terreno, de forma que, en base a lo detectado en las 3 imágenes se recrearía el borde o delineado del terreno de forma precisa. Se desarrollo un programa que detectaría puntos que tienen en común los 3 terrenos detectados desde 3 ángulos distintos, y en base a estos puntos, se reconstruiría el contorno de forma correcta, sin embargo, los resultados de este proceso no fueron los esperados.

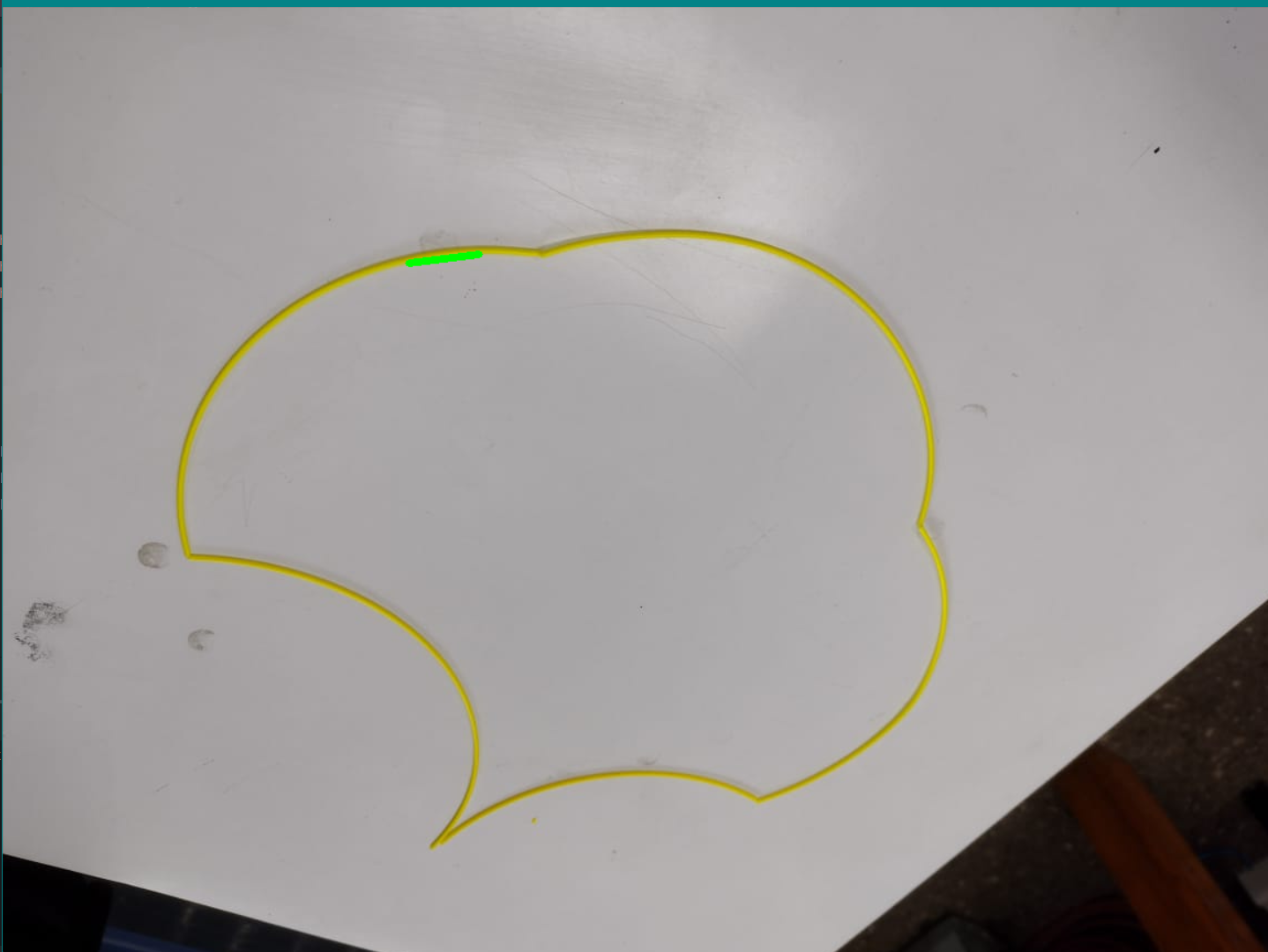
A drawing of a logo

Description automatically generated with medium confidenceA drawing of a shape of a crown

Description automatically generated with medium confidenceA graph with lines drawn on it

Description automatically generated

SLAM



* **Martes 1ro y viernes 4 de octubre:**

Se reevalúa el acercamiento que se ha tomado hacia el proyecto, sobre todo en el reconocimiento del terreno, si bien se ha tenido éxito en la detección del contorno del terreno en cada imagen, no se ha tenido éxito en la obtención de una “vista de pájaro” precisa del terreno a partir de lo observado. Después de investigar un poco más sobre la triangulación de objetos a partir de imágenes en OpenCV, se llegaron a varias conclusiones:

