

Modelado de visión por computadora y control de movimiento para la navegación autónoma de un robot móvil en entornos dinámicos

Enrique Sibaja Galindo¹, Florencia Bacilio López², Roberto Quintana Ramírez³, Jose Alberto Carrasco Francisco⁴, Armando Levid Rodríguez Santiago⁵

DOI: 10.56643/rcia.v3il.174

Como citar este artículo: Sibaja-Galindo E., Bacilio-López E., Carrasco Francisco J.A., Quintana-Ramírez R., Rodríguez-Santiago A.L. (2024) Modelado de visión por computadora y control de movimiento para la navegación autónoma de un robot móvil en entornos dinámicos. Revista Científica de Ingenierías y Arquitectura. 3(1). 44-60. DOI: <https://doi.org/10.56643/rcia.v3il.174>

RESUMEN

Este escrito aborda el desarrollo de un robot móvil que pueda interactuar con entornos dinámicos mediante la integración de visión por computadora y control de movimiento. Al respecto, se destaca la importancia de la detección y seguimiento de objetos en tiempo real para lograr una navegación segura en aplicaciones prácticas. Hoy en día, a pesar de los avances en visión por computadora y control robótico, persisten desafíos relacionados con la adaptación a entornos impredecibles. Por ende, la propuesta presentada consiste en un sistema que combina técnicas de procesamiento de imágenes y control robótico. El diseño del prototipo utiliza *hardware* y *software* libre, como lo son Python, Arduino, OpenCV y Numpy. La programación en Arduino se emplea para controlar un vehículo con cuatro motores mediante un módulo Bluetooth HC-05, mientras que el desarrollo de visión por computadora realiza el seguimiento de objetos de diferentes colores en tiempo real, enviando comandos a Arduino según el color detectado. El proyecto se considera un avance en robótica móvil, en tanto desarrolla un sistema integrado que permite a los robots navegar de manera segura en entornos dinámicos. En este caso, la combinación de visión por computadora y control de movimiento se presenta como una solución práctica respaldada por pruebas detalladas.

PALABRAS CLAVE: *Visión por computadora, Python, Arduino, OpenCV, Numpy.*

ABSTRACT

This writing addresses the development for the interaction of a mobile robot in dynamic environments by integrating computer vision and motion control. It emphasizes the importance of real-time object detection and tracking to achieve safe navigation in practical applications. Despite advancements in computer vision and robotic control, challenges persist in adapting to unpredictable environments. Therefore, the proposed solution consists of a system that combines image processing techniques and robotic control. The prototype's design utilizes open-source *hardware* and *software* such as Python, Arduino, OpenCV, and Numpy. Arduino programming is employed to control a four-motor vehicle using a Bluetooth HC-05 module, while computer vision development tracks objects of different colors in real-time, sending commands to Arduino

¹Egresado de la Universidad La Salle Oaxaca, Escuela de Ingenierías y Arquitectura. ORCID: <https://orcid.org/0009-0009-2530-0652>. Correo electrónico: sibaja.enrique.am@gmail.com

²Universidad La Salle Oaxaca, Escuela de Ingenierías y Arquitectura.

³Universidad La Salle Oaxaca, Escuela de Ingenierías y Arquitectura.

⁴Universidad La Salle Oaxaca, Escuela de Ingenierías y Arquitectura.

⁵Maestro en robótica. Docente de la Universidad La Salle Oaxaca, Escuela de Ingenierías y Arquitectura y de la Universidad Tecnológica de los Valles Centrales de Oaxaca. ORCID: <https://orcid.org/0000-0001-5510-9394>. Correo electrónico: levid.rodriguez@gmail.com

Recibido: 24/11/2023 | Aceptado: 21/03/24 | Publicado: 08/06/2024

Los autores declaran no tener ningún conflicto de interés. La investigación fue financiada con recursos de los autores.

based on the detected color. The project is considered an advancement in mobile robotics by developing an integrated system that enables robots to navigate safely in dynamic environments. The combination of computer vision and motion control is presented as a practical solution supported by detailed testing.

KEYWORDS: Computer vision, Python, Arduino, OpenCV, Numpy

5. INTRODUCCIÓN

Según Ibarra (2009), la visión por computadora y el control de movimiento son pilares fundamentales en el campo de la robótica móvil y permiten que los robots interactúen de manera efectiva y segura en entornos dinámicos. En un entorno real, los objetos y agentes tienden a moverse de manera impredecible, lo que plantea un desafío significativo para la navegación de robots móviles. La interacción segura y eficiente con un entorno dinámico se convierte en un problema crucial que debe abordarse para lograr la autonomía deseada en la robótica móvil.

Este artículo se enfoca en la interacción de robots móviles con entornos dinámicos, destacando los desafíos que suponen estos ambientes en cuanto a variabilidad y cambio constante. Al respecto, se examinan las dificultades inherentes a la detección de objetos en tiempo real, entre ellas, la ocultación parcial, la variabilidad de iluminación y la presencia de objetos en movimiento rápido, los cuales complican significativamente el procesamiento visual y la toma de decisiones autónoma por el robot.

