UNIVERSIDAD PRIVADA ANTENOR ORREGO

FACULTAD DE INGENIERÍA

ESCUELA PROFESIONAL DE INGENIERÍA ELECTRÓNICA

TESIS PARA OBTENER EL TÍTULO PROFESIONAL

DE INGENIERO ELECTRÓNICO

"INTEGRACIÓN DE UN SISTEMA DE VISIÓN POR COMPUTADOR A UN ROBOT CARTESIANO CONTROLADO POR UN ORDENADOR DE PLACA DE RECURSOS COMPUTACIONALES LIMITADOS PARA LA GENERACIÓN DE TRAYECTORIAS DURANTE EL PROCESO DE REPIQUE DE PLANTINES DE ALCACHOFA EN UN ENTORNO DE INTERIOR"

Área de investigación:
Procesamiento Digital de Señales e Imágenes

AUTOR: Br. Linares Otoya, Paulo Enrique

JURADO EVALUADOR:

Presidente: Alva Alarcón, Jorge Luis

Secretario: Llanos León, Lenin Humberto

Vocal: Linares Vertiz, Saul Noe

ASESOR: MSc. Gonzalez Cadenillas, Clayder Alejandro

CÓDIGO ORCID: https://orcid.org/0000-0002-6777-4479

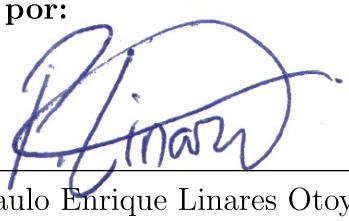
TRUJILLO - PERÚ

2021

Fecha de sustentación: 2021/05/07

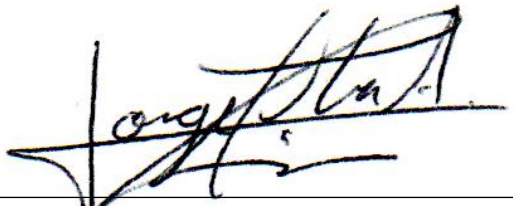
"INTEGRACIÓN DE UN SISTEMA DE VISIÓN POR COMPUTADOR A UN ROBOT CARTESIANO CONTROLADO POR UN ORDENADOR DE PLACA DE RECURSOS COMPUTACIONALES LIMITADOS PARA LA GENERACIÓN DE TRAYECTORIAS DURANTE EL PROCESO DE REPIQUE DE PLANTINES DE ALCACHOFA EN UN ENTORNO DE INTERIOR"

Elaborado por:

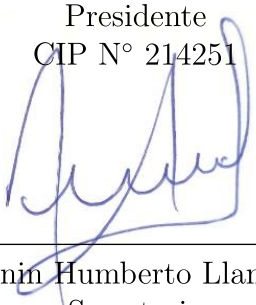


Br. Paulo Enrique Linares Otoya
Tesista

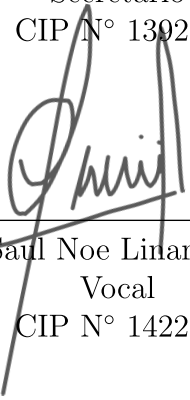
Aprobado por:



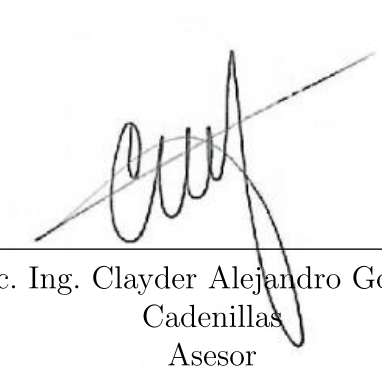
Ing. Jorge Luis Alva Alarcón
Presidente
CIP N° 214251



Ing. Lenin Humberto Llanos León
Secretario
CIP N° 139213



Ing. Saul Noe Linares Vertiz
Vocal
CIP N° 142213



MSc. Ing. Clayder Alejandro González Cadenillas
Asesor
CIP N° 240498

DEDICATORIA

A mis padres Luis Linares C. y Virginia Otoya E., por ayudarme en los momentos más difíciles de mi vida.

