Ciencia e Ingeniería Neogranadina
2023
Vol. 33(2)
julio-diciembre de 2023 ■ ISSN: 0124-8170 · ISSN-e: 1909-7735 ■ pp. 79-97
Editorial Neogranadina

DOI: https://doi.org/10.18359/rcin6919

Coordinación mano-ojo de un brazo robótico utilizando una cámara estéreo*

Alay Camilo López Castaño[a] ■ Innis Dapney Salazar García[b] ■ Rodolfo García Sierra[c] ■ Germán Darío Zapata Madrigal[d]

Resumen: Este trabajo presenta los resultados obtenidos luego de implementar un algoritmo de visión artificial desarrollado en Python para estimar la posición de un objeto por medio de información visual de una cámara estéreo. La estimación de la posición del objeto es utilizada por el controlador de un brazo robótico, para posicionarlo y sujetar el objeto; sin embargo, el brazo robótico no siempre llega de forma precisa al punto esperado, en consecuencia, se complementó este proceso con un algoritmo de corrección de la posición basado en el algoritmo de optimización Gradient Descent y el proceso de coordinación mano-ojo que hacen los seres humanos. Los valores de posición son enviados, mediante wifi, a través del protocolo TCP/IP y por medio de sockets, al controlador del brazo robótico. Los resultados experimentales obtenidos muestran que, a mayor resolución de la imagen de la cámara, mejor es la estimación de la posición del objeto y, en general, con el algoritmo de corrección implementado, la distancia entre la posición final del robot y la posición del objeto no supera los 10 mm.

Palabras clave: brazo robótico; calibración estéreo; control servo-visual; gradiente descendiente; segmentación por color; triangulación 3D

Recibido: 05/09/2023 **Aceptado:** 17/10/2023 **Disponible en línea:** 27/12/2023

Cómo citar: A. C. López Castaño, I. D. Salazar García, R. García Sierra, y G. D. Zapata Madrigal, «Coordinación mano-ojo de un brazo robótico utilizando una cámara estéreo», Cien.Ing. Neogranadina, vol. 33, n.º 2, pp. 79–97.Diciembre 2023.

* Artículo de investigación

a Ingeniero electricista. Universidad Nacional de Colombia, Medellín, Colombia.
Correo electrónico: alclopezca@unal.edu.co ORCID: https://orcid.org/0009-0007-4283-5531

b Ingeniera de control. Universidad Nacional de Colombia, Medellín, Colombia.
Correo electrónico: isalazar@unal.edu.co ORCID: https://orcid.org/0009-0002-7652-6538

c Ph. D. en Innovation & Strategy Future Thinking, Master of Science, ingeniero electricista. Enel-Codensa S.A. E.S.P. (Grupo Enel), proveedor de servicios de energía, Bogotá, Colombia.
Correo electrónico: rodolfo.garcia@enel.com ORCID: https://orcid.org/0000-0002-3892-6189

d Ph. D. en ciencias aplicadas, magíster en automática, especialista en alta gerencia con énfasis en calidad, ingeniero eléctrico. Universidad Nacional de Colombia sede Medellín, Colombia.
Correo electrónico: gdzapata@unal.edu.co ORCID: https://orcid.org/0000-0002-7739-1578

Hand-Eye Coordination of a Robotic Arm Using a Stereo Camera

Abstract: This work presents the results obtained after implementing a computer vision algorithm developed in Python to estimate the position of an object through visual information from a stereo camera. The object's position estimation is utilized by the controller of a robotic arm to position it for grasping the object. However, the robotic arm does not always reach the expected point accurately. Consequently, this process was complemented with a position correction algorithm based on the Gradient Descent optimization algorithm and the hand-eye coordination process performed by humans. The position values are sent via WIFI using the TCP/IP protocol through sockets to the robotic arm controller. Experimental results demonstrate that, with higher camera image resolution, the object's position estimation improves. Overall, with the implemented correction algorithm, the distance between the robot's final position and the object's position does not exceed 10 mm.

Keywords: Robotic Arm; Stereo Calibration; Visual-Servo Control; Gradient Descent; Color Segmentation; 3D Triangulation

Introducción

Los brazos robóticos son dispositivos mecánicos diseñados para llevar a cabo diversas tareas, imitando en cierta medida la funcionalidad de un brazo humano. Estos dispositivos han adquirido una enorme importancia en diversos campos de aplicación, como en las líneas de producción industrial, la medicina, la investigación submarina y la agricultura, debido a sus múltiples ventajas y aplicaciones, lo que los ha convertido en una parte integral de la revolución industrial en curso (industria 4.0). Sin embargo, la mayoría de estos actualmente solo puede trabajar de manera fiable en entornos estructurados, es decir, con una repetición predefinida de tareas, lo que lleva a que, si el entorno de trabajo o el objeto a manipular presentan algún cambio, el robot debe reprogramarse debido a que convencionalmente no está equipado con herramientas que le permitan tomar decisiones con base en su entorno, como lo hace el ser humano al agarrar un objeto. Si se dota a un brazo robótico con un sistema de visión artificial, se le proporciona la capacidad de "ver" y procesar información visual de su entorno. Así, la combinación de brazo robótico con visión artificial abre un amplio abanico de posibilidades y ventajas en diversas aplicaciones, otorgando adaptabilidad y flexibilidad que permiten realizar tareas en entornos no estructurados. Con todo, aunque la manipulación robótica basada en visión ha tenido buenos resultados en diferentes contextos, la precisión sigue siendo un problema [1], [2]. Por lo anterior, este trabajo contribuye en la minimización del error entre la posición del efector final de un brazo robótico y el objeto a sujetar, basándose en el algoritmo Gradient Descent y el comportamiento de coordinación ojo-mano que realiza el ser humano al sujetar un objeto utilizando visión estereoscópica.

Materiales y métodos

Método

En este trabajo se analiza la integración de un brazo robótico y una cámara estéreo. Con la cámara estéreo y algoritmos de visión artificial se lleva a cabo la detección y ubicación espacial 3D de un objeto, lo cual se utiliza para mover el brazo a la posición del objeto y sujetarlo, permitiendo al robot desempeñarse en entornos no estructurados, es decir, cuando el objeto a manipular presenta algún cambio en su posición. La detección del objeto se realiza utilizando segmentación por color, y para la estimación de su posición se utiliza triangulación 3D, basada en la geometría proyectiva de la cámara estéreo y el modelo de una cámara estenopeica.

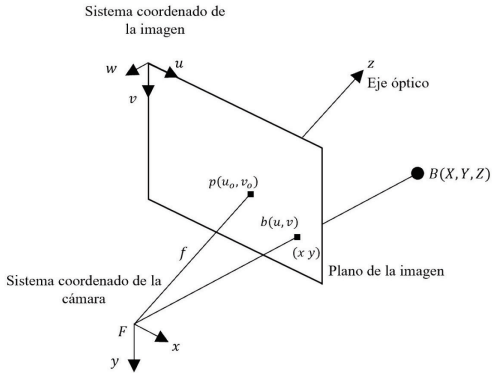
La cámara estéreo brinda la posibilidad de realizar una estimación del punto 3D del objeto a manipular, a partir de sus datos de posición 2D en las imágenes capturadas con la cámara, para lo cual es necesario ejecutar un proceso llamado *calibración estéreo*, en el cual se estiman parámetros del modelo matemático de la cámara. Los datos de posición se envían a través de una comunicación *socket*, por medio de una interfaz inalámbrica wifi, al controlador del robot, para que este lo posicione en la ubicación determinada.

La estimación de la posición del objeto por medio de la cámara es imprecisa, por lo que en este estudio se propuso un algoritmo para la corrección de la posición del robot, basado en el algoritmo de optimización Gradient Descent y la coordinación mano-ojo que realizan los seres humanos para alcanzar y sujetar objetos.

Modelo de la cámara

Las cámaras son representadas con el modelo estenopeico, el cual se compone de un plano de formación de imagen y un centro óptico ubicado en el origen de coordenadas de la figura 1. En este modelo, un punto 3D del espacio B es proyectado en un punto b en el plano de la imagen, a través de una transformación lineal. El punto p es la intercepción del eje óptico Z con el plano de la imagen (generalmente ubicado en el centro del plano de la imagen, conocido como el punto principal), mientras que f es la distancia entre el plano de la imagen y el centro óptico, conocida como distancia focal.

Figura 1. Geometría proyectiva de la cámara estéreo



Fuente: elaboración propia.

Utilizando semejanza de triángulos entre los puntos B y (x, y) de la figura 1, es posible obtener las relaciones mostradas en (1):

$$x = f\frac{X}{Z};\ y = f\frac{Y}{Z} \tag{1}$$

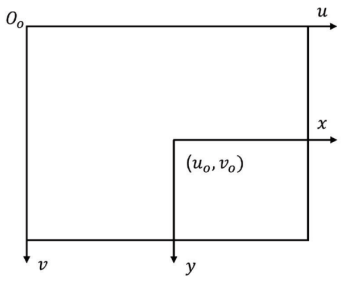
Aquí, (x, y) son las coordenadas de proyección del punto (X, Y, Z) en el plano de la imagen. Es posible expresar estas ecuaciones en forma matricial como (2):

$$Z\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \tag{2}$$

Esto permite reemplazar Z por un factor λ sin perder generalidad.