En un entorno en constante cambio, la detección y seguimiento de objetos son tareas críticas para evitar colisiones y garantizar una navegación segura. La capacidad de un robot para adaptarse de manera dinámica a situaciones imprevistas y reaccionar ante obstáculos en movimiento es esencial para su funcionamiento en aplicaciones del mundo real, como la logística, la atención médica y la exploración de entornos desconocidos (Arroyave et al., 2018).

A lo largo de las últimas décadas, se han realizado avances significativos en el campo de la visión por computadora y el control de robots móviles. Sin embargo, la interacción con entornos dinámicos sigue siendo un desafío técnico importante.

Los enfoques tradicionales basados en mapas estáticos y planificación de trayectorias predeterminadas a menudo resultan insuficientes para lidiar con entornos impredecibles. Al respecto, se han propuesto soluciones como la fusión de sensores, algoritmos de aprendizaje profundo y técnicas de control más avanzadas, pero todavía existe una brecha entre la investigación teórica y la implementación práctica en entornos reales (Ericksen y María, 2017).

La justificación que sustenta el abordaje de este problema de investigación radica en la necesidad de desarrollar sistemas de navegación de robots móviles más robustos y autónomos, a fin de que éstos puedan operar de manera segura en entornos cambiantes y dinámicos. Según Velázquez et al. (2015), la capacidad para detectar, rastrear y responder a objetos en movimiento en tiempo real no sólo mejora la seguridad del robot, sino que también amplía su gama de aplicaciones prácticas.

Este estudio se distingue de investigaciones anteriores en que integra los últimos avances en inteligencia artificial y aprendizaje automático para optimizar el procesamiento visual y la reactividad en tiempo real. Además, se exploran aplicaciones emergentes de la robótica en áreas críticas como la asistencia sanitaria y la gestión de desastres, abordando tanto las tendencias tecnológicas actuales

como los futuros desafíos en robótica móvil. Estos aspectos representan una mejora significativa en comparación con los enfoques tradicionales y se alinean con las necesidades y expectativas del campo.

Desde el punto de vista de la robótica aplicada, se trata de un problema de gran relevancia, con alto impacto en diversas áreas de la vida cotidiana y la industria. Por ejemplo, en el ámbito doméstico, robots móviles equipados con capacidades avanzadas de detección y navegación pueden realizar tareas de limpieza y mantenimiento de manera más eficiente, adaptándose a cambios en la disposición del hogar.

En la industria, estos sistemas mejoran la logística interna en almacenes dinámicos en que los robots deben navegar entre estanterías en constante cambio y evitar colisiones con empleados o vehículos de carga. Además, en el sector salud los robots móviles podrían desempeñar roles críticos en hospitales, como la entrega de medicamentos y el traslado de equipos, adaptándose a entornos altamente variables y ocupados. Estas aplicaciones concretas evidencian la relevancia y el potencial de las investigaciones en robótica móvil, para transformar significativamente nuestras interacciones con la tecnología en el día a día y en contextos industriales.

Por lo anterior, el objetivo principal de este trabajo es desarrollar un sistema integrado de modelado de visión por computadora y control de movimiento que permita a un robot móvil navegar de manera segura y eficiente en entornos dinámicos. Este sistema detecta objetos en movimiento y adapta trayectorias en tiempo real.

Para evaluar la efectividad de este sistema, se utilizarán criterios específicos, como la precisión y la rapidez de detección de objetos en movimiento, y la capacidad del robot para ajustar su ruta en respuesta a cambios imprevistos sin intervención humana. Además, se medirá el éxito de la adaptación de trayectorias mediante la reducción de incidentes de colisiones y la eficiencia en la ejecución de rutas en pruebas de campo.

Este enfoque integrado, que combina técnicas avanzadas de procesamiento de imágenes y control robótico, está diseñado para superar los desafíos que supone interactuar con un entorno en constante cambio y contribuir al avance de la robótica móvil en aplicaciones del mundo real.

MATERIALES Y MÉTODOS

2.1 Diseño de arquitectura del prototipo

Los elementos esenciales para este proyecto incluyen tanto *hardware* como *software* de código abierto. En la parte de *software*, se seleccionaron herramientas específicas para el desarrollo y la programación: Visual Studio Code, Python, el ide de Arduino y las librerías de OpenCV y Numpy. Se eligieron estas herramientas por el grado de aceptación con que cuentan en la comunidad de desarrollo, por su flexibilidad para la integración y su robustez para procesar imágenes y controlar dispositivos de *hardware*. Visual Studio Code proporciona un entorno versátil para escribir y depurar código, Python ofrece una extensa biblioteca de paquetes para el procesamiento de datos, Arduino facilita la interfaz con *hardware* y OpenCV; junto con Numpy, son fundamentales para las operaciones de visión por computadora.