A mi abuela Marina Espinoza.

A mis hermanos Virginia Linares O. y Luis Linares O. .

AGRADECIMIENTOS

El presente trabajo fue parcialmente financiado por el Vicerrectorado de Investigación de la Universidad Privada Antenor Orrego, a través del proyecto de investigación "Desarrollo y construcción de un sistema robotizado para optimizar el repique de plantines en viveros industriales de la región la libertad-Peru" con Resolución Rectoral RR-4231-2017-R-UPAO y dirigida por el Dr. Ing. Sixto Ricardo Prado Gardini, al que expreso mi total gratitud por la acertada orientación, el soporte y discusión crítica que me permitió un buen aprovechamiento en el trabajo realizado, y que esta tesis llegara a buen término.

A mi asesor MSc. Clayder Alejandro Gonzalez Cadenillas por su constante apoyo durante mis estudios de pregrado y durante mi carrera como investigador. Y por sus acertadas sugerencias durante la redacción de este documento.

A mis padres, abuela y hermanos por alentarme siempre a lo largo de mis estudios de pregrado. Por apoyarme en todos los proyectos en que he participado y por ayudarme a seguir adelante a pesar de las adversidades.

RESUMEN

En la presente investigación se desarrolló una secuencia base de algoritmos propios de un sistema de visión por computador y se integraron sus resultados a un robot cartesiano marca Farmbot, de manera que fue posible automatizar el proceso de repique de plantines de alcachofa en un entorno de interior (i.e. no en un campo abierto). Como primer paso, se realizó una revisión bibliográfica con la finalidad de identificar los parámetros más relevantes al momento de clasificar un plantín de alcachofa de acuerdo a su calidad. Adicionalmente se realizaron visitas a viveros especializados en la tarea de producción y repique de plantines hortícolas, como Agrogénesis, con la finalidad de obtener información relacionada al proceso de control de calidad y clasificación de plantines de alcachofa en La Libertad. Como segundo paso se realizó una revisión bibliográfica respecto al estado del arte de la utilización de sistemas de visión por computador en la agroindustria, específicamente aquellos relacionados con el control de calidad de plantas hortícolas en su etapa temprana. Tercero, se realizó una secuencia de calibración de cámara, utilizando el método de Zhang y un patrón de calibración del tipo patrón de ajedrez. Cuarto, se desarrolló una secuencia base de algoritmos de visión por computador con la finalidad de lograr clasificar plantines con una edad de 20 dias después de la siembra, en tres clases según calidad. Quinto, se utilizaron los parámetros intrínsecos y extrinsecos obtenidos durante la calibración de la cámara para localizar los plantines dentro del espacio de trabajo del robot cartesiano. Finalmente, se ejecutó el proceso de repique de plantines en 25 ocasiones, evaluando el performance del clasificador y del repique en general en términos de tiempo y aciertos. Para evaluar al clasificador se consideró como métrica principal al *F1-Score* obteniendo valores de 0.941 al clasificar plantines de clase A, 0.785 al clasificar plantines de clase B y 0.88 para plantines de clase C. Además se obtuvo un porcentaje de aciertos durante el transplante (i.e. manipulación correcta de plantines usando el gripper) de $20 - 30\,\%$.