El sistema de coordenadas de una imagen digital (u, v) se mide en pixeles, de modo que el origen de coordenadas se encuentra en la esquina superior derecha de la figura 2, mientras que corresponde a las columnas y v a las filas.

Figura 2. Relación entre las coordenadas de la imagen física y la imagen digital; O_0 corresponde al origen de coordenadas de u y v



Fuente: elaboración propia.

Debido a que la representación de coordenadas (x, y) de la imagen antes de ser digitalizada es medida en pixeles, se hace necesario tener una relación entre las coordenadas (u, v) de la imagen digital y las coordenadas (x, y) de la imagen física (figura 2), medidas en unidades de longitud. Si definimos dx y dy como las unidades de $[\frac{\boldsymbol{longitud}}{\boldsymbol{píxel}}]$ en dirección u y v respetivamente, es posible relacionar las coordenadas de la imagen física y la imagen digital como (3):

$$u = \frac{x}{dx} + u_0\ ;\ v = \frac{y}{dx} + v_0 \tag{3}$$

Estas relaciones se pueden expresar en forma matricial homogénea (4):

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{1}{dx} & 0 & u_0 \\ 0 & \frac{1}{dy} & v_0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{4}$$

Si no se toma el origen de coordenadas del punto (X, Y, Z) como el sistema de coordenadas de la cámara, sino cualquier otro punto origen arbitrario, se debe realizar una transformación del punto (X, Y, Z) del origen de coordenadas arbitrario al origen de coordenadas de la cámara (en coordenadas homogéneas), como en (5):

$$\begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}_{cámara} = \begin{bmatrix} [R]_{3\times3} & [t]_{3\times1} \\ 0_{1\times3} & 1 \end{bmatrix}\begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{5}$$

Donde $[R]_{3\times3}$ y $[t]_{3\times1}$ corresponden a una rotación y a una traslación en tres dimensiones, respectivamente.

Con las ecuaciones (2), (3) y (4) se obtiene (6):

$$\lambda\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{1}{dx} & 0 & u_0 \\ 0 & \frac{1}{dy} & v_0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} [R]_{3\times3} & [t]_{3\times1} \\ 0_{1\times3} & 1 \end{bmatrix}\begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{6}$$

Multiplicando las dos primeras matrices de (6) se obtiene (7):

$$\lambda\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{f}{dx} & 0 & u_0 \\ 0 & \frac{f}{dy} & v_0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} [R]_{3\times3} & [t]_{3\times1} \\ 0_{1\times3} & 1 \end{bmatrix}\begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{7}$$

Si $\boldsymbol{f_x} = \frac{\boldsymbol{f}}{\boldsymbol{dx}}$ y $\boldsymbol{f_y} = \frac{\boldsymbol{f}}{\boldsymbol{dy}}$ se obtiene (8):

$$\lambda\begin{bmatrix}u\\v\\1\end{bmatrix} = \begin{bmatrix}f_x & 0 & u_0\\0 & f_y & v_0\\0 & 0 & 1\end{bmatrix}\begin{bmatrix}[R]_{3\times3} & [t]_{3\times1}\\0_{1\times3} & 1\end{bmatrix}\begin{bmatrix}X\\Y\\Z\\1\end{bmatrix};$$

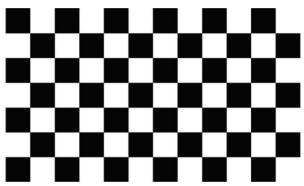
$$\lambda\begin{bmatrix}u\\v\\1\end{bmatrix} = K * P[R|t]\begin{bmatrix}X\\Y\\Z\\1\end{bmatrix} \quad (8)$$

Aquí, K se conoce como matriz intrínseca de la cámara y solo depende de las propiedades específicas de cada cámara. El proceso por el cual se obtienen los parámetros (f_x, f_y, u_0, v_0), también llamados parámetros intrínsecos, se denomina calibración. Por su parte, P se conoce como la matriz extrínseca y no depende las propiedades de la cámara [3], [4].

Calibración de la cámara estéreo

La calibración de la cámara y la estimación espacial de objetos por métodos de identificación, detección y clasificación de objetos en imágenes, son procesos fundamentales en algoritmos de visión artificial y en sistemas de percepción visual. La calibración permite determinar los parámetros intrínsecos y extrínsecos de una cámara, lo cual resulta necesario para lograr mediciones y reconstrucciones a partir de las imágenes capturadas. En este trabajo, dicha actividad de calibración de la cámara se realizó con ayuda de la herramienta Stereo Camara Calibrator de Matlab R2022b, haciendo uso de un patrón de tablero de ajedrez con un número impar de columnas, como el que se puede ver en la figura 3. Esta herramienta de calibración se basa en los conocimientos y técnicas descritos en los artículos [5] y [6].

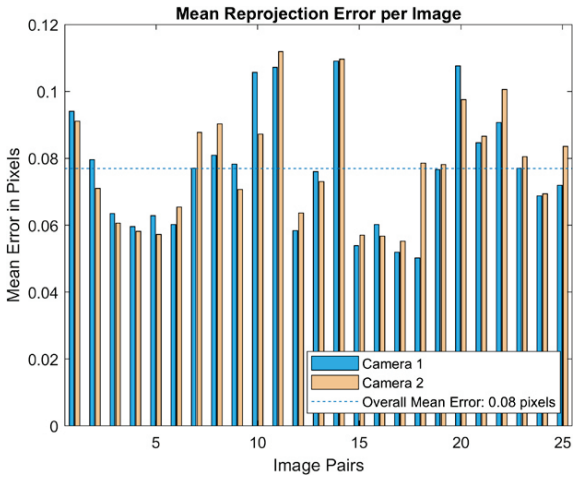
Figura 3. Tablero de ajedrez utilizado para la calibración



Fuente: elaboración propia.

El proceso de calibración fue aplicado a 54 fotografías, 26 tomadas por la cámara izquierda y 26 por la cámara derecha, en tres resoluciones distintas: 320x240, 640x480 y 1280x720 pixeles. Para mejor precisión en la calibración, se eliminaron parejas de imágenes que inducían errores altos y se siguieron las recomendaciones de la documentación del Matlab. La figura 4 corresponde al error medio de reproyección por imagen, para una resolución de 320x240 pixeles, pudiéndose apreciar que el error medio de las parejas de imágenes es menor a 0.08 pixeles y que el error máximo se encuentra por debajo de 0.12 pixeles.

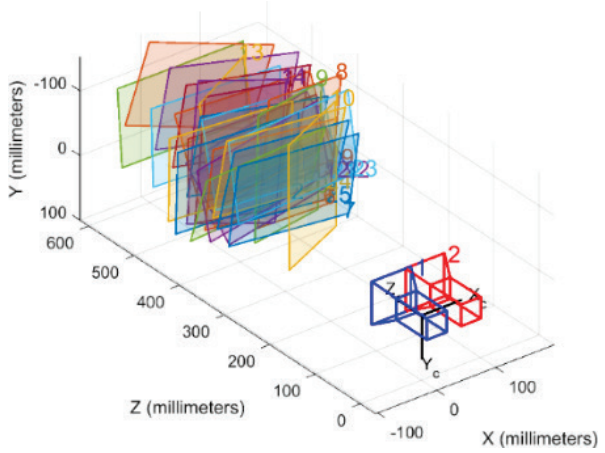
Figura 4. Error medio de proyección por imagen en resolución 320x240 pixeles



Fuente: elaboración propia.

En el proceso de calibración de estéreo de Matlab, la matriz de los parámetros extrínsecos solo es estimada para la cámara derecha. Esta matriz transforma los puntos 3D del origen de coordenadas de la cámara derecha al origen de coordenadas de la cámara izquierda, que viene a determinar el origen de coordenadas de todo el sistema estéreo. En la figura 5 se puede visualizar una representación gráfica de la estimación de los parámetros extrínsecos: los resultados muestran que la cámara derecha se encuentra en dirección positiva en el eje de las x respecto al origen de coordenadas de la cámara izquierda y que su eje óptico es aproximadamente paralelo al eje óptico de esta última.

Figura 5. Visualización de parámetros extrínsecos, resolución 320x240 pixeles



Fuente: elaboración propia.

A continuación, en la tabla 1 se resumen los resultados de la estimación de los parámetros intrínsecos obtenidos en el proceso de calibración para diferentes resoluciones de la cámara.