- El procedimiento experimental se llevó a cabo en varias etapas claves:
- Configuración del entorno: instalación de los ide y bibliotecas necesarias en un entorno de desarrollo unificado.
- Programación y simulación: uso de Python y Arduino para desarrollar y simular el algoritmo de detección y el seguimiento de objetos.
- Integración *hardware-software*: conexión del código con los componentes físicos a través del ide de Arduino, asegurando la comunicación efectiva entre el *software* y el *hardware*.
- Pruebas de campo: implementación del sistema en un entorno controlado para monitorear la detección de objetos en movimiento y la adaptación de trayectorias. Con base en los resultados obtenidos, se realizaron ajustes iterativos.

Cada paso fue documentado meticulosamente, de manera de garantizar la reproducibilidad y validar la efectividad del sistema propuesto. La elección de estas técnicas y herramientas está directamente alineada con los objetivos de la investigación de mejorar la capacidad de navegación autónoma de los robots en entornos dinámicos.

En lo relativo al *hardware*, se utilizó una placa de desarrollo compacta y versátil que se basa en el microcontrolador ATmega328P de Atmel, con una frecuencia de reloj de 16 MHz. Ofrece 32 KB de memoria flash para almacenar programas, 2 KB de sram para datos en tiempo de ejecución y 1 KB de eeprom para almacenamiento no volátil. Integra 14 pines de entrada/salidas digitales, seis de ellos capaces de generar señales pwm y seis pines de entrada analógica, con capacidad de comunicación serial y usb y con un entorno de desarrollo basado en C/C++, así como módulos de acondicionamiento de señales que se describen a continuación.

El *protoboard* es un dispositivo esencial en el montaje de prototipos electrónicos, como se ilustra en la figura 1. Su forma es rectangular y está compuesto por una matriz de contactos interconectados dispuestos en filas y columnas. Cada intersección de esta matriz constituye un punto de conexión eléctrica, en el que se pueden insertar componentes electrónicos y cables. Las filas dentro del *protoboard* están conectadas horizontalmente, por lo general en grupos de cinco o 10, lo que facilita la conexión de componentes en paralelo. En contraste, las columnas están unidas verticalmente, posibilitando conexiones en serie entre diferentes elementos del circuito. Esta organización facilita el ensamblaje rápido y flexible de circuitos experimentales sin necesidad de soldadura, lo que hace

posible ajustar fácilmente las configuraciones durante la fase de pruebas y desarrollo.

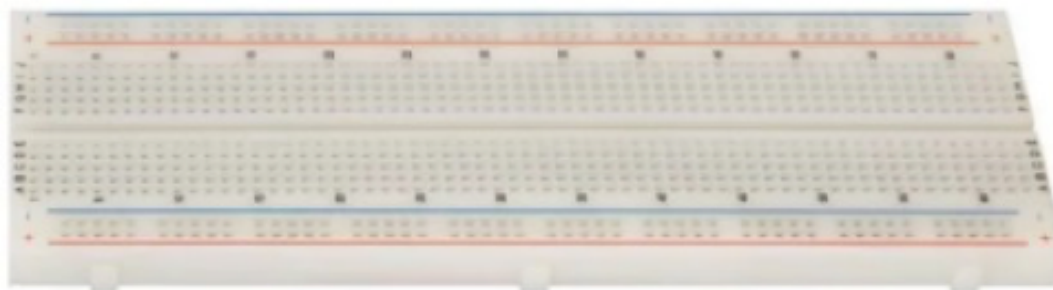


Figura 1. Protoboard de ensamble a presión, 1 bloque, 2 tiras y 830 perforaciones

El motorreductor, un componente mecánico crucial en numerosas aplicaciones de automatización y robótica, se compone de un motor eléctrico acoplado a una caja de engranajes. Este conjunto es del tipo doble eje tipo I y permite la transmisión de movimiento en dos direcciones diferentes. Como se muestra en la figura 2, el motorreductor opera con un voltaje de 3 a 6 Vcc, lo que lo hace adecuado para proyectos que requieren portabilidad y eficiencia energética. Su relación de transmisión es de 1:48, lo que optimiza la fuerza de salida del motor a expensas de la velocidad, con un límite operativo de 207 RPM a 6 Vcc. Esta configuración permite un equilibrio eficiente entre velocidad y torque, haciendo que este motorreductor sea ideal para aplicaciones en las que se requieren movimientos precisos y controlados.



Figura 2. Motor reductor de doble eje.