ABSTRACT

In the present research, a base sequence of algorithms typical of computer vision systems was developed and its results were integrated into a Farmbot Cartesian robot, so that it was possible to automate the process of classifying/transplanting artichoke seedlings in an indoor environment (i.e. not in an open field). As a first step, a bibliographic review was carried out in order to identify the most relevant parameters when classifying an artichoke seedling according to its quality. In addition, other fellows and I made visits to nurseries specialized in the tasks of producing and quality control of horticultural seedlings and, such as Agrogénesis, in order to obtain information related to the quality control process and classification of artichoke seedlings in La Libertad. As a second step, a bibliographic review was carried out regarding the state of the art of the use of computer vision systems in the agricultural industry, specifically those related to the quality control of horticultural plants in their early stage. Third, a camera calibration sequence was performed, using the "Zhang method" and a chess calibration pattern. Fourth, a base sequence of computer vision algorithms was developed in order to classify plants with an age of 20 days after sowing, into three classes according to quality. Fifth, the intrinsic and extrinsic parameters obtained during the camera calibration were used to locate the seedlings within the workspace of the Cartesian robot. Finally, the seedling classifying/transplanting process was executed 25 times, evaluating the performance of the classifier and the transplanting process in general in terms of time and hits. In order to evaluate the classifier, the *F1-Score* was considered as the main metric, obtaining values of 0.941 when classifying class A seedlings, 0.785 when classifying class B seedlings and 0.88 for class C seedlings. In addition, the percentage of success obtained during transplantation (i.e. correct handling of seedlings using the gripper) had values between $20 - 30\,\%$.

Índice general

Índice de figuras

Índice de Tablas

Notación utilizada

Definiciones en visión por computador

$I(u, v)$: Imagen digital

u , v : Coordenadas fila-columna dentro de una imagen digital.

$L(u, v)$: Componente de iluminación dentro de una imagen digital.

$R(u, v)$: Componente de reflexión dentro de una imagen digital.

$B(u, v)$: Imagen digital binaria

Geometría utilizada en visión por computador

$\mathbf{M}$: Matriz de proyección de un punto en el espacio sobre el plano imagen.

W : Sistema de coordenadas del robot cartesiano.

D : Sistema de coordenadas del plano imagen discretizado.

f_x, f_y : Distancia focal en eje X y Y.

s_x, s_y : Escala de milímetros-pixeles en el plano imagen.

$\mathbf{u_c}$: Pixel correspondiente al centro de proyección de la cámara

$\mathbf{A}$: Matriz de parámetros intrínsecos de la cámara.

$\mathbf{W}$: Matriz de parámetros extrínsecos de la cámara.

$\mathbf{R}$: Matríz de rotación correspondiente a los parámetros extrínsecos.

$\mathbf{t}$: Vector de traslación correspondiente a los parámetros extrínsecos.

$\mathbf{H}$: Matríz de homografía.

$\mathbf{k}$: Vector de parámetros de distorsión de la cámara.

$\mathbf{P_n}$: Posición absoluta del vértice n del patrón de calibración respecto al sistema del robot.

$\mathbf{u_n}$: Ubicación en pixeles de un vértice dentro del patrón de calibración.

$\mathbf{T}$: Vector que representa la posición del efector final respecto al sistema del robot.

$\tilde{\mathbf{P}}_\mathbf{n}$: Posición del vértice n del patrón de calibración respecto al sistema del efectot final.

Operaciones sobre imágenes digitales

$T(r)$	: Transformación de intensidad de una imagen $r = I_s(u, v)$.
$h(r_k)$	: Histograma de intensidad de una imagen.
$\Gamma(s, t)$	: Transformada discreta de Fourier en 2D de una imagen digital.
$H(s, t)$	: Función de un filtro en frecuencia.
$A \oplus B$	: Operación de dilatación.
$A \ominus B$	: Operación de erosión.
$s_k = u_k + jv_k$	: Contorno de una región expresado como variable compleja.
$\tilde{\mathbf{u}} = warp(\mathbf{u}, \mathbf{k})$	: Operación de warping propia de una cámara con distorsión radial.
$H_{ideal}(s, t)$	: Función de filtro en frecuencia pasa altos ideal.
$H_{Butt}(s, t)$	: Función de filtro en frecuencia pasa altos Butterworh.
$H_{Gauss}(s, t)$	: Función de filtro en frecuencia pasa altos Gaussiano.
$a(t)$	: Descriptores de Fourier de un contorno.

