

# Sistema de Reconstrucción de Mosaicos con Fusión de Información de Profundidad (Noviembre de 2022)

Gilberth Steeven Montejo Diaz

**Resumen** - Se ha desarrollado una aplicación para el sistema operativo Ubuntu 18.04 en lenguaje Python 3.6, que permite la reconstrucción de mosaicos con fusión de información de profundidad. El software cuenta con una interfaz que permite el lanzamiento de dos subsistemas. El primer subsistema se encarga de establecer una conexión con el Kinect V1, (que es el dispositivo de adquisición de datos utilizado) y permite la captura y almacenamiento de muestras de una escena. El segundo subsistema se encarga de procesar los datos e integra un filtro de distancia, que permite seleccionar o excluir objetos de interés. El resultado final se obtiene en formato ply y se crea a partir del procesamiento de una nube de puntos, con la información adquirida.

**Índice de Términos** – Nube de Puntos, Mesh, Mosaico, Visión Artificial, Reconstrucción, Python

## I. INTRODUCCIÓN

El uso de tecnologías para la reconstrucción de entornos se ha desarrollado en diferentes áreas de la ciencia, para poder obtener datos manipulables de escenas de interés. Dentro de este proyecto, se ha buscado generar un aplicativo con fines generales de aplicación, mediante la captura y procesamiento de datos obtenidos a partir de un Kinect V1; con esto también se ha considerado que, los modelos finales cuenten con medidas de profundidad en centímetros, esto con el fin de ofrecer al usuario final, una herramienta que permita la generación de modelos, similares a la escena del espacio real.

Con el desarrollo de este proyecto, se implementa un sistema de reconstrucción de mosaicos con fusión de información de profundidad, con una interfaz gráfica y dos subsistemas: uno para la toma de datos y otro para el procesamiento de los mismos. Dentro del procesamiento de los datos, se incluye un filtro de distancia, que permite seleccionar o excluir objetos de interés a partir del ingreso de distancias en centímetros, y siempre pensado rápido de señales.

## II. MARCO TEÓRICO

### A. Fotogrametría

La fotogrametría es una técnica que permite realizar medidas 3D y generar volúmenes a partir de imágenes, y que, ha evolucionado con la tecnología y el desarrollo de algoritmos de visión artificial hasta llegar a incursionar en el campo de la reconstrucción 3D [1]. Para el caso último mencionado, se utilizan técnicas como la búsqueda de características similares en imágenes y el posicionamiento de cámaras mediante algoritmos, sin embargo, para este proyecto, se toma únicamente los píxeles centrales de la imagen RGB y sus correspondientes en el mapa de profundidad y se ubican de manera horizontal.

### B. Kinect

El kinect es un dispositivo de bajo costo desarrollado por Microsoft, inicialmente enfocado para sus consolas Xbox y luego llevado a compatibilidad con computadoras de escritorio. Este sistema puede capturar imágenes RGB y mapas de profundidad de alta resolución [2], [3].

Este sistema se divide en dos partes: un láser emisor de luz Infrarroja (IR) y la cámara de luz infrarroja. Para la detección de profundidad se utilizan patrones de luz infrarroja emitidas por el proyector IR; la reflexión de este patrón es captada por la cámara IR. Los datos capturados son transmitidos a un chip que se encarga de traducirlos y generar la imagen de profundidad.

El alcance de este sistema es de 58° en horizontal, 45° en vertical y 70° en diagonal con un rango de operación de entre 0.8m y 3.5m [4]. Para evitar que la luz del ambiente tenga influencia, dado que el emisor es de una longitud de onda exacta, entonces, existe un filtro para las longitudes de onda de 830nm sobre la cámara IR. El problema principal con el Kinect en exteriores, se halla en que, el rango de frecuencias de la luz solar es amplio en el espectro de infrarrojo. La distancia es limitada por el diodo que se utiliza como sistema de emisión de ondas.

Las aplicaciones de este dispositivo se centran en la visión artificial, y para la aplicación actual, se trabaja sobre el sistema de libre distribución Ubuntu y como driver la librería libfreenect [5].