El módulo Bluetooth HC-05 es un componente esencial para la comunicación inalámbrica en proyectos de tecnología, especialmente en aquellos relacionados con la robótica y la automatización. Este módulo permite establecer conexiones Bluetooth y opera principalmente en modo esclavo, como se ilustra en la figura 3. Para la comunicación, utiliza una interfaz UART, que hace posible la transmisión de datos entre el módulo y el dispositivo que maneja la lógica de la visión por computadora. La funcionalidad UART facilita la integración con diversos microcontroladores y sistemas de procesamiento, lo que convierte al HC-05 en una opción versátil y eficaz para añadir capacidades de

comunicación inalámbrica a sistemas que requieren interacción remota o control a distancia, sin la necesidad de conexiones físicas directas.

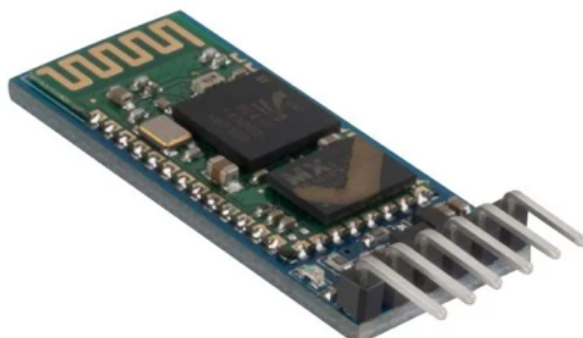


Figura 3. Módulo Bluetooth HC-05

2.2 Programación en Arduino

El código de Arduino se utilizó para controlar un vehículo con cuatro motores mediante un módulo Bluetooth HC-05. Dicho módulo permitió la comunicación inalámbrica con un dispositivo, como un teléfono móvil o una computadora, para enviar comandos que controlan el movimiento del vehículo. Al respecto, cabe detallar los aspectos señalados a continuación.

Se incluyó la biblioteca *SoftwareSerial.h* para habilitar la comunicación serial en pines digitales arbitrarios (en este caso, pines 10 y 11), junto con la definición de variables y pines necesarios.

Los pines digitales se configuraron y se estableció la comunicación con los motores y los pines de control. Cada motor tenía dos pines de control (In_1 e In_2, In_3 e In_4, In_5 e In_6, In_7 e In_8) destinados a controlar la dirección de giro; además, se establecieron pines de habilitación (EN_A, EN_B, EN_C, EN_D) que controlaban la velocidad de cada motor utilizando la función `analogWrite()`.

En la función `setup()` se inicializó la comunicación serial con el módulo HC-05 a una velocidad de 9600 bps y se configuraron los pines como entrada o salida según correspondiera.

En la función `loop()` se verificaba si había datos disponibles desde el módulo HC-05. Si se recibía un carácter válido, se ejecutaba un conjunto de instrucciones basado en el carácter recibido, que correspondía a un comando de control para el vehículo. Los comandos válidos son:

'1': Avanzar

'2': Girar a la derecha

'3': Retroceder

'4': Girar a la izquierda

'5': Detener todos los motores

Cada comando ejecutaba una función específica para controlar los motores y dirigir el movimiento del vehículo. Las funciones `adelante()`, `reversa()`, `giro_derecha()`, `giro_izquierda()` y `detener()` configuraban los pines de control de los motores para lograr el movimiento deseado.

El código realizaba el seguimiento de objetos de diferentes colores en tiempo real, utilizando para ello una cámara de video, y enviaba comandos a un dispositivo Arduino a través de una conexión serial, según el color del objeto detectado. A continuación, se explican a detalle partes importantes del código.

Se importaron las bibliotecas, siendo cv2 para OpenCV (para procesamiento de imágenes), numpy para operaciones numéricas, serial para comunicación con Arduino y time para introducir retrasos.

Se estableció la comunicación serial con el dispositivo Arduino conectado al puerto com7 y se introdujo un pequeño retraso para asegurar que la comunicación estuviera establecida.

La función dibujar (*mask*, color) detectaba y rastreaba objetos del color especificado (azul, amarillo o rojo) en la imagen capturada por la cámara, utilizaba la detección de contornos y calculaba el centro de masa del objeto para determinar su posición. Luego dibujaba un círculo y mostraba las coordenadas del objeto en la imagen.

Las máscaras de color se definieron en el espacio de color HSV para cada uno de los colores: azul, amarillo y rojo. Estas máscaras se utilizaron para aislar en la imagen los objetos de interés.

El bucle principal capturaba continuamente fotogramas de la cámara y aplicaba las máscaras de color para detectar objetos. Luego, llamaba a la función dibujar para procesar y rastrear objetos de cada color. Si se detectaba un objeto de color azul, se enviaba un comando "1" a Arduino; para el amarillo, se enviaba "2"; y para el rojo, se enviaba "3".

La ventana de video en tiempo real se mostraba en la pantalla y se verificaba si se presionaba la tecla "s" para finalizar el programa.

Finalmente, cuando se cerraba la ventana de video, se liberaba la cámara y se cerraban todas las ventanas abiertas. El código permitía realizar un seguimiento de objetos de diferentes colores utilizando una cámara y controlar un dispositivo Arduino según el color detectado. Los comandos enviados a Arduino podían utilizarse para realizar diversas acciones, por ejemplo, controlar un robot o activar actuadores en función de los objetos detectados.