Capítulo I

INTRODUCCIÓN

1.1. Realidad problemática

El Ministerio de Agricultura y Riego (MINAGRI) señaló a la agricultura como la segunda actividad económica que generó más divisas al país durante el año 2019 y 2018 (Infomercado, 2020), sin embargo, según indicadores de estudios realizados por la INEI al 2018, se sitúa a la agricultura como una de las actividades con menor productividad laboral frente a la minería, construcción y manufactura. A pesar de ello, la agricultura concentra un alto porcentaje de la PEA en el Perú (24.1 %), frente a porcentajes menores al 10 % de las actividades con mayor productividad laboral, permitiendo concluir que la productividad laboral en la agricultura depende en gran magnitud de su mano de obra (Infomercado, 2018). A nivel de américa latina, se tiene un elevado déficit de una fuerza de trabajo calificada, siendo una de sus causantes, la tendencia a la migración por parte de esta, evaluándose esta tendencia como pérdidas valiosas en recursos humanos cualificados por parte de los países de origen (Gandini, 2012). En efecto, la agricultura no es ajena a esta realidad, aunque la naturaleza de su déficit sea de diferente origen. Desde comienzos de siglo XXI, se han venido desarrollando nuevas políticas que permitan mejorar la agricultura de exportación, evidenciando ciertas falencias por parte de empleados y empleadores. Debido a las exigencias en la calidad del producto de exportación y a la falta de mano de obra calificada, los empleadores en el rubro de agricultura

no tradicional de exportación se vieron obligados a realizar intensas capacitaciones a sus empleados, buscando su permanencia en el puesto para evitar futuras capacitaciones (Damiani, 2000). La realidad en el empleo agrícola en el Perú, evidencia, sin embargo, la constante falta de mano de obra cualificada, ya sea por períodos de intensa actividad o por poco interés en retener a los empleados hábiles (Chirinos, 2019). Por otra parte, la demanda de mano de obra estacional (concentrada en ciertos meses del año), característica del agro de exportación de frutas y vegetales, permite la contratación de personal no calificado para suplir la demanda (Velazco & Velazco, 2012). Según Velasco (2017), para suplir algunas tareas realizadas por humanos en la agroindustria es que en las últimas décadas se han venido utilizando máquinas que realicen las mismas actividades ya sea con la finalidad de hacer más con menos recursos, por falta de mano de obra o por los elevados costos asociados. Por otro lado, De Clercq, Vats y Biel (2018) estiman que debido a la creciente demanda en alimentos a nivel mundial, será necesario producir 70 % más alimento para el 2050, sin embargo, la escasez en recursos naturales y humanos obliga a producir más con menos. A partir de esta realidad es que se plantea la introducción de la automatización en la agricultura, con la finalidad de reducir costos de producción, reemplazar el trabajo manual tedioso y mejorar el control sobre el entorno (Edan, Han & Kondo, 2009). Edan et al. (2009) dividen a la automatización en agricultura según su campo de aplicación: maquinaria de campo (tractores, máquina de labranza, fertilizadores, cosechadores, etc.), sistemas de irrigación (temporizado y basado en sensores), automatización en invernaderos (control de clima, producción de plantines, cosechadores) y producción de frutas. A nivel nacional, se han venido desarrollando proyectos en I+D+i para alinearse con las nuevas tendencias en agricultura. Estos abarcan el uso de drones y aviones no tripulados (UAV) para la toma de imágenes aéreas, ya sean multiespectrales (para identificar estrés o enfermedades en cultivos) (Flores, Saito, Paredes & Trujillano, 2017) o imágenes de espectro visible para análisis topográfico, supervisión, detección de incendios, etc. Además, en el rubro de la conservación de frutales, existen empresas como FrioPacking que se dedican a construir soluciones de refrigeración de alimentos a escala industrial (Balza, 2019). La realidad actual de la agricultura en el Perú, indica la prioridad de cultivos intensi-

vos en terreno, debido a las condiciones climáticas y a la riqueza de nutrientes en el sustrato (Agencia Peruana de Noticias, 2019). Es por ello que crece la necesidad de superar problemas que generen ineficiencia en los cultivos propios de una agricultura intensiva. Uno de los problemas que afecta a la cadena de producción de la agricultura intensiva en su etapa inicial, es la obtención de plantines hortícolas de calidad en viveros industriales, dado que el éxito en el crecimiento de hortalizas y su rendimiento dependen en un 30 % de la calidad del plantín, ya que la baja calidad en este no podrá ser corregido en etapas posteriores (MINCETUR, 2009).