Tabla 1. Resultados de la estimación de los parámetros intrínsecos

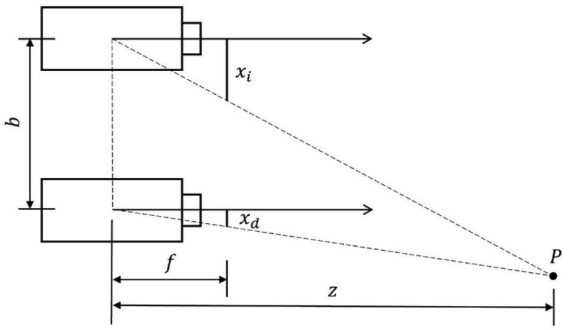
Resolución (pixeles)	f_x (pixeles)	f_y (pixeles)	u_0 (pixeles)	v_0 (pixeles)	Error de proyección medio (pixeles)	Error de proyección máximo (pixeles)
Cámara izquierda						
320x240	291.5448	291.6590	173.7110	131.7255	0.08	0.11
640x480	584.0250	584.0595	346.0125	263.0601	0.13	0.19
1280x720	1165.0841	1163.5909	704.4501	520.8506	0.19	0.25
Cámara derecha						
320x240	291.3675	291.6206	171.8543	134.1549	0.08	0.11
640x480	584.0593	584.5008	343.0687	268.9346	0.13	0.19
1280x720	1164.2101	1163.9696	696.3438	530.8029	0.19	0.25

Fuente: elaboración propia.

Estimación de la posición

El cálculo de las coordenadas espaciales de la pieza se hace a través de la geometría estéreo (figura 6). Para realizar este proceso es necesario contar con dos cámaras, de tal manera que se capte una imagen izquierda y una derecha del mismo objeto.

Figura 6. Geometría estéreo



Fuente: elaboración propia.

En este trabajo se realizaron dos suposiciones: un montaje con cámaras idénticas, de ejes ópticos paralelos. Esto es válido, y se confirma con los resultados obtenidos en la calibración estéreo de la cámara utilizada en este trabajo. La semejanza entre los parámetros intrínsecos corrobora la primera suposición, y la segunda suposición se confirmó analizando la matriz extrínseca obtenida en el mismo proceso de calibración, en la cual se observó que los parámetros de rotación son casi nulos y que la posición relativa del centro óptico de la cámara derecha está esencialmente a 58 mm en dirección positiva sobre el eje x respecto del origen de coordenadas de la cámara izquierda.

La figura 6 muestra una vista superior de la disposición de las cámaras: los centros ópticos están separados por una distancia llamada línea de base (b) en unidades de longitud, x_i y x_d hacen referencia en pixeles de las coordenadas en x, medidas desde el punto principal, de la proyección del punto 3D P sobre la imagen izquierda y la derecha, respectivamente, mientras que f es la distancia focal en pixeles. Aplicando semejanza de triángulos en la figura 6 se obtiene (9):

$$\frac{x_i}{X} = \frac{f}{Z} \; ; \; \frac{x_d}{X - b} = \frac{f}{Z} \tag{9}$$

Donde X es la coordenada en x del punto $P(X, Y, Z)$, desde el origen de coordenadas de la cámara izquierda. De las relaciones de (9) se obtiene (10):

$$Z = \frac{b \cdot f}{x_i - x_d} = \frac{b \cdot f}{d} \tag{10}$$

Aquí, la variable $d = (x_i - x_d)$ es conocida como disparidad. Del mismo modo se obtienen el resto de las coordenadas del punto P (11):

$$X = \frac{Z \cdot x_i}{f} \; ; \; Y = \frac{Z \cdot y_i}{f} \tag{11}$$

Las ecuaciones (10) y (11) permiten estimar las coordenadas de un punto 3D a partir de dos imágenes, identificando la proyección en pixeles del punto, tanto en la imagen izquierda como en la derecha. En este trabajo, el parámetro f se tomó como la media entre los f_x y f_y de ambas cámaras (izquierda y derecha) obtenidos en el proceso de calibración, y el parámetro b representa la coordenada en x del vector de translación de la matriz extrínseca del mismo proceso [3], [4].

En este estudio se utilizó la segmentación por color y forma, y el método de los momentos, para estimar la ubicación del centro del objeto considerado en las imágenes capturadas, dando como resultado las coordenadas en pixeles del objeto desde el origen en la esquina superior izquierda de las mismas imágenes capturadas, lo que hace necesario realizar las transformaciones de coordenadas mostradas en (12):

$$x_i = u_i + u_{0i} \; ; \; y_i = v_i + v_{0i} \; ; \; x_d = u_d + u_{0d} \tag{12}$$

Aquí, u_i es la coordenada en u del objeto en la imagen izquierda, teniendo como origen la esquina superior izquierda de la imagen; u_{0i} es la coordenada en u del punto principal en la imagen izquierda; v_i es la coordenada en v del objeto en la imagen izquierda, teniendo como origen la esquina superior izquierda de la imagen; v_{0i} es la coordenada en v del punto principal en la imagen izquierda; u_d es la coordenada en u del objeto en la imagen derecha, teniendo como origen la esquina superior izquierda de la imagen; y u_{0d} es la coordenada en del punto principal en la imagen derecha.

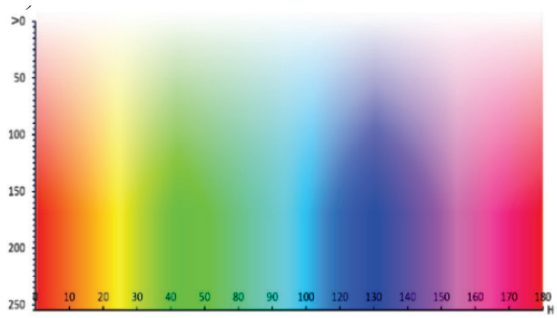
Segmentación basada en color

El espacio de color HSV (matiz, saturación, valor) es un modelo de representación del espacio de color similar al modelo RGB, el cual resulta muy útil en el procesamiento de señales para segmentar objetos según su color. El matiz representa la tonalidad del

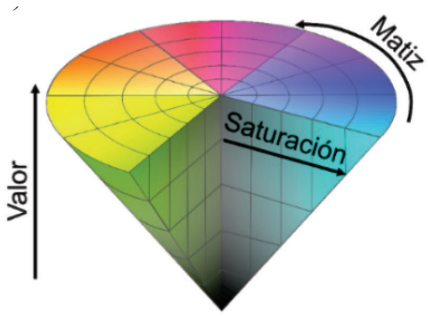
color (eje horizontal), en un rango de 0° – 179°; por ejemplo, podríamos decir que el tono de verde está contenido aproximadamente entre 40° – 80° (figura 7-a).

Figura 7. a) Espacio de color HSV en dos dimensiones: eje vertical "valor", eje horizontal "matiz"; b) espacio de color HSV en tres dimensiones

a)



b)



Fuente: elaboración propia.

La saturación varía entre 0 – 255, al igual que en el espacio RGB, indicando qué tan "fuerte" es el color: cuanto menor sea la saturación de un color, mayor tonalidad grisácea tendrá y más decolorado estará (figura 7-b). Por su parte, el valor, que varía entre 0 – 255 niveles, al igual que la saturación, representa el brillo o la intensidad del color, de modo que, cuanto menor sea, más oscuro o negro se tornará el color.

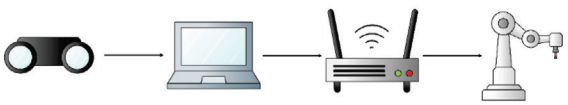
La pieza que se utilizó para el desarrollo de este trabajo corresponde a una pieza circular de color verde, así que el rango de valores de segmentación de color va desde las coordenadas HSV [50, 200, 40] hasta HSV [90, 255, 255].

Adicionalmente, en este trabajo se implementó un algoritmo de corrección de la posición del brazo robótico utilizando la posición de la herramienta del robot, la cual corresponde a una ventosa de color rojo, por lo que el rango de valores en el espacio de color HSV utilizado para la segmentación de este objeto fue desde [0, 100, 80] hasta [15, 255, 255] y desde [170, 100, 80] hasta [179, 255, 255], para cubrir todo el rango de colores del rojo (ver figura 12) [7], [8].

Arquitectura

En esta sección se describen los diferentes componentes que participan en la comunicación de los elementos del sistema físico del montaje realizado para la integración del brazo robótico y la visión artificial (figura 8), del mismo modo que se explicarán los procesos y los protocolos bajo los que se rigen dichas comunicaciones.

