

SISTEMA DE DIRECCIONAMIENTO DE VEHICULOS MEDIANTE CODIGOS ARUCOS

Cristian Steeven Maquilon Mejía - T00058083

Fabricio Lambis Orozco – T00057980

Deimer Javier Padilla Méndez – T00057851

Samuel Eduardo Echenique Pérez - T00057843

Anyelina Marcelo Vergara - T00054405

Yarling Patricia Salas Borja - T00058215

PROYECTO DE VISIÓN ARTIFICIAL

Presentado a:

RAÚL ANDRES VARGAS RAMIREZ

Resumen: El sistema de direccionamiento diseñado tiene por objetivo guiar a un objeto móvil mediante un espacio de trabajo preestablecido, enviando mediante señales de radiofrecuencia los giros e intersecciones a atravesar necesarias para llegar de un punto a otro y devolverse. Esto fue posible mediante el trabajo conjunto de una cámara ubicada perpendicularmente sobre el área de trabajo y códigos arucos para identificar y diferenciar de manera satisfactoria los objetos.

1. INTRODUCCIÓN:

La visión artificial es un campo de la Inteligencia artificial que le brinda al usuario una amplia gama de información a través de las imágenes o videos digitales. Este campo entrena máquinas para la realización de funciones a través de cámaras y datos algorítmicos [1]. Uno de los tantos usos de la visión artificial puede ser el guiado de un vehículo a través de un espacio previamente conocido, esto mediante una cámara que le transmitirá la información del camino al robot móvil, debido a que la cámara tomará una posición de referencia gracias a la alineación de marcas visuales establecidas [2].

Por ello, este documento propone un sistema de direccionamiento de robots móviles haciendo uso de marcadores Arucos y el uso de una cámara que permita la visualización de la pista donde se encontrarán los robots.

2. DESARROLLO Y RESULTADOS:

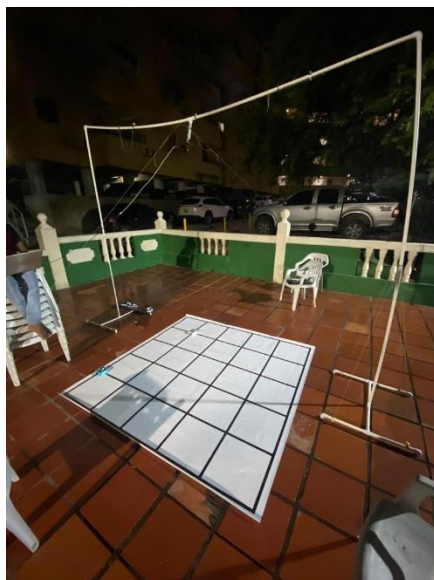
2.1 DETECCIÓN DE ARUCOS:

Previo a la implementación de los robots móviles, es necesario realizar un montaje el cual permitiera la detección de los marcadores Arucos, con el fin de obtener una posición en un sistema de referencia global fijo, para esto se empleó la cámara del móvil Samsung Galaxy A52s la cual cuenta con una cámara principal de 64MP, además de contar con la opción de gran angular, en donde le brinda al usuario una amplitud de detección de 123° lo cual era una ventaja para nosotros, debido a que la distancia a la que se debía encontrar la cámara de la superficie era muy alta.

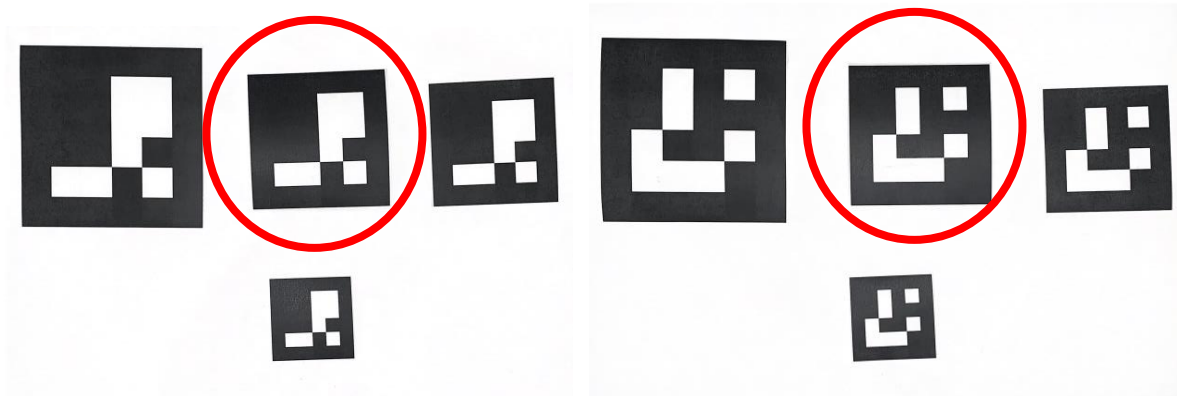
Para lograr satisfactoriamente la detección de los marcadores aruco realizamos pruebas empíricas teniendo ya el montaje donde iría el cámara realizado. El montaje contaba de unos tubos de PVC que formaban una especie de marco de en donde estaría la cámara centrada con el espacio de trabajo cuyas medidas eran de 2x2m.