1.2. Delimitación del problema

A comienzos del año 2019, al realizar visitas a viveros industriales en la región La Libertad, especializados en la obtención de plantines hortícolas de calidad, se encontró que el proceso de repique (selección de plantines por calidad previo al trasplante) demandaba demasiados trabajadores o jornaleros lo cual implicaba mayores costos y tiempo (dependiendo de la habilidad o capacitación del jornalero). Al mismo tiempo, se encontraron ciertas prácticas que favorecerían la producción de plantines de calidad (uso de mantas para reducir luminosidad, frecuencia de regado de 2 por dia y control de temperatura y humedad) y los criterios que permitirían determinar la calidad de un plantín (dimensiones y color de la hoja y desarrollo radicular). Por otra parte, Thompson (1985) establece ciertos parámetros que pueden determinar la calidad de plantines. Algunos de ellos, como el color y la forma de las hojas, coinciden con lo observado en la realidad de los viveros de La libertad. Otros como la altura, diámetro del tallo, peso entre otros, también influirían en la determinación de la calidad del plantín, sin embargo, se consideran como prácticas invasivas (requieren manipular el plantín con la posibilidad de dañarlo). Tomando en consideración los criterios no-invasivos usados en los viveros de La libertad, se encontraron estudios que permitirían la automatización del proceso de evaluación y clasificación de plantines (disminuyendo costos y baja eficiencia de los procesos manuales) basados en la tecnología de visión por computador. En efecto, Tian,

Wang, Liu, Qiao y Li (2020) detallaronn el estado del arte en visión por computador aplicado a la clasificación y control de calidad de productos agrícolas, resaltando la capacidad de esta tecnología para realizar la inspección de calidad externa de productos agrícolas con un alto grado de flexibilidad y repetitividad a un relativo bajo costo y alta precisión. Ali Ashraf, Kondo y Shiigi (2011) desarrollaronn un sistema de visión por computador aplicado a la selección de plantines de papa para su posterior injertado. En este escenario, se precisa conocer principalmente el diámetro de los plantines, su distancia inter-nodal y la distribución (arreglo) de hojas; ya que estos determinan si es posible el injertado. El software del sistema incluye la librería OpenCV y un compilador para C++. Por otro lado, se hace uso de técnicas básicas (binarización, conteo de pixeles) para la implementación del sistema, obteniendo un 97 % de aciertos en la clasificación. Otra característica que es posible evaluar de manera rápida y sin manipulación, es el área foliar del plantín. En efecto Tong, Jiang y Zhou (2012) desarrollaron un sistema de visión por computador para automatizar el trasplante, basado en el área de las hojas. El sistema fue desarrollado en lenguaje C, utilizando la librería Matrox para el filtrado y segmentación. Se utilizaron técnicas de segmentación por color (para las hojas), binarización, conversión a escala de grises y el cálculo de área por plantín se estableció mediante una conversión de pixeles a mm^2. Se enfatiza el uso del algoritmo de segmentación por watersheds para evitar traslape de hojas, obteniendo una precisión de $95 - 99\,\%$ en la clasificación. Los estudios realizados hasta ahora en visión por computador aplicado al control de calidad son desarrollados en su mayoría en laboratorio, en condiciones controladas (principalmente de iluminación). Cuando se desea aplicar a un ambiente natural los datos pueden variar, generando errores (Tian et al. 2020). Para llevar a cabo el proceso de repique de manera automatizada, es preciso contar con un sistema de visión por computador (para la evaluación de calidad) y un sistema robótico que permita actuar de acuerdo los resultados del sistema de visión por computador. Tong et al. (2012) plantean el uso de un robot cartesiano en combinación con un sistema de visión por computador para evaluar la calidad de plantines de tomate y posteriormente clasificarlos haciendo uso del efector final propio del sistema robótico. Sin embargo, a pesar de tener una plataforma de integración de visión artificial y

una máquina, cabe resaltar que para procesar las imágenes adquiridas y generar trayectorias a la máquina, se deben superar ciertos inconvenientes, como la adecuada calibración de la cámara, el campo visual, la óptica de la cámara (enfoque), heterogeneidad en la coloración de una misma clase de objetos, iluminación y reflexión de la luz (Al-Kindi & Zughaer, 2012).