### C. Color en las Nubes de Puntos}

Para obtener el color en las nubes de puntos basta con habilitar la imagen RGB y obtener la equivalencia de la nube respecto a la imagen RGB. El problema se basa en que la imagen RGB no coincide con la nube de puntos debido que no están ubicadas en el mismo lugar, por lo tanto, las imágenes capturadas están ligeramente desviadas. Sin embargo, al utilizar la librería de freenect, este problema es corregido.

### D. Teoría para superponer imágenes obtenidas por el kinect V1

Para poder determinar la cantidad de datos que se deben de cada una de las imágenes RGB se analiza el sistema de adquisición de datos ubicado en el espacio tal y como se muestra en Fig. 1. En el modelo dispuesto se tiene el sistema coordenado (x,y) y un distancia d desde el foco del sensor hasta el punto de interés. El ángulo  $\alpha$  indica la inclinación de la línea d respecto al eje coordenado y.

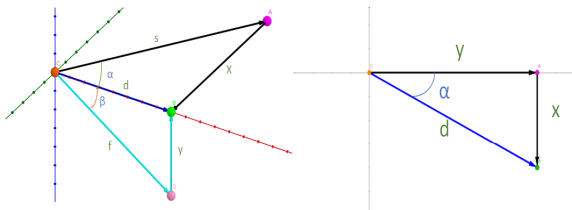


Fig. 1. Modelo de adquisición de datos ubicado en el espacio

Para determinar la ubicación del pixel, se debe tener en cuenta que el máximo ángulo horizontal es de  $57^\circ$ , lo que quiere decir que para una resolución de 640px se tienen  $0.089^\circ/\text{px}$ ; en cuanto al máximo ángulo vertical, este es de  $43^\circ$ , de manera que para una resolución de 480px, corresponden  $0.0896^\circ/\text{px}$ . Sabiendo esto, y asumiendo que el punto B es el centro de la imagen (Ver Fig. 1), entonces:

$$B = (320,240)$$

De tal manera que el ángulo  $\beta$  viene dado por la distancia entre  $B_x$  y  $A_x$  y multiplicado por la resolución vertical correspondiente a cada pixel. Para el caso de  $\alpha$  funciona de igual manera, pero será la distancia entre  $B_y$  y  $A_y$  y la resolución horizontal. En ecuaciones, se tiene:

$$\beta = |B_x - A_x| * 0.089^\circ/\text{px} \quad (1)$$

$$\alpha = |B_y - A_y| * 0.0896^\circ/\text{px} \quad (2)$$

Por otro lado, se tiene un valor  $\lambda$ ; la forma de calcular este se da dentro de la librería de libfreenect:

$$\lambda = \frac{1}{(-3.07 * 10^{-3} * \text{ValorKinect} + 3.33)} \quad (3)$$

Y esta ecuación sirve para convertir a metros los valores de profundidad medidos por el sensor de profundidad del Kinect. Esta ecuación se limita al rango de valores de 500 a 1023 entregados por el dispositivo. Ahora sí, retornando al modelo

propuesto en la Fig. 1, se determinan las ecuaciones para el mismo:

$$d = \lambda \cos(\beta) \quad (4)$$

$$x = d \sin(\alpha) \quad (5)$$

$$y = d \cos(\alpha) \quad (6)$$

Para las ecuaciones (4), (5), (6) los valores se obtienen en metros.

### III. ESTADO DEL ARTE

Los trabajos de investigación en el campo de la reconstrucción tridimensional son de aplicaciones generales y evidencian la aplicación de la visión artificial en los diferentes campos del conocimiento, como la agricultura en donde una de sus posibles aplicaciones se basa en la reconstrucción de malas hierbas para un análisis geométrico de las mismas [6]; en campos como la arqueología y sus derivados eventos turísticos [1] sirve como referencia para ofrecer mejores experiencias.

Para las aplicaciones mencionadas anteriormente, se parte de fotografías y mediante el posicionamiento de cámaras [7] se generan nubes de puntos representativas de una escena, que posteriormente son utilizados para generar las mallas correspondientes [8]. El modelo generado se puede exportar a diferentes formatos y plataformas, y haciendo uso de las propiedades del procesador de Texto LaTeX, la generación de documentos interactivos con modelos 3D en formato Universal 3d [9]. Es de resaltar que los resultados finales tienen medidas en píxeles.