2.4 Integración de códigos para desarrollo de software consolidado

A continuación, se muestra el diagrama de flujo inherente a la interacción de los programas:

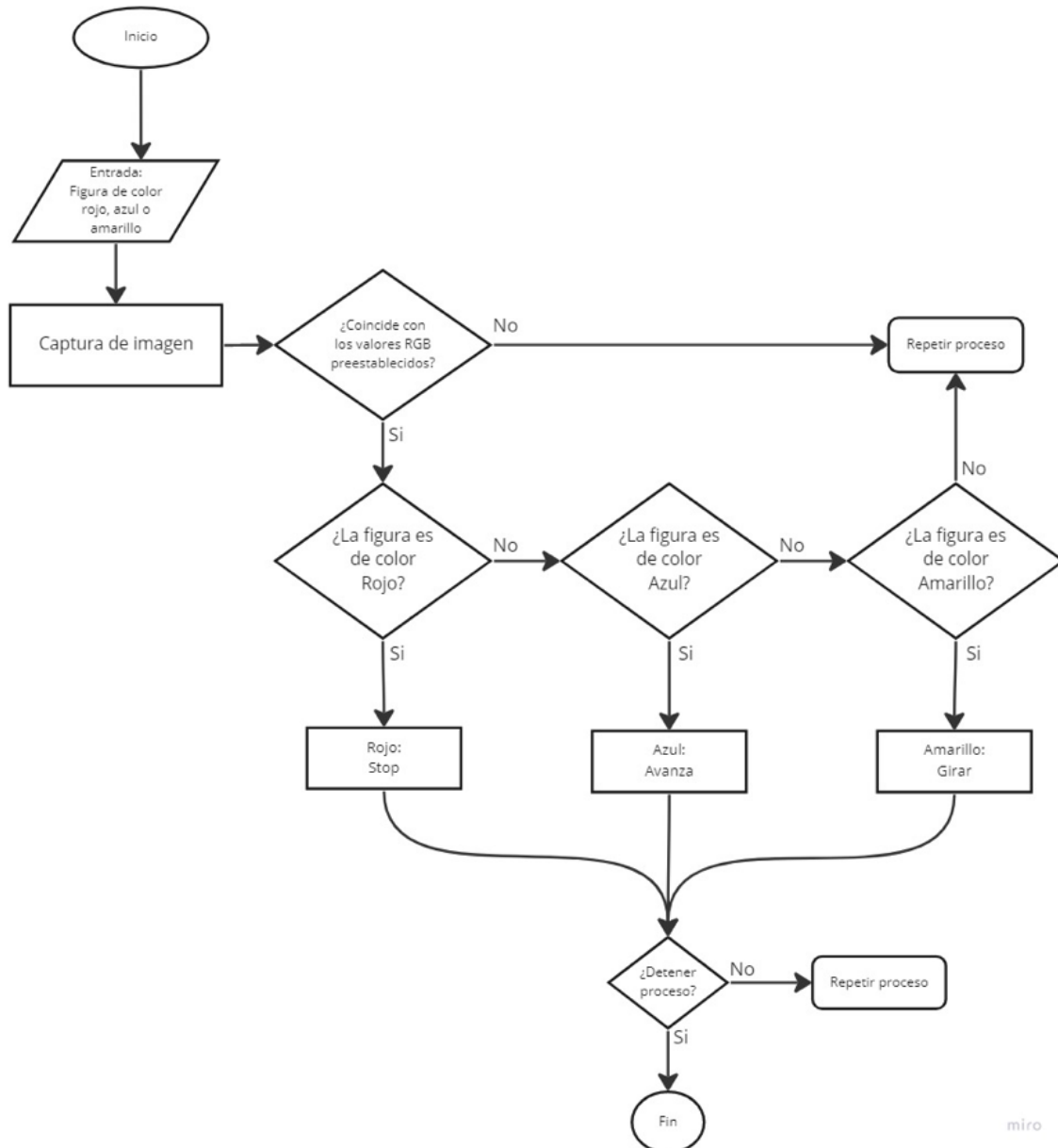


Figura 4. Diagrama de flujo de la interacción de los dos programas

2.5 Desarrollo del prototipo

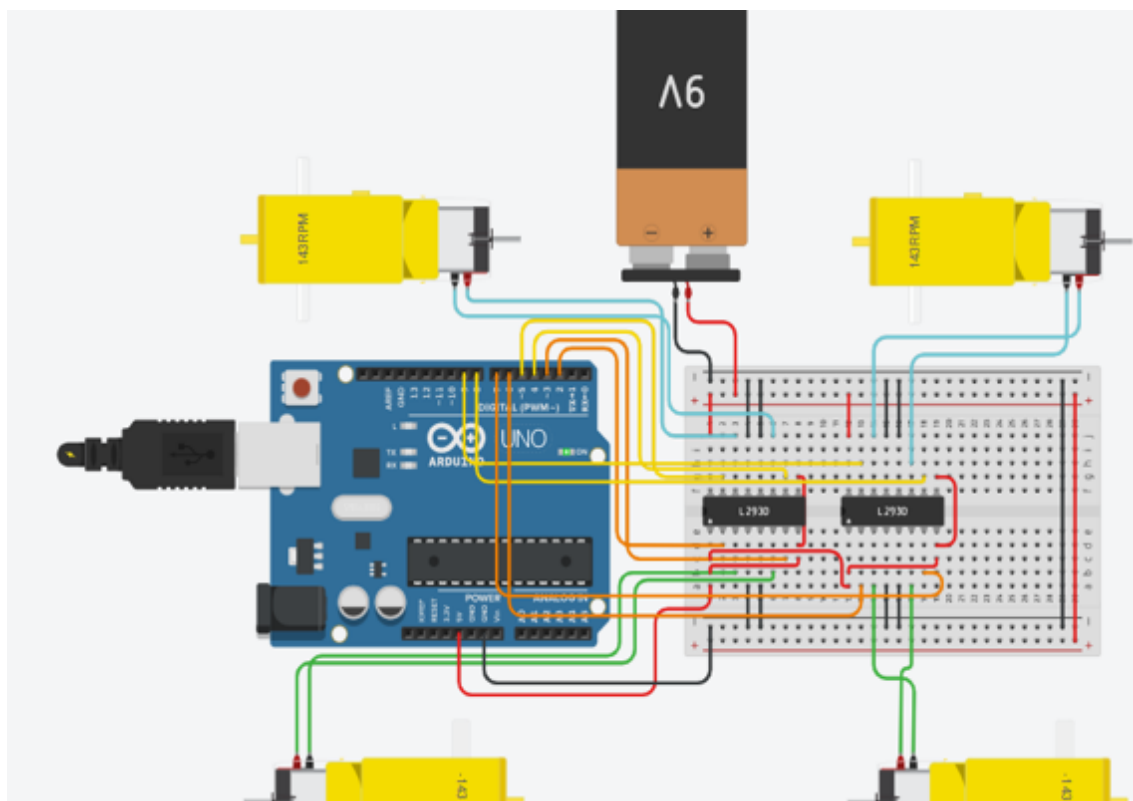


Figura 5. Simulación de la interacción del Hardware con la programación en Arduino.

En la figura 5 se presenta una simulación realizada en la plataforma TinkerCad; ésta ilustra la interacción entre los componentes de *hardware* y la programación en Arduino para el funcionamiento de un robot móvil.

En este escenario, el Arduino uno desempeña un papel crucial como emisor de señales de control hacia el microcontrolador L2930. Este último es responsable de gestionar y controlar la entrega de voltaje al motor. La señal emitida por el Arduino uno determina la dirección y el sentido del giro del motor. Para una implementación específica en el experimento, se configura el Arduino de manera que envíe comandos digitales al L2930, que modula la dirección del voltaje aplicado al motor según estos comandos.

Esto permite que el motor gire en sentido horario o antihorario, dependiendo de la instrucción recibida. Dicha configuración se prueba en un entorno controlado en el que se monitorean y ajustan las entradas del Arduino para optimizar la respuesta del motor y asegurar su funcionamiento correcto y eficiente en aplicaciones prácticas, como robots móviles o sistemas automatizados de transporte.