- Altura a la que se encontraba la cámara: 2.5m



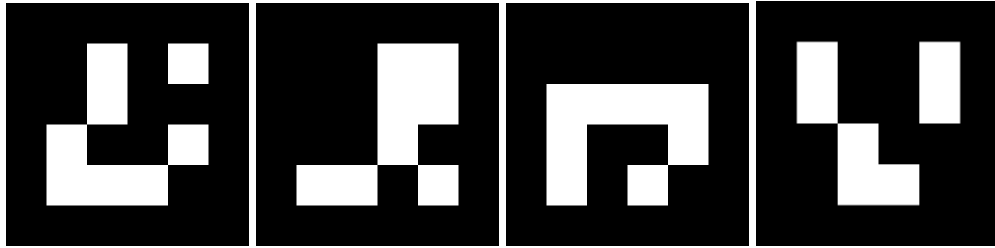
Luego de tener el montaje realizado se procedió a la impresión de 4 distintos tamaños de marcadores arucos para realizar las pruebas de la detección de estos mismos. Logrando de esta manera la selección del mejor tamaño de aruco el cual fuera capaz de detectar la cámara y cupiera en los 10x10cm de tamaño del carro. El tamaño seleccionado fue el siguiente:



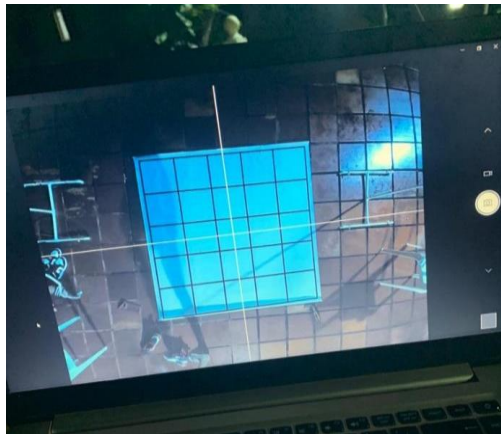
Evidencia del montaje del marcador aruco en el robot móvil:



Códigos arucos generados para la detección de la visión artificial:



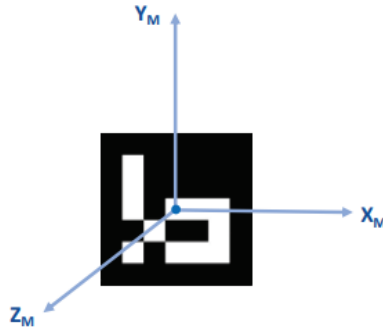
El sistema de referencia global será generado mediante a visión artificial para luego hacerse coincidir con el sistema de referencia móvil (en este caso las cuadrículas del pendón) y que el movimiento de los códigos arucos en la visión sea coincidente con el de los robots en el espacio de trabajo. Este proceso fue realizado manualmente



Luego de todo este proceso, se realizó satisfactoriamente la detección de los marcadores aruco y de la coincidencia entre los sistemas.

2.2 MAPEADO DE LOS MARCADORES ARUCOS:

Para llevar a cabo el direccionamiento de los robots móvil se emplearon códigos Arucos, los cuales son una especie de marcadores cuadrangulares compuestos por un borde negro y una matriz binaria interna que determina un identificador [3]. Para realizar satisfactoriamente este proceso es necesario conocer la orientación del marcador con respecto a un sistema de referencia que fuese común entre el vehículo y la visión artificial. Para resolver este problema se decidió pegar el marcador aruco al vehículo de tal manera en que los ejes del vehículo y del marcador coincidieran.



De manera que, si el vehículo se desplazaba hacia arriba o hacia abajo, este movimiento coincidiera con el eje Y del marcador o si el vehículo se desplazaba a la izquierda o la derecha, estos movimientos coincidieran con el eje X del marcador, mientras que el eje Z del marcador sería perpendicular a la cámara.

Realizando este proceso y trabajando en conjunto con la detección de los arucos ya podíamos detectar de manera satisfactoria la posición y orientación del código aruco con respecto al sistema de referencia global ubicado en el monitor.