En aplicaciones del kinect como sensor, se utilizan sistemas muy similares al anteriormente mencionado, pero diferenciado en los resultados finales por ofrecer resultados con medidas en milímetros tal y como se hace en [2], [10], [11]; bajo el mismo sensor se desarrollan técnicas de ubicación de objetos en el espacio y determinación de distancias a partir del mapa de profundidad.

Para la reconstrucción de entornos en tres dimensiones existen diferentes métodos, sin embargo, todas se reducen en el mismo principio: medir la distancia desde el sensor hasta el punto observado; esta medición genera puntos 3D de posición. Los datos obtenidos también pueden denominarse como nube de puntos [12]. Un aspecto que se debe tener muy en cuenta es las condiciones del entorno como la iluminación, el ángulo de captura y la velocidad de captura, pues estos, pueden afectar los resultados finales.

Existen dos técnicas de reconstrucción: reconstrucción pasiva y activa. En cuanto a la técnica de reconstrucción pasiva se parte de escáneres que devuelven información en forma de puntos 3D. Dentro de estos tipos de escáneres es posible encontrar los LIDAR y los Sensores de Profundidad. Para la técnica de reconstrucción pasiva se parte de un conjunto de imágenes (videos incluidos) y mediante la estimación de la posición de cámaras, se establece y generan los modelos de puntos 3D. También existe un tercer método, aunque menos desarrollado, y consiste en la combinación de las dos técnicas anteriormente mencionadas [13].

Algunos sistemas de reconstrucción de entornos pueden llegar a ser demasiado costosos, dentro de estos se encuentran los que involucran satélites. Cuando se involucran satélites las medidas del objeto a reconstruir se pueden estimar a partir de la longitud, la latitud y la altitud. Dentro de estas técnicas, es posible encontrar que se apliquen métodos diferentes a los mencionados anteriormente; estos se pueden basar en la consorción de pares de imágenes y la estimación de gradientes de energía [14]. Con estos datos procesados, se analiza posteriormente el peso de cada uno de los resultados y mediante sistemas de Inteligencia artificial se determina si los puntos pertenecen o no al objeto que se está reconstruyendo.

Tal y cómo se mencionaba anteriormente, la inteligencia artificial también es aplicada en el campo de la reconstrucción de entornos y objetos tridimensionales; con esto, se da la introducción y tal y como hace [14], se utilizan redes neuronales profundas para la estimación de puntos pertenecientes a un modelo y además de ello para aumentar la precisión en la estimación de la profundidad asociada a cada punto de la nube de puntos generada [15].

#### IV. METODOLOGÍA

Para el desarrollo de la aplicación de este proyecto se utiliza el modelo en cascada (Fig. 2) que consiste en encadenar las diferentes etapas de la ejecución del proyecto y finalmente, establecer una retroalimentación del producto final. Este modelo se divide en cinco (5) pasos:

- **Análisis:** Es la fase de preparación del proyecto y el objetivo es dar el punto de partida para el desarrollo del proyecto. En esta etapa se plantean las necesidades y los objetivos a alcanzar, para con ello generar un banco de requisitos que debe cumplir el software.
- **Diseño:** Se plantean la estructura fundamental del código de la aplicación y con ello la organización de los elementos que debe contener el proyecto y cómo se relacionan entre sí. En esta etapa se debe tener en cuenta el diseño de la interfaz y siempre se debe pensar en el usuario final.
- **Implementación:** En el diseño, se da un bosquejo de lo que se requiere, en esta etapa se define el lenguaje y se integra todo lo diseñado en un código.
- **Verificación:** Comprobación y ejecución de todo el código final, y verificación de funcionamiento.
- **Mantenimiento:** Analizar los resultados obtenidos, y si es necesario, realizar los cambios pertinentes y dar por terminado el proyecto.