1.3. Características de la realidad problemática

- El proceso de repique de plantines en los viveros industriales requiere una elevada cantidad de personal calificado para llevarse a cabo en el menor tiempo posible y con un alto grado de repetitividad.
- La diferencia de condiciones ambientales (principalmente la iluminación y la diferencia visual entre objetos de interés y el fondo) afecta al desarrollo de un sistema robusto de visión por computador para la automatización de tareas agrícolas.
- El hardware que compone el sistema de visión por computador, así como el software (lenguaje de programación y algoritmos desarrollados) determinan el rendimiento del sistema en términos de tiempo y exactitud en sus resultados.
- En un sistema de visión por computador, la generación de trayectorias para un robot cartesiano implica una relación entre el sistema de coordenadas de la cámara y el sistema de coordenadas del robot.

1.4. Análisis de las características de la realidad problemática

- Actualmente existen manuales dedicados a la producción de plantines en viveros artesanales (temporales y permanentes), los cuales resaltan la importancia de la labor manual en todas las etapas de producción, desde la siembra y preparación de sustrato

hasta el repique. En este último proceso se lleva a cabo un control de calidad de acuerdo a la experiencia del trabajador, en el cual mediante inspección visual se desechan plantines pequeños, bifurcadas o defectuosas y enfermas. En viveros de baja producción inclusive el riego es manual y las condiciones climáticas no son controladas (Oliva, Vacalla, Pérez & Tucto, 2014). Por otra parte la producción de plantines de calidad con certificación GLOBALG.A.P PPM por parte de algunos viveros industriales en el Perú (e.g. Agrogénesis), obliga a los productores de material de propagación vegetal a tener buenas en el uso de fertilizantes, condiciones del vivero, plagas y manejo de residuos lo cual conlleva a una semi-automatización del proceso de producción (e.g. riego automatizado por nebulización, control de temperatura) (GLOBALG.A.P, 2020), dejando el proceso de evaluación de calidad del plantín al personal calificado, el cual determinará la calidad del plantín de acuerdo al color, desarrollo foliar, desarrollo radicular, grosor de tallo, presencia de plagas, entre otros factores. Sin embargo, el número elevado de plantines (72 plantines por bandeja) hace del proceso de evaluación de calidad y repique, un proceso lento y costoso.

- La iluminación es uno de mas importantes componentes de los sistemas de visión por computador para la inspección de productos, ya que de esta depende la calidad de las imágenes adquiridas, especialmente para productos con superficies reflectantes (Edan et al. 2009). Patrício y Rieder (2018) mencionan que invertir en mejorar la iluminación incrementaría el performance y confiabilidad del sistema, reduciendo la complejidad del software en la etapa de procesamiento. Este es el caso del proceso de segmentación en el procesamiento de las imágenes, siendo relativamente simple en un entorno controlado y pudiendo ser muy complejo en un entorno de exterior, variante en iluminación y con un fondo complejo. McCarthy, Hancock y Raine (2010) hacen referencia a los sistemas de visión por computador en interior (e.g. laboratorios, fábricas, invernaderos), resaltando que la iluminación controlada y las restricciones en el posicionamiento de los objetos de interés mejoran la confiabilidad en la ubicación y permiten lidiar con la eliminación

del fondo.