2.3 FUNCIONAMIENTO DE LA VISIÓN ARTIFICIAL EN EL PROCESO:

Para comprender el trabajo de la visión artificial en el sistema es necesario realizar una narrativa del proceso, la cual es la siguiente:

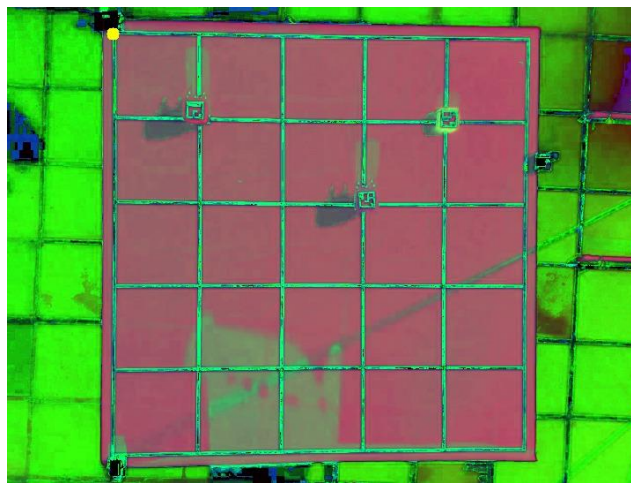
- Los robots móviles inicialmente se encontrarán estáticos en el espacio de trabajo, mediante la visión artificial se solicitará un determinado objeto en una estación mediante un clasificador de colores, luego con los códigos arucos se identifica cuál vehículo se encuentra más cercano a la estación donde se encuentra el objeto solicitado, luego se realiza el camino de la trayectoria del robot a la estación del objeto requerido y luego de la estación del objeto requerido a la estación del objeto solicitado.

Estos caminos que estará detectando la visión artificial serán transmitidos mediante radiofrecuencia al vehículo que lo requiera de manera cartesiana, es decir las intersecciones tanto en X como en Y que debe recorrer el objeto de un punto a otro.

2.4 DETECCIÓN DE COLORES:

Para la detección de colores lo primero que se hace es convertir la imagen a HSV (espacio de color de matriz de tonos) usando la función `cv2.cvtColor()`. Esta función toma la imagen en BGR (Blue Green Red) como su primer argumento y devuelve la imagen convertida a HSV. luego de esto se establecen los límites para detectar un color específico. Estos límites se proporcionan como una lista de tres elementos conteniendo los valores mínimos y máximos de los canales de color HSV. por ejemplo el caso del color rojo sería: `lower_red = np.array([150,50,50])` y `upper_red = np.array([180,255,255])` respectivamente.

utilizando la función `cv2.inRange ()` para crear una máscara binaria. Esta función toma la imagen HSV como su primer argumento, luego los límites inferior y superior como sus argumentos siguientes y devuelve una máscara binaria, se puede utilizar la función `cv2.bitwise_and()` para aplicar la máscara a la imagen original. Esta función toma la imagen original como su primer argumento y la máscara binaria como su segundo argumento y devuelve la imagen original con los píxeles fuera del rango especificado marcados como cero. Por último, la función `cv.findContours()` para encontrar los contornos del color que se haya especificado y `cv.contourArea ()` para calcular el área de los contornos en este paso podemos agregar algún condicional para hacer una limpieza de los colores y contornos detectados por la cámara.



2.5 POSICIONAMIENTO Y ORIENTACIÓN DE LOS VEHICULOS:

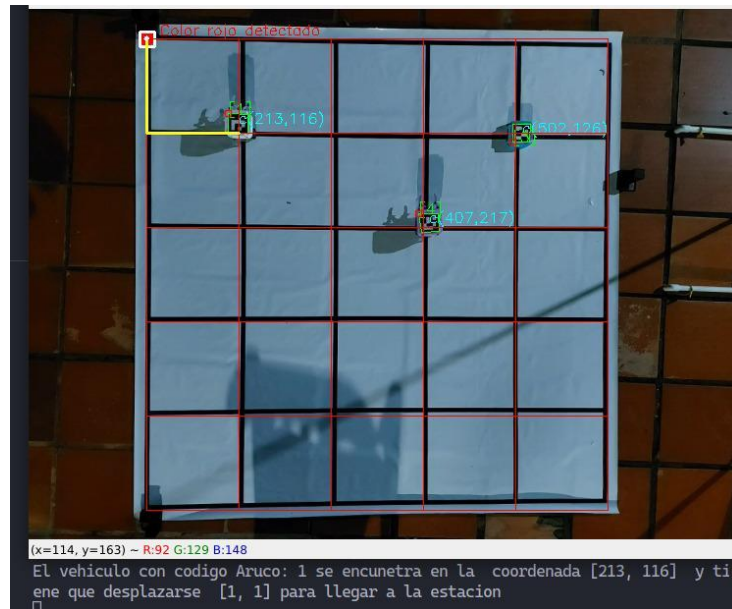
Como el sistema global (visión artificial) y el sistema móvil (espacio de trabajo) son coincidentes en el monitor, podemos realizar el análisis del direccionamiento de un vehículo de un punto a otro en el espacio como se es solicitado en la narrativa del proceso, dividiendo la pantalla en cuadrículas similares a la pista donde transita el vehículo, dichas cuadrículas se encuentran etiquetadas con sus coordenadas como puntos cartesianos en donde la intersección (0,0) se refiere a la esquina superior izquierda y la intersección (5,5) hace referencia a la esquina inferior derecha, siendo el orden (columnas, filas). Después de esto hacemos uso de la detección de colores con la que se identifica la estación o el punto de llegada del vehículo, vemos el siguiente ejemplo:



La detección del color verde en la estación de la intersección (0,5) indica que el vehículo deberá moverse hasta aquel punto, para el posicionamiento y el cálculo de las distancias se deben tener en cuenta la geometría taxicab o geometría del taxista, usada cuando el tránsito no es libre, como en este caso que el movimiento del vehículo se encuentra restringido por las líneas de la cuadrícula.

Una geometría de taxi es una forma de geometría en la que la función de distancia habitual o métrica de la geometría euclidiana se reemplaza por una nueva métrica en la que la distancia entre dos puntos es la suma de las diferencias absolutas de sus coordenadas cartesianas .

Esta geometría a diferencia de la euclidiana permite calcular la distancia dadas las restricciones del carro en este caso líneas rectas de 15cm de largo y podemos ver las ordenes que se le deben mandar al carro para que llegue al destino como en el siguiente ejemplo:



```

1 coordenadas, ordenes = intersecciones.Cercano(arucos_lista, estacion)
2 cordena_aruco = correccion(coordenadas, arucos)
3 id_aruco = list(arucos.keys())[list(
4     arucos.values()).index(cordena_aruco)]
5 print(
6     f"El vehiculo con codigo Aruco: {id_aruco} se encuentra en la coordenada {cordena_aruco} y tiene que desplazarse {ordenes} para llegar a la estacion")

```

En este se aprecia la posición en coordenadas píxel del vehículo con aruco de ID 1 que se encuentra más cercano a la estación solicitada por el color rojo ubicada en (0,0) de nuestro sistema cartesiano, y en la parte baja vemos las ordenes [x,y] para llegar a ese punto, en este caso debe avanzar 1 en x y 1 en y para llegar a la estación.

3. CONCLUSIONES:

En el desarrollo de este proyecto se implementó un sistema que permitía el direccionamiento de un robot móvil en un espacio de trabajo mediante la visión artificial, en cuanto a los resultados esperados de la visión artificial fueron satisfactorios, debido a que el programa diseñado sí fue capaz de identificar la cantidad de intersecciones en X y en Y que debía recorrer el vehiculo, además de identificar los colores de solicitud en las estaciones. Lastimosamente el proceso de sinérgico entre la visión artificial y el robot móvil no fue el correcto, debido a errores en la programación del movimiento del vehiculo y en las tarjetas controladoras de este mismo, por ende, el proceso completo no fue validado con total satisfacción.

Como trabajo futuro nos enfocaremos en la unión entre los 4 vehículos móviles y el sistema de visión artificial debido a que solo fuimos capaz de realizar las pruebas con un vehiculo. Se desea también la realización de una interfaz amigable que le brinde un fácil entendimiento del sistema al usuario y al operario en el uso de la visión artificial.

REFERENCIAS BIBLIOGRAFICAS:

- [1] ¿Qué es la Visión Artificial? | IBM. (2022). Retrieved December 5, 2022, from Ibm.com website: <https://www.ibm.com/co-es/topics/computer-vision>
- [2] RdR. (2020, February 5). Vehículos AGV, los vehículos de guiado automático inteligentes. Retrieved December 6, 2022, from REVISTA DE ROBOTS website: <https://revistaderobots.com/robots-y-robotica/robot-agv-aiv-los-vehiculos-de-guiado-automatico-inteligentes/>
- [3] Sara, Fernández García, Álvaro, Álvarez García, Ignacio, & Rafael, R. (2019). Localización de robots móviles en entornos industriales usando un anillo de cámaras. XL Jornadas de Automática, 849–858. <https://doi.org/978-84-9749-716-9>