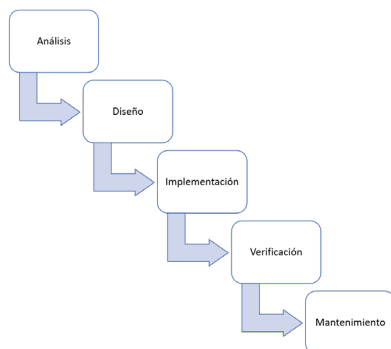


Fig. 2. Metodología de desarrollo de Software aplicada

#### V. RESULTADOS

Se presentan y analizan los resultados del proyecto realizado; para esto, se realiza la ejecución completa del software desarrollado y se mostrará el paso a paso de cada una de las dos funcionalidades que este integra. Para llegar a este capítulo se han pasado las tres últimas fases de la metodología de desarrollo. En la fase de implementación, se ha desarrollado el software en lenguaje Python 3.6 y en el sistema Operativo Ubuntu 18.04. Para la parte de verificación del sistema se han realizado diferentes pruebas, y luego de aplicar algunas modificaciones obtenidas en la fase de mantenimiento, se obtienen los resultados que se presentarán a continuación. Para efectos de una mejor presentación de los resultados, la interfaz gráfica de usuario será presentada inicialmente, seguida de la interfaz de adquisición de datos y finalmente, la interfaz de procesamiento y visualización del modelo final.

##### A. GUI Desarrollada

La interfaz principal del sistema (Fig. 3) cuenta con 4 botones, el primero permite al usuario acceder al sistema de adquisición de datos, el segundo, cuyo nombre dice “Modelo 3D”, se encarga de lanzar la interfaz para procesar los datos y generar los modelos finales. El tercer botón permite la configuración del tamaño de pantalla y el último termina la ejecución del software.

La interfaz principal del sistema (Fig. 3) cuenta con 4 botones, el primero permite al usuario acceder al sistema de adquisición de datos, el segundo, cuyo nombre dice “Modelo 3D”, se encarga de lanzar la interfaz para procesar los datos y generar los modelos finales. El tercer botón permite la configuración del tamaño de pantalla y el último termina la ejecución del software.



Fig. 3. Interfaz principal

##### B. GUI: Adquisición de Datos

Tal y como se había planteado en la fase de diseño, la interfaz gráfica para la adquisición de datos (Fig. 4) contiene un botón para la selección del directorio donde se desea guardar la información. Este botón abre una ventana emergente que permite el desplazamiento dentro del explorador de archivos y seleccionar la carpeta en la que deseamos se guarden las muestras del proyecto en desarrollo. Una entrada de tipo texto,

permite el ingreso del nombre del proyecto y su validación para evitar sobre escribir proyectos capturados.

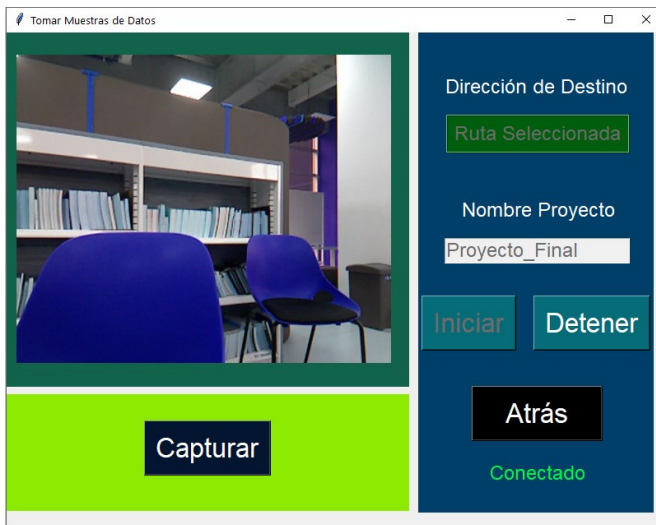


Fig. 4. Interfaz gráfica del sistema de adquisición de datos

Por otro lado, están los botones de acción, el botón de Iniciar se encarga de validar que la información ingresada sea válida para proceder a la toma de muestras. Se puede observar que, hay un apartado que nos muestra lo que el sensor está enfocando en tiempo real y se tiene a la parte inferior derecha un letrero que dice “Conectado” y hace referencia al estado del Kinect. Mientras el letrero indique un estado positivo del Kinect será posible almacenar datos, si este cambia a “Sin conexión” el software emitirá la alerta respectiva y no dejará seguir adquiriendo datos. La Adquisición de datos se realiza de manera manual, cada vez que se oprime el botón de “Capturar”; Aunque en la interfaz no se muestran los datos de profundidad, el sistema se encarga de guardar tanto el mapa de colores, como el mapa de profundidad.