- Matuska, Hudec y Benco (2012) dividen a los algoritmos de visión por computador en 3 tipos: los de bajo nivel (operaciones primarias de pre-procesamiento para reducir ruido, mejora de contraste, etc.) caracterizados por tener imágenes como entrada y como salida, los de nivel medio (segmentación, identificación de puntos de interés, descripción, y clasificación) caracterizados por tener imágenes a la entrada y características extraídas a la salida y los de alto nivel buscan dar sentido a los objetos encontrados. Dado que los algoritmos pertenecientes al nivel bajo y medio son necesarios para desarrollar cualquier sistema de visión por computador, cabe analizar la implementación de estos algoritmos en diferentes plataformas de software y hardware. En efecto, Matuska et al. (2012) realizan una comparación entre algoritmos desarrollados en OpenCV y sus versiones en MATLAB. La comparación consiste en analizar el tiempo de uso de CPU en algoritmos de filtrado, operaciones morfológicas, construcción de pirámides de imágenes y detección de bordes. Para utilizar todo el potencial de la computadora en la que se realiza la evaluación, se hace uso de todos los procesadores presentes el CPU (multi-core) y de la capacidad de multithreading. Como resultado se obtuvieron tiempos de uso de CPU hasta 40 veces menores en los algoritmos desarrollados por OpenCV comparados con los de MATLAB. Es preciso resaltar que el uso de la CPU y otros recursos involucrados en la implementación de los algoritmos de visión por computador conformarían la complejidad computacional (Encyclopedia Britannica, 2020), siendo esta última un indicador de la aptitud del algoritmo para su uso en sistemas en tiempo real o que requieran analizar gran cantidad de imágenes en un corto tiempo. Con la finalidad de comparar la ejecución de un mismo tipo de algoritmo en 2 plataformas de hardware diferentes, Poudel y Shirvaikar (2010) evalúan el performance de una computadora personal con un procesador Pentium 4 y una computadora de placa simple BeagleBoard con procesador basado en ARM Cortex-A8. Para esto se ejecutan los algoritmos de convolución y de transformada discreta de Fourier (DFT)

-implementados en OpenCV- en ambas plataformas y se obtiene el tiempo de ejecución, dando como resultado un tiempo 24 veces menor en la computadora con procesador Pentium 4 en el algoritmo de DFT, y un tiempo hasta 11 veces menor en el algoritmo de convolución. De esta manera se pone en evidencia la dependencia de la plataforma de software (librerías, sistema operativo, lenguaje de programación) y la de hardware (CPU, GPU, memoria RAM, etc).

- Para generar trayectorias o indicar la posición (respecto al robot) de un objeto en una imagen capturada por el sistema de visión por computador se debe realizar un proceso de calibración. Tal como indican Rui, Huawei, Yuyang y Jianliang (2017), se deben obtener matrices de transformación que relacionen el plano digital de la imagen (en pixeles) con las coordenadas en milímetros de un robot cartesiano. De igual manera; Lazar, Panescu, Laczko y Braescu (2003) plantean una estrategia para ubicar y orientar el efector final de un brazo robótico de 6 GDL, basado en una matriz homogénea que permite relacionar el sistema de coordenadas de la cámara con el sistema base del robot.

1.5. Formulación del problema

¿Qué técnicas o algoritmos de visión por computador permiten integrar sus resultados a un robot cartesiano controlado por un ordenador de recursos limitados, de manera que sea posible llevar a cabo la tarea de repique de plantines de alcachofa en un entorno de interior?

1.6. Formulación de la hipótesis

Los algoritmos de visión por computador de bajo y mediano nivel, así como el modelo de cámara digital permiten generar las trayectorias de un robot cartesiano durante el proceso de repique de plantines de alcachofa en un entorno de interior.

1.7. Objetivos

1.7.1. Objetivo general

Automatizar el proceso de repique de plantines de alcachofa mediante la integración de un sistema de visión por computador a un robot cartesiano controlado por un ordenador de recursos limitados.

1.7.2. Objetivos específicos

- Definir los parámetros involucrados en la clasificación de un plantín de alcachofa por su calidad.
- Desarrollar y evaluar el performance del sistema de visión por computador para la evaluación de calidad de plantines de alcachofa.
- Generar trayectorias para el efector final del robot cartesiano a partir de las decisiones del sistema de visión por computador.
- Evaluar el performance del sistema robótico durante la actividad de repique de plantines, en términos de tiempo.