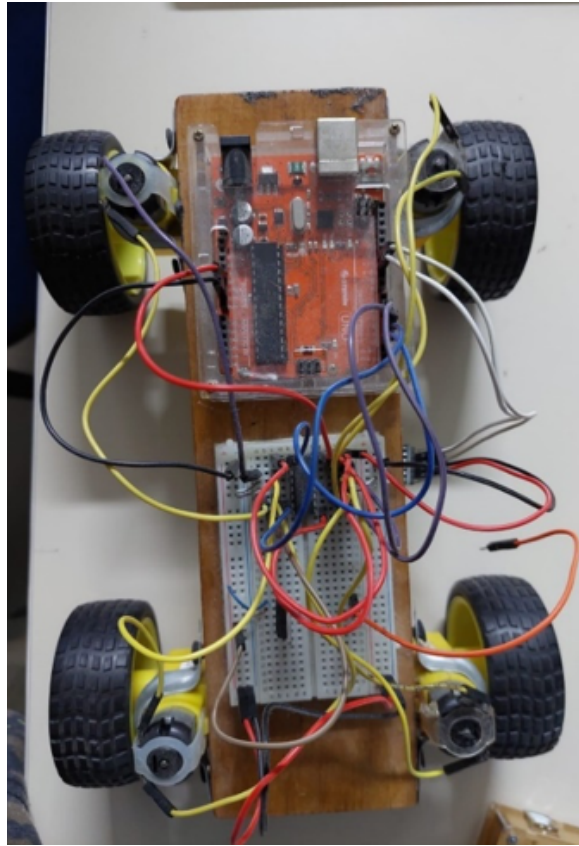


Figura 6. Prototipo armado

2.6 Realización de pruebas

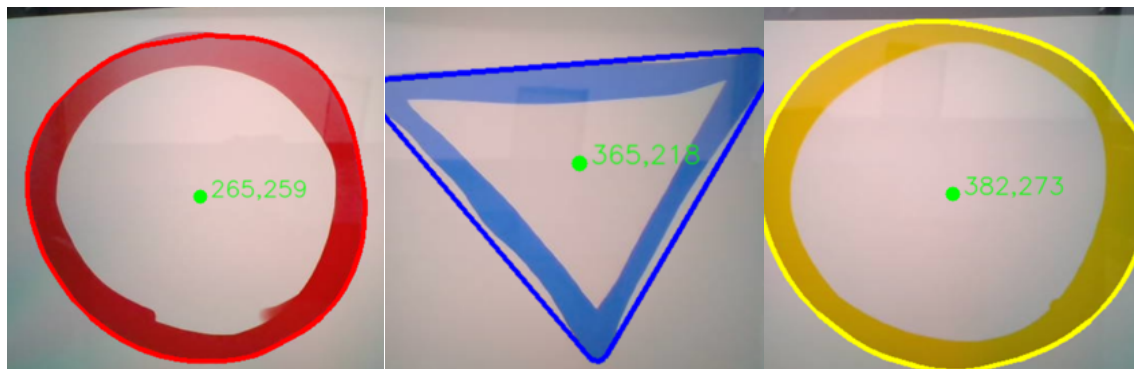


FIGURA 7. Detección de bordes de color rojo, enfatizando el centro de la figura

FIGURA 8. Detección de bordes de color azul, enfatizando el centro de la figura

FIGURA 9. Detección de bordes de color amarillo, enfatizando el centro de la figura

2.7 Desarrollo del prototipo final

La implementación del prototipo final para la navegación autónoma del robot móvil se centró en tres aspectos: el montaje mecánico, el control mediante la placa Arduino y la implementación de algoritmos de visión por computadora con Python.

En el ámbito mecánico se realizó un montaje de los motorreductores en el chasis, buscando el equilibrio entre tracción y potencia suficiente para el movimiento.

La configuración de la placa Arduino se enfocó en una programación para un control de los motorreductores y una comunicación con el módulo Bluetooth que permitiera el funcionamiento de la visión por computadora.

La implementación de algoritmos de visión por computadora se llevó a cabo con Python, para posibilitar el reconocimiento de obstáculos y la definición de rutas para la navegación autónoma en caso de detectar algún objeto que le impidiera seguir con el movimiento actual.

La sincronización entre los códigos de Arduino para el control de movimiento del prototipo y Python, así como para el análisis de visión por computadora y la elección de movimientos dirigidos al robot, se optimizó a fin de conseguir una operación fluida.

La calibración y los ajustes optimizaron los parámetros en aras de lograr una respuesta óptima de los motorreductores con base en la retroalimentación visual. Las pruebas validaron la estabilidad y eficiencia del prototipo, demostrando su capacidad para la navegación autónoma, lo que posibilitará incorporarle una carrocería que mejore su aspecto visual.

RESULTADOS

Durante las pruebas de detección de objetos en tiempo real utilizando la visión por computadora y el sistema de control implementado se obtuvieron los siguientes resultados.

3.1 Precisión de detección:

La detección de objetos de diferentes colores (azul, amarillo y rojo) mostró una precisión promedio de 90%, siendo verificada por medio de múltiples pruebas de seguimiento en entornos variados. Es importante mencionar que, en ocasiones, la precisión de 90% se vio afectada por factores externos como el contraste o la iluminación ambiental. Se recomienda evitar elementos externos que puedan influir en la tonalidad del color detectado, como la luz solar o la oscuridad.

3.2 Integración de visión por computadora y control de movimiento

Control de movimiento:

El control de movimiento fue altamente efectivo, ya que los comandos enviados al Arduino permitieron un control eficiente y preciso del vehículo; éste fue controlado y cumplió con las instrucciones de avance, retroceso y giro, tanto a la derecha como a la izquierda, de manera satisfactoria.

3.3 Pruebas de funcionalidad del prototipo

Maniobrabilidad del robot:

En los casos de prueba el robot móvil logró navegar y evitar obstáculos de manera efectiva en entornos dinámicos.

Para visualizar el funcionamiento del prototipo desarrollado, se puede acceder al siguiente enlace:
Video

CONCLUSIONES

El desarrollo e implementación del sistema de visión por computadora y control de movimiento para la navegación de un robot móvil produjo resultados significativos y perspectivas prometedoras en el campo de la robótica aplicada. A partir de los resultados y hallazgos obtenidos, se puede mencionar que la combinación de la visión por computadora y el control de movimiento demostró ser efectiva para lograr una navegación segura y adaptable.