Figura 8. Arquitectura usada por el sistema físico



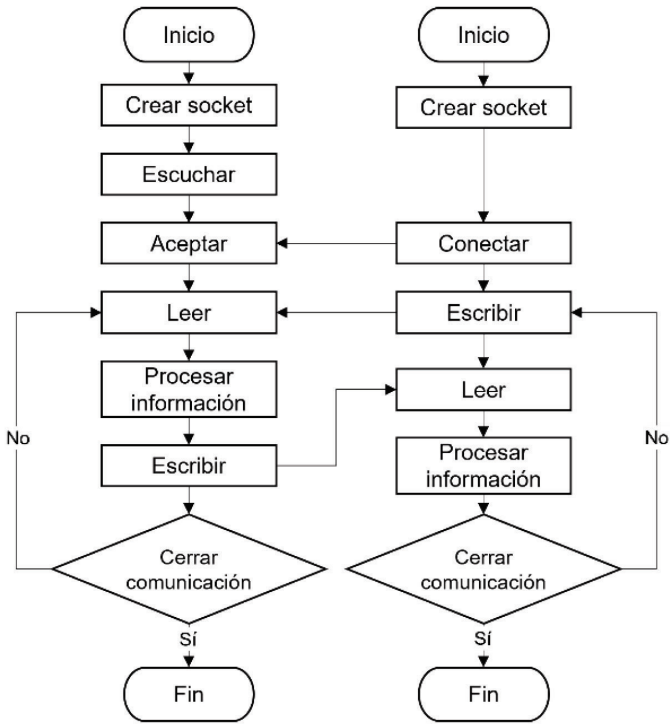
Fuente: elaboración propia.

Inicialmente, se hace la captura de video utilizando una cámara estéreo de doble lente, y el video capturado se convierte en imágenes en tiempo real. Es necesario separar cada *frame* del video en dos partes, correspondientes al lente izquierdo y al lente derecho, para poder realizar el proceso de triangulación de manera precisa. Por medio del procesamiento de las imágenes y la segmentación basada en color y forma se identifica el objeto o pieza deseada en las imágenes capturadas y se delimita su contorno, a partir de lo cual se calcula el centro en pixeles del objeto en cada imagen, utilizando el método de los momentos, a fin de realizar la estimación de posición. Una vez calculado este valor, se envía una cadena de caracteres, mediante un enrutador wifi, al controlador del robot, utilizado *sockets* de internet. Luego, el controlador integrado del robot extrae las coordenadas de la cadena de caracteres recibida y lleva a cabo las acciones necesarias para posicionar el robot en dichas coordenadas.

Comunicación socket

Los *sockets* se interpretan como puertas bidireccionales por las cuales se intercambia información entre dos programas, según la arquitectura de cliente-servidor, de modo que el servidor realiza el control del servicio enviando información al cliente cuando este lo solicite. Para poder realizar la conexión, es necesario conocer la IP (el número de identificación de un dispositivo conectado a una red, ya sea un celular, un computador, etc.) y el puerto de enlace, el cual se entiende como el canal por el cual circulan los datos. Por lo anterior, se escogió el puerto 8000. La dirección IP del robot por defecto es la 192.168.125.1, mientras que la dirección IP del computador portátil debe ser dinámica (protocolo DHCP) y asignada por el controlador del robot al momento de la conexión, debido a que debe estar en el mismo segmento de red que el robot. El diagrama de flujo de la conexión por *socket* se muestra en la figura 9 [9], [10], en donde el computador portátil toma el rol de cliente y el controlador del robot el de servidor.

Figura 9. Diagrama de flujo de la comunicación socket

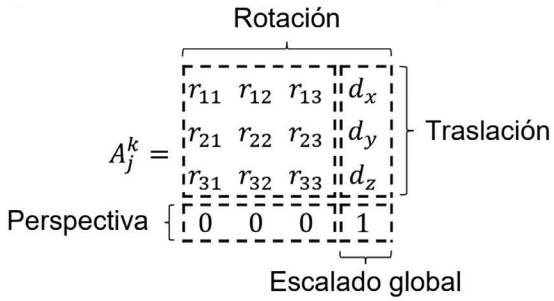


Fuente: elaboración propia.

Matriz extrínseca para referir el punto 3D del origen de coordenadas de la cámara al origen de coordenadas del robot

La matriz extrínseca obtenida mediante Matlab permite conocer los parámetros de rotación y traslación de una cámara respecto a la otra, pero no permite conocer la transformación del origen de coordenadas de la cámara a las del robot. El origen de coordenadas del robot se encuentra en su base, siendo imposible colocar la cámara en esa posición, por lo que se hace necesario trasladar y rotar el origen de coordenadas de la cámara al origen de coordenadas del robot mediante una transformación lineal. Para solventar este inconveniente se estimó una matriz de transformación homogénea que contiene la transformación de rotación y translación (figura 10).

Figura 10. Matriz de transformación homogénea



Fuente: elaboración propia.

Esta matriz permite plantear la ecuación matricial (13):

$$P_{Robot} = A \cdot P_{Cámara} \tag{13}$$

Donde $P_{Cámara}$ es un punto 3D que tiene como origen de coordenadas el origen de coordenadas de la cámara, y P_{Robot} es el mismo punto $P_{Cámara}$, pero con el origen de coordenadas del robot.

Si en (13) reemplazamos $P_{Cámara}$, P_{Robot} por puntos análogos conocidos, se genera un sistema de tres ecuaciones con doce incógnitas, por lo que es necesario contar con tres pares de puntos adicionales que generen un sistema de doce ecuaciones con doce incógnitas que satisfacen la ecuación matricial (14).

$$P_{Robot} = M \cdot V_{Parámetros} \tag{14}$$

Aquí, P_{Robot} es un vector compuesto por puntos conocidos del robot (15), $V_{Parámetros}$ es un vector compuesto por los parámetros desconocidos de la matriz A (16), y M es una matriz 12x12 compuesta por puntos conocidos de la cámara (16).

$$P_{Robot} = \begin{bmatrix} P1_{Rx} \\ P1_{Ry} \\ P1_{Rz} \\ P2_{Rx} \\ P2_{Ry} \\ P2_{Rz} \\ P3_{Rx} \\ P3_{Ry} \\ P3_{Rz} \\ P4_{Rx} \\ P4_{Ry} \\ P4_{Rz} \end{bmatrix} \tag{15}$$

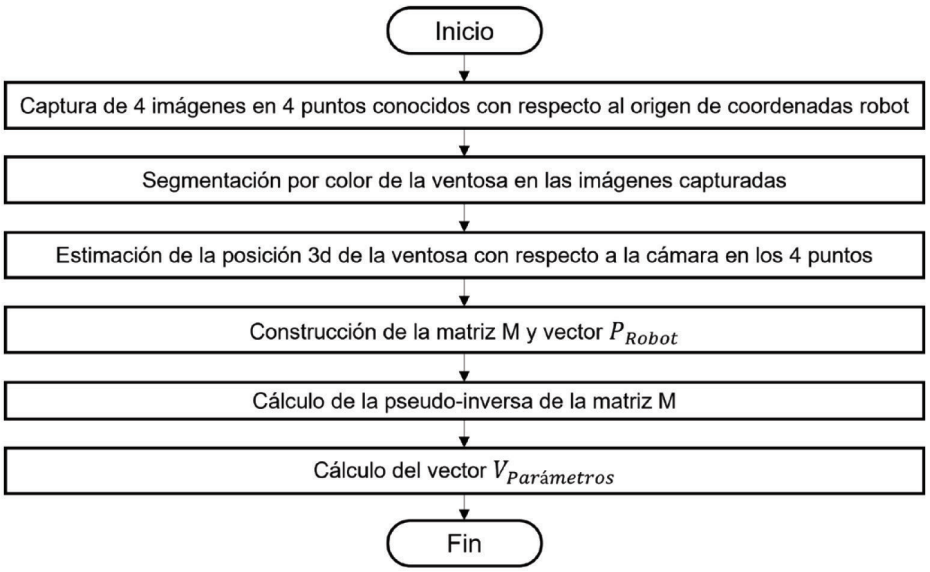
$$V_{Parámetros} = \begin{bmatrix} r_{11} \\ r_{12} \\ r_{13} \\ d_x \\ r_{21} \\ r_{22} \\ r_{23} \\ d_y \\ r_{31} \\ r_{32} \\ r_{33} \\ d_z \end{bmatrix}; M = \begin{bmatrix} P1_x & P1_y & P1_z & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & P1_x & P1_y & P1_z & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & P1_x & P1_y & P1_z & 1 \\ P2_x & P2_y & P2_z & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & P2_x & P2_y & P2_z & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & P2_x & P2_y & P2_z & 1 \\ P3_x & P3_y & P3_z & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & P3_x & P3_y & P3_z & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & P3_x & P3_y & P3_z & 1 \\ P4_x & P4_y & P4_z & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & P4_x & P4_y & P4_z & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & P4_x & P4_y & P4_z & 1 \end{bmatrix} \tag{16}$$

Así, el cálculo de los parámetros se consigue al aplicar (17), donde M^{-1} corresponde a la pseudo-inversa de M.

$$V_{Parámetros} = M^{-1} \cdot P_{Robot} \quad (17)$$

El algoritmo utilizado para calcular los parámetros de la matriz A sigue el diagrama de flujo mostrado en la figura 11.