Para un pequeño proyecto realizado con el software, en el que se requerían 5 muestras de la escena, los datos almacenados quedan tal y como se observa en la Fig. 5; en esta podemos observar que los datos se guardan con los nombres clave (depth e img) propuestos en las fases de análisis y diseño y con los formatos (csv o archivo de valores separados por comas de Microsoft Excel para los mapas de profundidad y mapas de color respectivamente). También los nombres incluyen la numeración requerida para el procesamiento de los datos.

depth0	31/10/2022 5:09 p. m.	Archivo de valores separados por comas de Microsoft Excel	7,500 KB
depth1	31/10/2022 5:09 p. m.	Archivo de valores separados por comas de Microsoft Excel	7,500 KB
depth2	31/10/2022 5:09 p. m.	Archivo de valores separados por comas de Microsoft Excel	7,500 KB
depth3	31/10/2022 5:09 p. m.	Archivo de valores separados por comas de Microsoft Excel	7,500 KB
depth4	31/10/2022 5:09 p. m.	Archivo de valores separados por comas de Microsoft Excel	7,500 KB
img0	31/10/2022 5:09 p. m.	Archivo .JPG	97 KB
img1	31/10/2022 5:09 p. m.	Archivo .JPG	91 KB
img2	31/10/2022 5:09 p. m.	Archivo .JPG	82 KB
img3	31/10/2022 5:09 p. m.	Archivo .JPG	85 KB
img4	31/10/2022 5:09 p. m.	Archivo .JPG	93 KB

Fig. 5. Datos almacenados para un proyecto de 5 muestras

### C. GUI: Procesamiento de los Datos

Esta interfaz (Fig. 6) tiene más elementos que la anterior, sin embargo, no todos son requeridos para la obtención de un Mosaico con Fusión de información de profundidad; dentro de los valores obligatorios se encuentran las rutas de origen de archivos y destino del archivo final, el nombre del proyecto, y la distancia entre las capturas. El filtro es opcional, en caso de

no ingresar algún valor, se tomará como referencia para la distancia máxima el valor de 400 centímetros y de 50 para la mínima. La unidad de referencia tanto para el filtro de distancia como para la distancia entre capturas, es centímetros.



Fig. 6. Interfaz Gráfica de Usuario para el procesamiento de Datos y generación del Mosaico con fusión de información de profundidad

Al ingresar los datos del proyecto que se desea procesa y obtener un modelo 3D manipulable, se oprime el botón de Generar; una vez realizada esta acción, el sistema iniciará a procesar los datos con los valores ingresados. El tiempo de procesamiento dependerá mucho de las características del computador sobre el cual se está ejecutando y de la cantidad de datos que tenga que procesar, pues no es lo mismo, generar un modelo con 5 mapas de profundidad con sus respectivas imágenes, a generar un modelo a partir de 100 mapas de profundidad con sus respectivas imágenes.



Fig. 7. Mosaico con fusión de Información de Profundidad con muestras cada 20cm y sin filtro de distancia

Al finalizar el procesamiento de los datos, se abrirá una ventana (Fig. 7) en donde se visualizará el modelo final generado y permitirá su observación. Ahora, ejecutando la misma escena, pero esta vez con un filtro de distancia para tomar todos los objetos en un rango de 150cm y 300cm para quitar una de las dos sillas que se observan, y los objetos que se encuentran más al fondo de la escena (Ver Fig. 8).



Fig. 8. Mosaico con fusión de Información de Profundidad con muestras cada 20cm y filtro de distancia de 150cm y 300cm para mínimo y máximo respectivamente.

Para la concatenación de estos datos, se ha utilizado un medidor de distancia láser de la marca Milesey que permite la medición de distancia de entre 0 y 100m con una precisión de  $\pm 1.5\text{mm}$ . La distancia desde el Kinect hasta la silla que ha sido eliminada es de 172cm. Para el caso de los objetos del fondo, la distancia medida es de 382 cm. Sin embargo, es de destacar que el rango del Kinect es de 50cm a 400cm. Por tanto, la medida queda justo dentro del rango de medición del sistema.