Se debe profundizar en el perfeccionamiento de la detección de objetos, considerando posibles mejoras en el algoritmo para minimizar la influencia de factores externos. Asimismo, es necesario explorar estrategias que permitan aumentar la adaptabilidad del robot a entornos más desafiantes.

En conjunto, los resultados obtenidos respaldan la viabilidad y efectividad de la combinación de visión por computadora y control de movimiento en la navegación autónoma de robots móviles. Estos logros representan un paso enorme hacia el desarrollo de sistemas robóticos más avanzados y adaptativos para diversas aplicaciones en el mundo real.

REFERENCIAS

- Arroyave, L. F., Carvajal, M., y Ceballos, N. D. M. (2018b). Navegación de robots móviles en entornos con discontinuidades: una revisión. *Revista Politécnica*, 14(27), 103-115. <https://doi.org/10.33571/rpolitec.v14n27a10>
- Velásquez Hernández, C. A., Chávez Chávez, J. J., Córdoba Nieto, E. (2015). Implementación de sistema de navegación autónomo en robot móvil experimental para reconstrucción y exploración de entornos desconocidos. *Revista eia. Memorias*. <https://revistas.eia.edu.co/index.php/mem/article/download/817/734/2346>
- Cazorla, M., y Viejo, D. (2015). Experiences Using an Open Source *Software* Library to Teach Computer Vision Subjects. *Journal of Technology and Science Education*, 5(3), 214-227.
- Ericksen, P., y María, A. (2017). Implementación y comparativa de algoritmos de control y planificación local para robots móviles utilizando ros. Archivo Digital UPM. <https://orcid.org/0000-0002-8652-0300>
- Espinosa Valcárcel, F. A., Gordillo Cháves, C. A., Jiménez Moreno, R., y Avilés Sánchez, O. F. (2013). Machine vision algorithms applied to dynamic traffic light control. *Dyna*, 80(178), 132-140.
- Ibarra Bonilla, M. N. (2009). *Navegación autónoma de un robot con técnicas de localización y ruteo*. Tesis de maestría. inaoe repositorio. <http://inaoe.repositorioinstitucional.mx/jspui/handle/1009/394>
- Montoya Holguín, C., Cortés Osorio, J. A., y Chaves Osorio, J. A. (2014). Sistema automático de reconocimiento de frutas basado en visión por computador. *Ingeniare. Revista Chilena de Ingeniería*, 22(4), 504-516.
- Muñoz, D. J. (2006). Proceso de reconocimiento de objetos, asistido por computador (visión artificial), aplicando gases neuronales y técnicas de minería de datos. *Scientia Et Technica*, XII(30), 385-390.
- Rehman, U. et al. (2023). Human tracking robotic camera based on image processing for live streaming of conferences and seminars. *Heliyon*, 9(8).
- Rosenfeld, A., y Chellappa, R. (2003). Image Processing. *Encyclopedia of Physical Science and Technology* (Third Edition), 595-630.
- Saldaña, E., Siche, R., Huamán, R., Luján, M., Castro, W., y Quevedo, R. (2013). Computer vision system in real-time for color determination on flat surface food. *Scientia Agropecuaria*, 4(1), 55-63.
- Tarлак, F., Ozdemir, M., y Melikoglu, M. (2016). Computer vision system approach in colour measurements of foods: Part I. development of methodology. *Ciencia y Tecnología de Alimentos*, 36(2), 382-388.
- Terashima, N. (2002). 12 – Computer Vision. Intelligent Communication Systems.
- Urbano López, J., Bacca Cortes, E. B., y Buitrago Molina, J. (2021). Vision-based *Software* Tool System for Position Estimation Using a Smartphone. *Ingeniería*, 26(2), 284-305.
- Valencia, C. et al. (2020). Driver-Assistant System Using Computer Vision and Machine Learning. *Revista Facultad de Ingeniería*, 29(54).

Derechos de Autor© 2024 Sibaja Galindo, Enrique, Bacilio López, Florencia, Carrasco Francisco, José Alberto, Quintana Ramírez, Roberto, Rodríguez Santiago Armando Levid



Este texto está protegido por una licencia [Creative Commons 4.0](https://creativecommons.org/licenses/by/4.0/). Usted es libre para Compartir, copiar y redistribuir el material en cualquier medio o formato y adaptar el documento, remezclar, transformar y crear a partir del material— para cualquier propósito, incluso para fines comerciales, siempre que cumpla la condición de: Atribución: Usted debe dar crédito a la obra original de manera adecuada, proporcionar un enlace a la licencia, e indicar si se han realizado cambios. Puede hacerlo en cualquier forma razonable, pero no de forma tal que sugiera que tiene el apoyo del licenciante o lo recibe por el uso que hace de la obra.