Figura 11. Diagrama de flujo del algoritmo que calcula la matriz extrínseca de la cámara

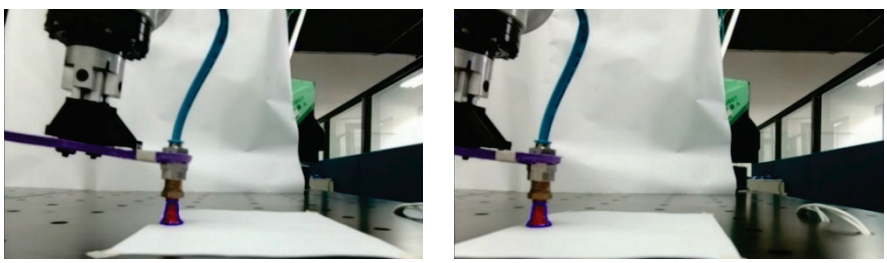


Fuente: elaboración propia.

Un ejemplo de las imágenes tomadas para el cálculo de la matriz de transformación se puede ver en las figuras 12-a y 12-b, las cuales corresponden al primer punto, observándose que el algoritmo de segmentación delimita de manera fiable la ubicación de la ventosa del robot en ambas im*á*genes por medio de un contorno de color azul.

Figura 12. a) Primer punto para la matriz extrínseca, cámara izquierda; b) primer punto para la matriz extrínseca, cámara derecha

a) b)

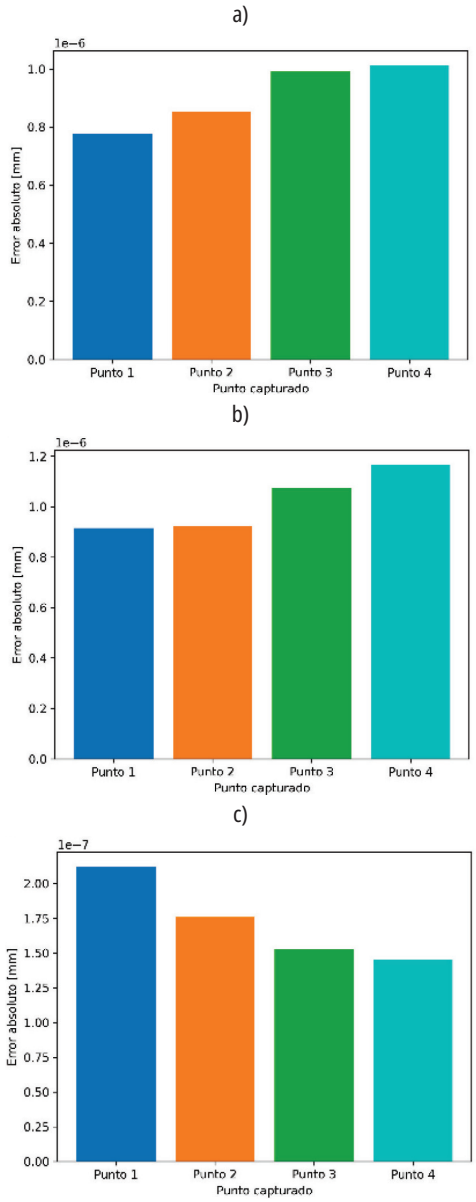


Fuente: elaboración propia.

Reorganizando el resultado obtenido al implementar el algoritmo de la figura 11 se obtuvo la matriz de transformación mostrada en (18).

$$\begin{bmatrix} 0.9811 & 0.0349 & -0.0027 & 470.7006 \\ -0.0003 & -1.4854 & 0.9140 & -208.0775 \\ 0.0209 & -1.6518 & -0.0004 & 41.4606 \\ 0.000 & 0.000 & 0.000 & 1.0000 \end{bmatrix} \quad (18)$$

Figura 13. a) Error absoluto en el eje *X* del valor calculado por la matriz y el valor convencionalmente verdadero; b) error absoluto en el eje *Y* del valor calculado por la matriz y el valor convencionalmente verdadero; c) error absoluto en el eje *Z* del valor calculado por la matriz y el valor convencionalmente verdadero



Fuente: elaboración propia.

Por su parte, las figuras 13-a, 13-b y 13-c muestran el error absoluto entre los cuatro puntos predefinidos conocidos donde se situó la ventosa, con respecto al origen de coordenadas del robot, así como los puntos calculados utilizando la estimación 3D a partir de los datos de la cámara y la matriz de transformación (18); en estas figuras, se puede observar que el error absoluto según la matriz (18) es cercano a cero, confirmando que la estimación de la matriz *A* con este es confiable.

Corrección de la posición

El algoritmo de corrección de este estudio se basó en el algoritmo de optimización iterativo Gradient Descent y en el proceso de coordinación mano-ojo que realizan los seres humanos. El algoritmo Gradient Descent es utilizado para encontrar mínimos de una función con múltiples parámetros, para lo cual hace uso del gradiente de la función, que le permite "guiar" al algoritmo para que de manera progresiva se acerque al mínimo de dicha función.

La ecuación (19) es la base del algoritmo Gradient Descent, donde $\nabla f(\theta^k)$ es el gradiente de la función evaluado en un vector de parámetros semilla θ^k, θ^{k+1} corresponde al valor actualizado del vector de parámetros, y β es una contante que varía entre $0-1$, utilizada para evitar la divergencia del algoritmo. El proceso iterativo del algoritmo termina cuando la diferencia entre θ^{k+1} y θ^k está dentro de una tolerancia determinada [11], [12].

$$\theta^{k+1} = \theta^k - \beta \nabla f(\theta^k) \quad (19)$$

La ecuación (20) es la base para el algoritmo de corrección de este trabajo.

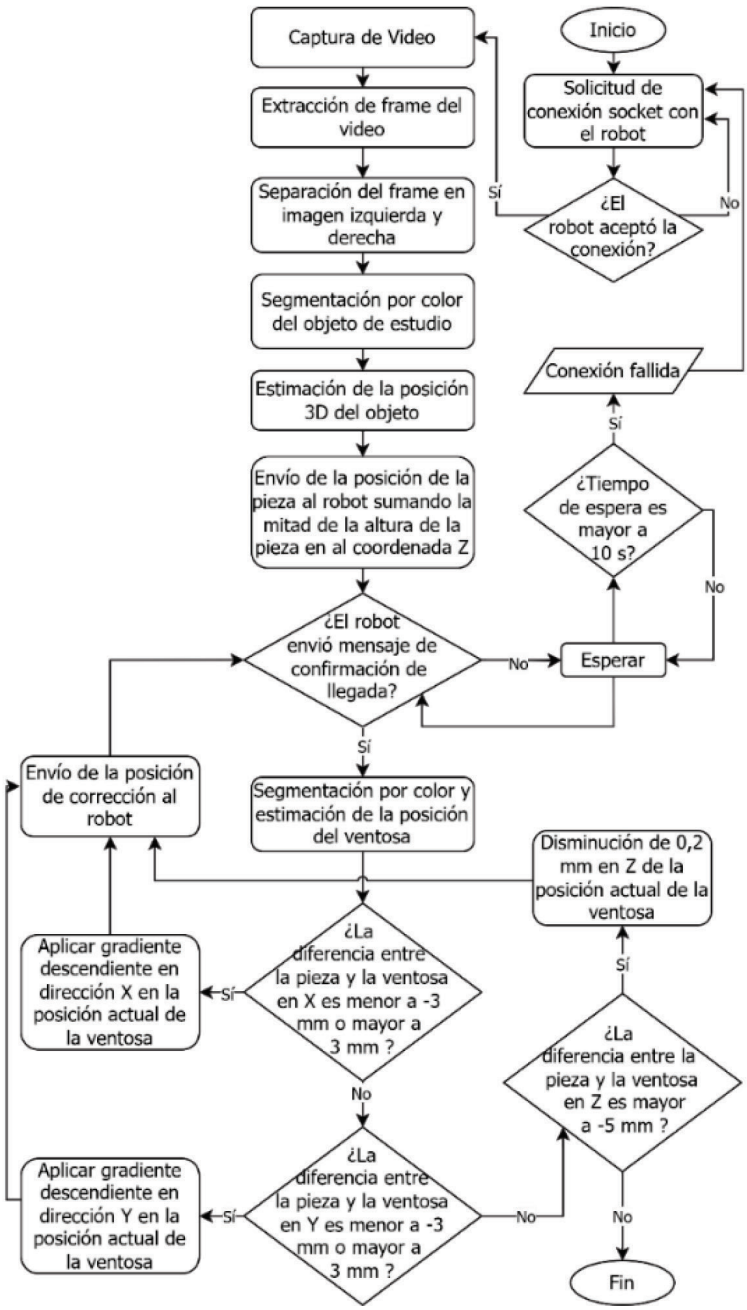
$$P^{k+1} = P^k + \beta(P_{objeto} - P^k) \quad (20)$$

Aquí, P^k representa la posición espacial de la ventosa antes de ser corregida (estimada por la cámara); P_{objeto} es la posición espacial del objeto (la cual se mantiene constante); P^{k+1} representa la posición espacial corregida de la ventosa; y β tiene la misma función que en (19). La ecuación (20) puede ser aplicada por cada componente *X*, *Y* y *Z*, obteniendo (21):

$$X^{k+1} = X^k + \beta\left(X_{objeto} - X^k\right);\ Y^{k+1} = Y^k + \beta\left(Y_{objeto} - Y^k\right);\ Z^{k+1} = Z^k + \beta\left(Z_{objeto} - Z^k\right) \quad (21)$$

Para evitar que el proceso de corrección en el eje Z bajara más allá del nivel de la mesa de trabajo o aplastara el objeto, solo se aplicó esta estrategia en las componentes *X* y *Y*. En Z se aplicó una aproximación diferente, la cual se puede observar en el diagrama de flujo del algoritmo implementado (figura 14).