## VI. CONCLUSIONES

Con el desarrollo de este proyecto se obtiene como producto final un aplicativo para el sistema operativo Ubuntu 18.04 que permite capturar datos de color y profundidad de una escena a partir del Kinect V1 para su posterior procesamiento de datos. Dentro del procesamiento de datos se analiza una estructura de escena general, y a partir de dicho análisis se genera una ecuación para generar la superposición de imágenes de un mosaico para obtener un modelo tridimensional a partir de este y la fusión con la información de profundidad.

El desarrollo de la ecuación para la superposición de datos, se puede considerar un filtro que se encarga de eliminar los datos coincidentes entre dos fotografías a partir de la estimación de posición desde donde se han capturado. Al eliminar información compartida entre dos conjuntos de datos, se reduce el tiempo de procesamiento final, pues el número de valores es menor, comparado con la cantidad de valores que suman las dos imágenes originales. Dado que se asume que la posición del sensor de profundidad y la cámara se encuentran en la misma posición en el dispositivo de adquisición de datos, es posible eliminar los mismos datos de la matriz de color y de la matriz de valores de profundidad.

Teniendo la matriz final, resultante de la superposición de imágenes, y de igual manera, la matriz final de valores de profundidad correlacionados al vector de color, la nube de puntos ha sido creada; la nube de puntos, radica en la forma en cómo se grafican los datos. Para graficar una nube de puntos, se debe graficar los valores de la matriz de profundidad (x, y son la posición en el plano, y z es la profundidad de la escena), y

utilizar los valores de la matriz de color, para indicar la tonalidad del punto graficado.

A partir de la nube de puntos, fue posible obtener un mesh o malla; para ello se procesaron los datos, de tal manera que, el primer paso fue crear el conjunto de vóxels representativos de la escena. El vóxel se crea a partir de la fusión de la información de profundidad con la información de color, que esencia, es lo que indica el título de este proyecto. Los Para crear el mesh final, se debió realizar el proceso de estimación de vectores normales de la escena, pues, con el modelo de datos inicial, estos valores no existían. Los valores normales que describen la superficie fueron requeridos, por el método de reconstrucción de superficies utilizado (método de Poisson). El formato del archivo final es ply que es compatible con diferentes plataformas de modelado tridimensional.

La interfaz gráfica desarrollada cumple con los requerimientos necesarios planteados en el análisis y diseño del software, y permiten una interacción amigable para desarrollar proyectos de generación de Mosaicos con fusión de profundidad e integra también, uno de los detalles más curiosos de esta investigación: el filtro de distancia. El filtro de distancia, se implementa con la interfaz, pero a nivel de ejecución, se aplica antes de obtener una nube de puntos final, lo que permite reducir aún más los tiempos de procesamiento de datos, para modelos con filtros seleccionados. El filtro cumple su objetivo de permitir seleccionar un rango de distancias en centímetros definido por el usuario, para seleccionar o excluir objetos de interés.

Adicional a esto, es de comentar que, la posición de los puntos de la nube generada es relativa y se toma desde el foco del sensor de profundidad hasta el objeto reconstruido. Esto sucede porque el Kinect no tiene un sistema de posicionamiento global, pues no es requerido y tampoco necesario para el enfoque del proyecto. Por otro lado, el mesh generado, conserva las medidas iniciales de la escena, aunque es difícil realizar una respectiva medición de este, esto es comprobable a partir del filtro de distancia que se aplica.

Finalmente, es importante resaltar que, tras la realización de este proyecto, se comprende la importancia del desarrollo de técnicas de visión artificial para el procesamiento rápido de datos y su implementación en herramientas que puedan llegar a hacer más fácil la vida de las personas; aunque este proyecto no se realiza para una aplicación puntual, en el apartado de proyectos futuros, se proponen varios proyectos de aplicación que pueden llegar a cumplir tal propósito.