* Para la triangulación efectiva de objetos, es necesario el conocimiento de cierta información estandarizada al respecto de las cámaras utilizadas, principalmente la distancia que existe entre las mismas, además de que los ángulos de las tomas deben de ser los mismos entre todas las tomas. Previamente se estuvo realizando experimentación a partir de 3 tomas en donde no existía ninguna estandarización entre las tomas.
* No es necesario que existan 3 cámaras para la detección de los límites del terreno. Para la triangulación de puntos y detección de distancias de dichos puntos basta con un par de cámaras posicionadas a un lado del terreno, ya que la comparación de imágenes de 2 cámaras posicionadas una al lado de la otra basta para el desarrollo de un programa que pueda aplicar trigonometría con la información que se conoce sobre la distancia entre cámaras y también información de los lentes usados por las mismas cámaras. La idea se centra en el hecho de que, al usar 2 cámaras, una posicionada una al lado de la otra, se evidenciaría un efecto 3D al comparar una imagen con la otra, similar a como los humanos obtenemos nuestra “sensación de profundidad” en nuestra visión con nuestros ojos.
* Se decide que, al menos para fases iniciales, la delineación del terreno será únicamente con líneas rectas, y orientar el desarrollo a la detección de los vértices de la delineación del terreno, para así llegar a medir la distancia que existe en cada uno de los vértices, y así generar la vista de pájaro del terreno de forma precisa, desarrollando el programa de forma que dibuje las líneas entre cada uno de los vértices en base a la distancia que se encuentran los mismos.

Para el posterior desarrollo del programa, se confecciono un nuevo “terreno de pruebas” para probar los programas desarrollados:

A paper with red lines on it

Description automatically generated

Con este nuevo terreno de pruebas se realizaron 2 sets de tomas para distintas posibilidades de posicionamiento de cámaras:

Posicionamiento 1:

Este juego de tomas se basa en 3 imágenes tomadas desde 3 ángulos en forma de triangulo equilátero, con todas las tomas tomadas a una distancia de 52cm desde el centro del terreno, y a un ángulo de 35 grados. La idea de este set de tomas es tener quizá un acercamiento similar al original, pero en donde todas las tomas son tomas estandarizadas para tener una comparación mas limpia entre todas las imágenes.

A white paper with a red line drawn on it

Description automatically generatedA paper with a red line on it

Description automatically generatedA white paper with a red line drawn on it

Description automatically generated  
  
Posicionamiento 2:

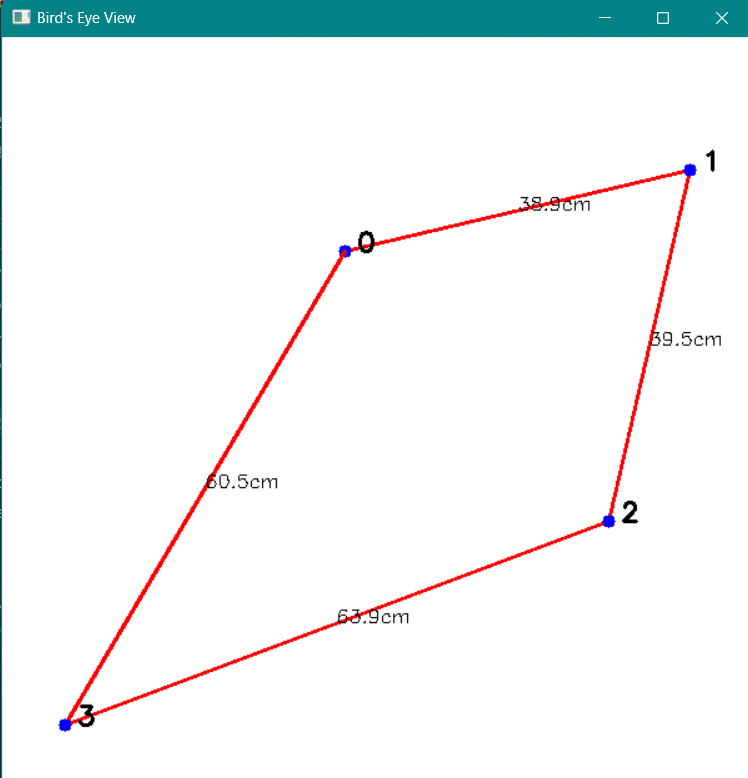
En este segundo juego de tomas se centran en únicamente 2 imágenes tomadas desde el mismo lado del terreno, ambas a una distancia de 70cm del centro del terreno y a un ángulo de 30 grados, ambas cámaras se encuentran a una distancia lateral de 20cm respecto a la otra.

 A desk with a white sheet of paper and a clock on it

Description automatically generated

Con estas imágenes serán con las que se continuaran las pruebas, ya que la visión “binocular” es lo que parece tener mas promesa en estos momentos.   
Se desarrollo un código con esto en mente, con la ayuda de OpenCV, estos son los resultados  
A screenshot of a computer

Description automatically generated



Si bien, es un gran avance, el resultado de la figura no se acerca mucho a la figura de la vida real, es posible que falte de calibrar ciertos aspectos del código, aunque lo obtenido es bastante mejor de lo que había de forma previa.