Figura 14. Diagrama de flujo de la corrección de la posición



Fuente: elaboración propia.

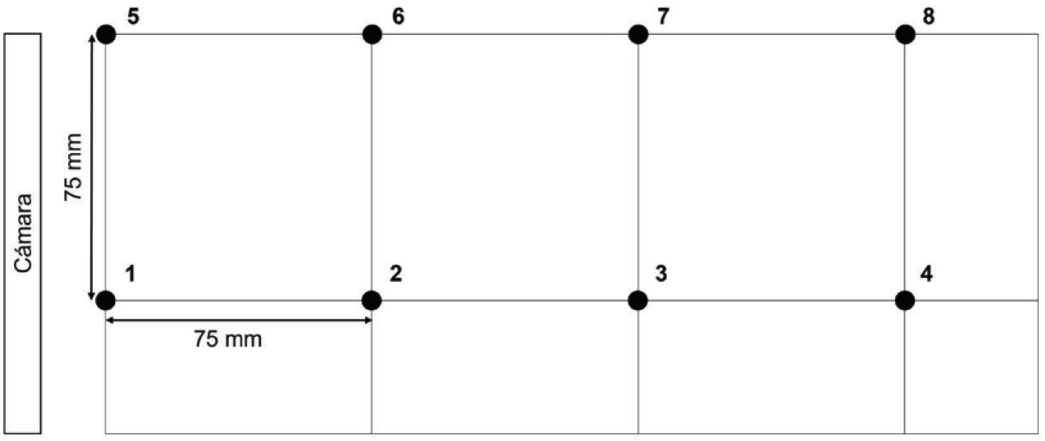
El proceso de corrección del diagrama de flujo anterior es similar al que empleamos con las manos para ejecutar una tarea determinada de manera simultánea y coordinada con base en información visual, proceso conocido como coordinación mano-ojo, asimilando la cámara con los ojos y el robot con la mano.

Resultados y discusión

Montaje del experimento

Las medidas tomadas se realizaron sobre una superficie horizontal marcada con una cuadrícula de cuatro columnas y dos filas, cuyos cuadros tienen una longitud horizontal y vertical de (figura 15).

Figura 15. Distribución de los puntos en donde se realizaron las mediciones



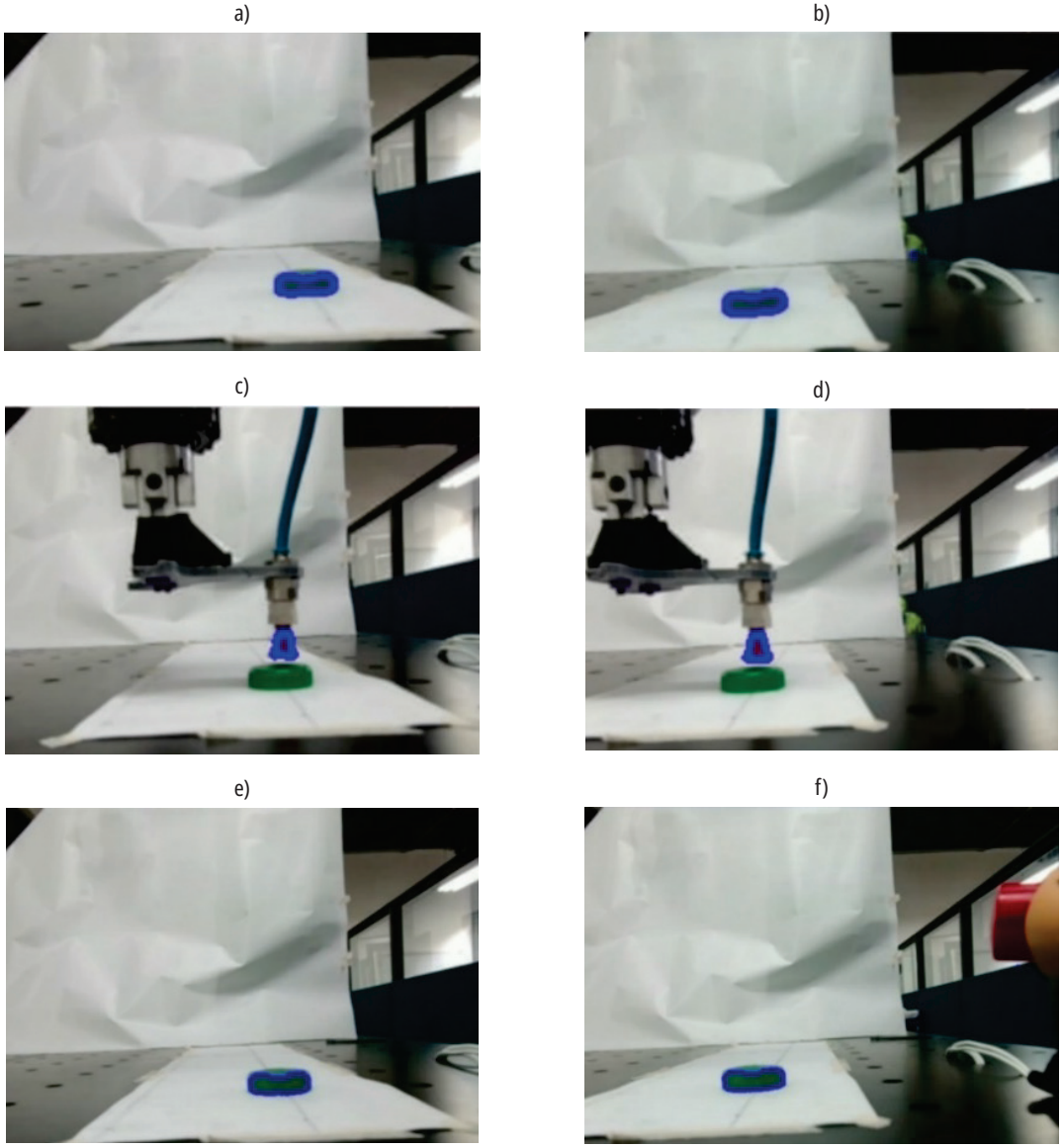
Fuente: elaboración propia.

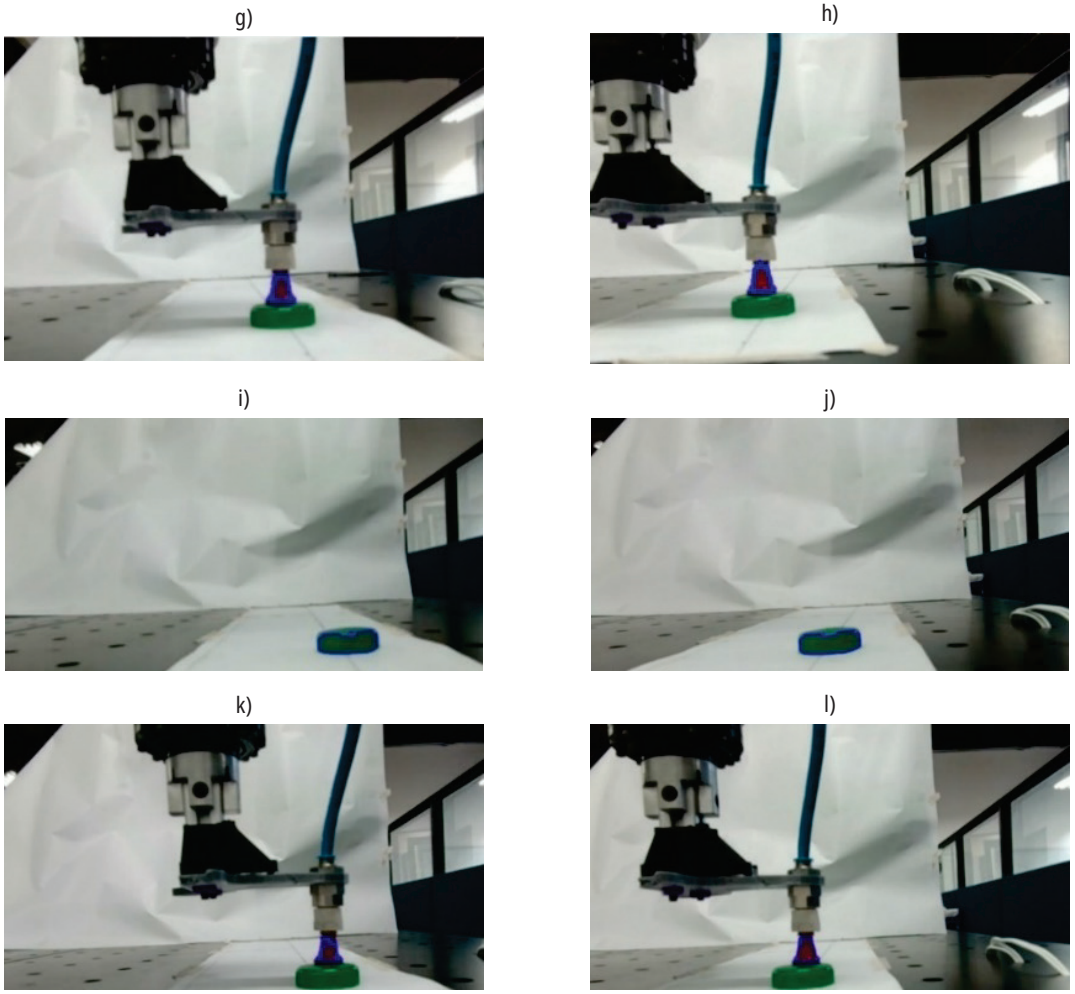
El centro de la pieza utilizada se posicionó en cada uno de los puntos mostrados en la figura 15. Posteriormente, se ejecutó el algoritmo de estimación y corrección de la coordenada 3D del objeto y se tomó la medida correspondiente en la posición final del robot. Este proceso se realizó en tres resoluciones: 320x240, 640x480 y 1280x720 pixeles.