## VII. REFERENCIAS

- [1] J. Caro, "Fotogrametría y modelado 3D: un caso práctico para la difusión del patrimonio y su promoción turística," 2012.
- [2] Y. Wan, J. Wang, J. Hu, T. Song, Y. Bai, and Z. Ji, "A study in 3D-reconstruction using kinect sensor," in *2012 International Conference on Wireless Communications, Networking and Mobile Computing, WiCOM 2012*, 2012. doi: 10.1109/WiCOM.2012.6478374.
- [3] V. E. L. Ang, F. E. O. Decinal, N. B. Linsangan, and J. J. Adtoon, "Volume Approximation Using Kinect

Sensor,” in *2021 IEEE 13th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment, and Management, HNICEM 2021*, 2021. doi: 10.1109/HNICEM54116.2021.9731913.

- [4] J. Han, L. Shao, D. Xu, and J. Shotton, “Enhanced computer vision with Microsoft Kinect sensor: A review,” *IEEE Trans Cybern*, vol. 43, no. 5, pp. 1318–1334, Oct. 2013, doi: 10.1109/TCYB.2013.2265378.
- [5] R. A. El-Laithy, J. Huang, and M. Yeh, “Study on the use of Microsoft Kinect for robotics applications,” in *Record - IEEE PLANS, Position Location and Navigation Symposium*, 2012, pp. 1280–1288. doi: 10.1109/PLANS.2012.6236985.
- [6] Andújar D, Ribeiro A, Bengoechea JM, San-Martín C, Fernández-Quintanilla C, and Dorado J, “RECONSTRUCCIÓN 3D DE MALAS HIERBAS UTILIZANDO CÁMARAS DE PROFUNDIDAD,” 2015.
- [7] R. Ruoff, “Aplicación para la obtención de un mapa 3D de una escena con un dispositivo Android,” 2015.
- [8] D. W. McClune, “Joining the dots: Reconstructing 3D environments and movement paths using animal-borne devices,” *Animal Biotelemetry*, vol. 6, no. 1, May 2018, doi: 10.1186/s40317-018-0150-6.
- [9] P. Rodríguez, “La fotogrametría digital automatizada frente a los sistemas basados en sensores 3D activos,” *Revista de expresión gráfica arquitectónica*, pp. 100–111, 2012.
- [10] L. Valhma, “3D reconstruction using Kinect v2 camera,” 2016.
- [11] E. Cinar, “3D Reconstruction Technique with Kinect and Point Cloud Computing, AWERProcedia Information Technology & Computer Science,” 2013. [Online]. Available: <http://www.world-education-center.org/index.php/P1754>. Available from: <http://www.world-education-center.org/index.php/P-ITCS>
- [12] A. Harjoko, R. M. Hujja, and L. Awaludin, “Low-cost 3D surface reconstruction using Stereo camera for small object,” in *2017 International Conference on Signals and Systems (ICSigSys)*, 2017, pp. 285–289. doi: 10.1109/ICSIGSYS.2017.7967057.
- [13] R. Siv, I. Ardiyanto, and R. Hartanto, “3D human face reconstruction using depth sensor of Kinect 2,” in *2018 International Conference on Information and Communications Technology (ICOIACT)*, 2018, pp. 355–359. doi: 10.1109/ICOIACT.2018.8350702.
- [14] H. Chen, W. Chen, and T. Gao, “Ground 3D Object Reconstruction Based on Multi-View 3D Occupancy Network using Satellite Remote Sensing Image,” in *2021 IEEE International Geoscience and Remote Sensing Symposium IGARSS*, 2021, pp. 4826–4829. doi: 10.1109/IGARSS47720.2021.9554612.
- [15] S. Kamyab, H. Taghadosi, and Z. Azimifar, “A Deep Learning Embedded System for 3D Human Face Shape Reconstruction From a Single Image,” in *2022 27th International Computer Conference, Computer Society of Iran (CSICC)*, 2022, pp. 1–6. doi: 10.1109/CSICC55295.2022.9780484.

## VIII. AUTORES

### A. Gilberth Steeven Montejó Díaz



Nació el 14 de junio del 2000 en el municipio de La Capilla Boyacá. Cursó la primaria, y secundaria en la Institución Educativa La Candelaria en su municipio natal, en donde obtuvo el título de bachiller académico en el año 2016. En el mismo año se graduó como técnico de sistemas del SENA. En el año 2018 ingresó a la Universidad Santo Tomás en donde cursa actualmente el programa de Ingeniería Electrónica.