Resultados de la detección en diferentes resoluciones

El resultado de la detección de la pieza en ambas cámaras a una resolución de pixeles se puede observar en las figuras 16-a y 16-b, donde el borde azul corresponde al contorno de la pieza detectado por el algoritmo. En lo que respecta a la detección de la ventosa, los resultados de ambas cámaras a una resolución de pixeles se observan en las figuras 16-c y 16-d, pudiendo apreciarse que la detección de la ventosa es bastante precisa. Cuando se aumenta la resolución de la captura de las imágenes a y pixeles, la precisión de la detección de la pieza (figuras 16-e, 16-f, 16-i y 16-j) y la ventosa (figuras 16-g, 16-h, 16-k y 16-l) en ambas cámaras no cambia. Al analizar la figura 16-k y compararla con la figura 16-l, se puede ver un pequeño error de detección, debido a que la superficie de la ventosa es ligeramente reflectante, por lo que el tono rojo no es interpretado exactamente igual por el algoritmo, dados los cambios de iluminación. Así, en algunas ocasiones se presentarán problemas de detección si las condiciones de iluminación de la habitación no son favorables, ya que el algoritmo tiene problemas para detectar tonos de color en los extremos de saturación y valor del espacio de color HSV.

Figura 16. a) Detección de la pieza cámara izquierda, resolución 320x240; b) detección de la pieza cámara derecha, resolución 320x240; c) detección de la ventosa cámara izquierda, resolución 320x240; d) detección de la ventosa cámara derecha, resolución 320x240; e) detección de la pieza cámara izquierda, resolución 640x480; f) detección de la pieza cámara derecha, resolución 640x480; g) detección de la ventosa cámara izquierda, resolución 640x480; h) detección de la ventosa cámara derecha, resolución 640x480; i) detección de la pieza cámara izquierda, resolución 1280x720; j) detección de la pieza cámara derecha, resolución 1280x720; k) detección de la ventosa cámara izquierda, resolución 1280x720; l) detección de la ventosa cámara derecha, resolución 1280x720



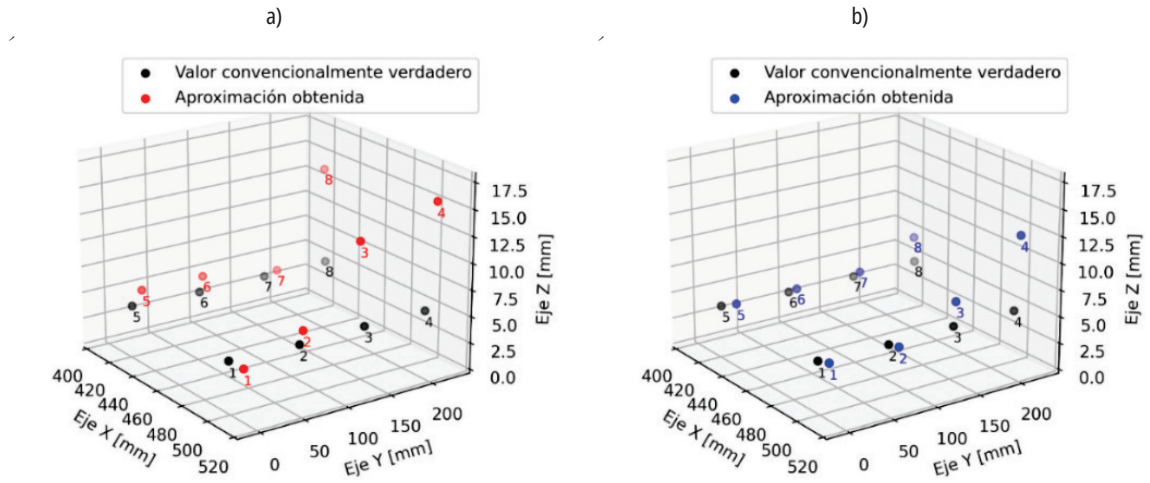


Fuente: elaboración propia.

Resultados de la estimación de la posición

Para efectos del experimento, se consideró el valor convencionalmente verdadero como el que se obtiene cuando se posiciona manualmente al robot sobre el centro de la pieza, conseguido directamente del valor de posición ofrecido por el controlador del robot. Es importante aclarar que la incertidumbre de posicionamiento del robot es de 0.05 mm. El valor medido corresponde al valor experimental de la posición final del robot al terminar el proceso de corrección de la posición, el cual, al igual que en el caso del valor convencionalmente verdadero, fue obtenido directamente del controlador del robot. Al graficar los resultados del valor convencionalmente verdadero vs el valor obtenido a una resolución de 320x240 (figura 17-a), se puede observar que hay una correlación clara entre la distancia de la pieza a la cámara y la precisión del resultado, evidenciándose que a mayor distancia menor es la precisión con la que llega el robot al punto esperado. Este resultado se debe a que, mientras más lejos esté el objeto de la cámara, el algoritmo presenta detecciones incompletas del color de objeto, lo que afecta de manera negativa el resultado de la estimación. Otro factor influyente es que, al alejarse el objeto de la cámara, este cubre pocos pixeles en la imagen, perdiéndose información. El valor más afectado por este error de estimación es el correspondiente al eje *Z*, donde hay una diferencia de entre el valor convencionalmente verdadero y el valor obtenido.

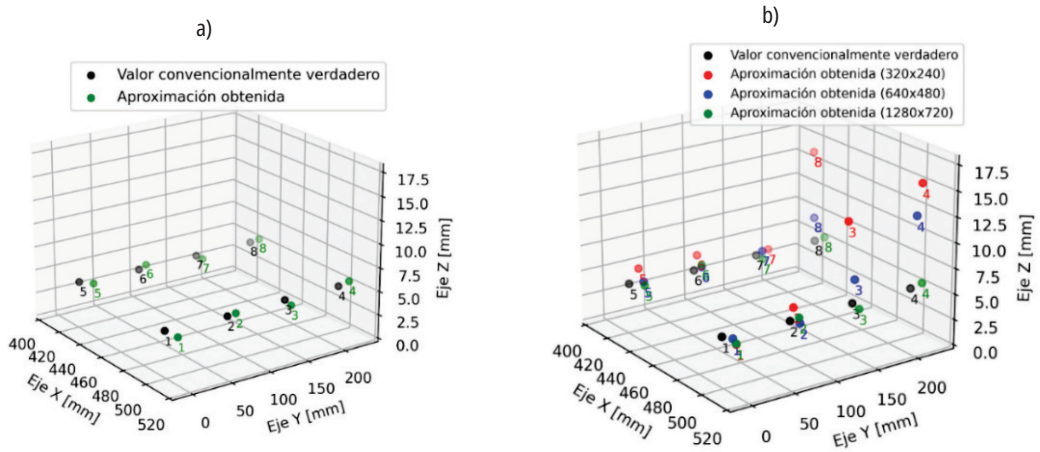
Figura 17. a) Valor convencionalmente verdadero vs valor obtenido en una resolución de ; b) valor convencionalmente verdadero vs valor obtenido en una resolución de 640 x 480



Fuente: elaboración propia.

Al cambiar la resolución a 640x480 pixeles, se obtiene la gráfica mostrada en figura 17-b, en la cual se evidencia una mejora significativa en los resultados obtenidos, siendo estos más cercanos al valor esperado. A pesar de la mejoría en la estimación, aún se puede observar que prevalece la tendencia de imprecisión en los puntos más alejados de la cámara, particularmente en el punto 4, donde hay una diferencia de 7 mm en el eje *Z*. Si se capturan las imágenes a una resolución 1280x720 pixeles, se obtiene la gráfica mostrada en la figura 18-a, la cual muestra los mejores resultados entre las tres resoluciones probadas. El error significativo que se encontró en el eje *Z* del punto 4 de las figuras 17-a y 17-b ya no se presenta en esta resolución. Finalmente, si se presentan todos los resultados en una misma gráfica (figura 18-b), se puede evidenciar cómo claramente el mejoramiento de la resolución de la imagen aumenta la precisión de la estimación de la posición del objeto.

Figura 18. a) Valor convencionalmente verdadero vs valor obtenido en una resolución de 1280x720; b) comparación entre el valor convencionalmente verdadero vs valor obtenido en todas las resoluciones



Fuente: elaboración propia.

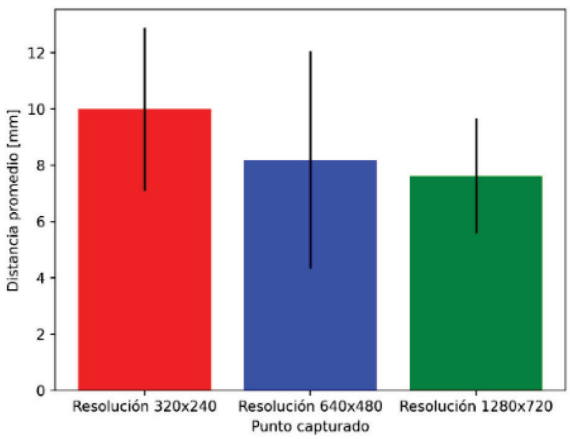
Distancia promedio de la estimación

Para calcular el error absoluto se separaron las coordenadas correspondientes para los tres ejes, datos que a su vez fueron discriminados por resolución y comparados con el valor convencionalmente verdadero, correspondiente al valor de colocar al robot manualmente sobre el centro de la pieza; hecho esto, se utilizó la fórmula de la distancia entre dos puntos en el espacio tridimensional (22).

$$P = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix};\ P_{cv} = \begin{bmatrix} X_{cv} \\ Y_{cv} \\ Z_{cv} \end{bmatrix};\ D = \sqrt{(X - X_{cv})^2 + (Y - Y_{cv})^2 + (Z - Z_{cv})^2} \quad (22)$$

Al promediar las distancias en cada uno de los puntos para cada resolución se obtuvo el resultado presentado en la figura 19.

Figura 19. Distancia promedio. Valor de la pieza obtenido vs valor convencionalmente verdadero



Fuente: elaboración propia.

La figura 19 evidencia que, en promedio, los errores no tienden a superar los de distancia entre el punto convecionalmente verdadero y el obtenido con el algoritmo de corrección. Esto permite afirmar que la estimación de la posición proporcionada por el algoritmo es confiable, siempre y cuando la resolución de la captura de la imagen sea lo más alta posible y las condiciones de iluminación sean óptimas. Para calcular estos errores, es importante tener en cuenta que los datos fueron tomados de forma secuencial a lo largo de un día, iniciando con la resolución de 1280x720 pixeles y terminando con la resolución de , lo cual implicó que cada tanda de medición tuviera una iluminación natural diferente, algo que pudo influir negativamente en los resultados obtenidos.

Conclusiones

Este estudio reafirma la utilidad de la geometría estéreo para estimar las coordenadas espaciales de un objeto. Utilizando dos cámaras con parámetros intrínsecos y extrínsecos calibrados, se logró calcular la posición tridimensional de un objeto con una precisión de 10 mm gracias a la adaptación del algoritmo de optimización Gradient Descent y la coordinación mano-ojo al problema específico de agarre robótico abordado en este trabajo.

La segmentación por color en el espacio HSV mostró ser efectiva para detectar objetos de interés en las imágenes capturadas. Además, el método de los momentos se utilizó con éxito para calcular el centro de los objetos detectados, lo que fue fundamental para la estimación de la posición. Sin embargo, la segmentación de objetos basada en color es muy susceptible a cambios de iluminación y superficies reflectantes, por lo cual solo se debe utilizar en ambientes controlados, como los de un laboratorio, o entornos industriales con buena iluminación y pocos cambios en ella.

La arquitectura del sistema utilizada en este estudio, que implica captura de video, procesamiento de imágenes y comunicación con el brazo robótico a través de *sockets* TCP/IP, demostró ser robusta y eficiente para lograr la interacción en tiempo real entre visión artificial y control robótico.

Se observó que la resolución de las imágenes desempeña un papel fundamental en la precisión de la estimación de posición, dado que a medida que se aumenta la resolución, la precisión mejora considerablemente. Esto respalda la importancia de adquirir imágenes de alta calidad en aplicaciones de visión artificial, si bien, pesar de los buenos resultados, se identificaron errores de estimación,

especialmente en la distancia Z, que pueden atribuirse a la distancia entre el objeto y la cámara.

Trabajos futuros

La estrategia de segmentación basada en color es muy sensible a los cambios de iluminación, lo que afecta negativamente al algoritmo implementado, por lo que en trabajos futuros se espera trabajar con algoritmos de detección y segmentación de objetos más sofisticados que utilizan redes neuronales convolucionales (CNN por sus siglas en inglés), como You Only Look Once (YOLO), Single Shot Detector (SDD) y Faster- Region Based Convolutional Neural Networks (Faster-RCNN).

Referencias

[1] F. von Drigalski, K. Hayashi, Y. Huang, R. Yonetani, M. Amaya, K. Tanaka, *et al*., "Precise Multi-Modal In-Hand Pose Estimation using Low-Precision Sensors for Robotic Assembly", *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 968–974. http://doi.org/10.1109/ICRA48506.2021.9561222.

[2] C. Li y Q. Bi, "Vision-driven High Precision Positioning Method for Bracket Assembly with Industrial Robot", *2022 5th World Conference on Mechanical Engineering and Intelligent Manufacturing (WCMEIM)*, 2022, pp. 825–830. http://doi.org/10.1109/WCMEIM56910.2022.10021493.

[3] D. A. Forsyth y J. Ponce, *Computer vision: a modern approach*, 2ª ed., New Jersey: Pearson, 2012.

[4] R. Hartley y A. Zisserman, *Multiple View Geometry in computer vision*. New York: Cambridge University Press, 2003.

[5] Z. Zhang, "A flexible new technique for camera calibration", *IEEE Trans Pattern Anal Mach Intell.*, vol. 22, no. 11, pp. 1330–1334, 2000, http://doi.org/10.1109/34.888718.

[6] J. Heikkila y O. Silven, "A four-step camera calibration procedure with implicit image correction", *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, IEEE Comput. Soc, pp. 1106–1112. http://doi.org/10.1109/CVPR.1997.609468.

[7] H.-C. Kang, H.-N. Han, H.-C. Bae, M.-G. Kim, J.-Y. Son, y Y.-K. Kim, "HSV Color-Space-Based Automated Object Localization for Robot Grasping without Prior Knowledge", *Applied Sciences*, vol. 11, no. 16, p. 7593, 2021, http://doi.org/10.3390/app11167593.

[8] D. Giuliani, "Metaheuristic Algorithms Applied to Color Image Segmentation on HSV Space", *J Imaging*, vol. 8, no. 1, p. 6, 2022, http://doi.org/10.3390/jimaging8010006.

[9] H. Zhang, G. Guan, H. Zhao, X. Liu, L. Xue, y C. Xiong, "Design of image transmission system based on TCP/IP protocol", en *2022 IEEE 4th International Conference on Civil Aviation Safety and Information Technology (ICCASIT)*, IEEE, 2022, pp. 617–622. http://doi.org/10.1109/ICCASIT55263.2022.9987111.

[10] IBM Corporation, "Sockets", IBM Corporation. Consultado: el 26 de julio de 2023. [En línea]. Disponible en: https://www.ibm.com/docs/es/aix/7.3?topic=concepts-sockets

[11] T. Sun, K. Tang, y D. Li, "Gradient Descent Learning With Floats", *IEEE Trans Cybern*, vol. 52, no. 3, pp. 1763–1771, 2022, http://doi.org/10.1109/TCYB.2020.2997399.

[12] J. Fliege, A. I. F. Vaz, y L. N. Vicente, "Complexity of gradient descent for multiobjective optimization", *Optim Methods Softw*, vol. 34, no. 5, pp. 949–959, 2019, http://doi.org/10.1080/10556788.2018.1510928.