1.8. Justificación de la investigación

1.8.1. Importancia de la investigación

En el ámbito de la visión por computador siempre se vienen desarrollando nuevos algoritmos, ya sea para un procesamiento a bajo nivel o a un nivel más abstracto. Es por ello que la presente investigación desarrollará algoritmos que permitan extraer las características morfológicas de plantines de alcachofa a partir de algoritmos de bajo nivel, como el filtrado lineal, segmentación por color, operaciones morfológicas de imágenes binarizadas, entre otros; estableciendo una estructura base para el desarrollo de futuros algoritmos orientados a evaluar la

calidad de plantines en entornos de interior. Por otra parte, se establecerán procedimientos de calibración de cámara basados en la captura de imágenes de patrones conocidos (tablero de ajedrez, círculos) con la finalidad de reducir la distorsión óptica producida en la lente de la cámara y establecer una relación entre el sistema de coordenadas de la cámara y el sistema cartesiano del robot. Cabe resaltar que, si bien hasta la actualidad se han desarrollado sistemas de visión por computador para evaluar la calidad de plantines (ya sea para trasplante o para injerto), estos sistemas presentan heterogeneidad en los parámetros considerados al ejecutar los algoritmos de procesamiento de imágenes (valores de nivel de color, umbrales para la binarización, entre otros). En parte debido a la diferencia de iluminación. Por ello la presente investigación, incluirá una etapa de configuración de los parámetros de la cámara (brillo, contraste, tiempo de exposición, entre otros) y otra de pre-procesamiento de las imágenes obtenidas para reducir el impacto de las variaciones en iluminación. Los sistemas de visión por computador o machine vision han optimizado los procesos de producción en la agroindustria a nivel global, tal como muestran Faridi y Aboonajmi (2017), es por ello que el desarrollo de un sistema de visión por computador para evaluar la calidad de plantines de alcachofa utilizando los criterios de los viveros industriales en la región, permitiría establecer una base para la inclusión de dicho tipo de sistema en la cadena de producción de plantines.

1.8.2. Viabilidad de la investigación

Actualmente se cuenta con acceso a la utilización de un robot de tipo cartesiano marca Farmbot, sobre el cual ya se han desarrollado estudios previos que permiten la generación de movimientos básicos y uso de la plataforma electrónica (Choque, 2018), lo cual permite el desarrollo del sistema de visión por computador y su futura integración a la plataforma robótica. Debido a que la parte electrónica y mecánica del robot ya cuenta con un funcionamiento estable, se precisa solo de un investigador para el desarrollo del sistema de visión por computador. Así mismo no se precisan adquisiciones de hardware (e.g. cámaras, ordenadores, microcontroladores, etc.) ni software. Por otra parte, debido al libre acceso a las páginas de documentación del robot cartesiano y de las librerías de visión por computador disponibles de

manera gratuita (e.g. SimpleCV, OpenCV, numpy, etc.) se aseguran los recursos necesarios para el desarrollo de la investigación.

1.8.3. Limitaciones del estudio

La plataforma electrónica encargada del control de los movimientos del robot, comunicación inalámbrica y otras prestaciones del robot, está conformada por una computadora de placa simple Raspberry Pi 3B y un Arduino Mega 2560 modificado. Debido a la utilización del software (sistema operativo y firmware del microcontrolador) que provee el fabricante, el performance de los algoritmos desarrollados se verán influenciados por el uso de este software.

Capítulo II

MARCO TEÓRICO

2.1. Antecedentes de la investigación

Tangmongkhonsuk et al. (2018) en "Development of a Software Program for Automatic Cartesian Farming Robot", como continuación de un trabajo previo en el que se construyó un robot cartesiano de marca Farmbot, se propuso desarrollar una plataforma de software que permitiera independizar al robot de una conexión a internet. Para implementar dicha plataforma, se empezó por modificar el software propio del robot, específicamente la configuración de la computadora que gobierna las acciones del robot (Raspberry Pi 3) como servidor. De esta manera, un host cliente puede comunicarse con la computadora del robot. Otra modificación (realizada en el lado del cliente), fue el desarrollo de una interfaz gráfica para controlar el robot cartesiano. Para dicha tarea se utilizó el lenguaje de programación Python en conjunto con la librería TkInter. Los resultados se observan en los aspectos de conexión y control del robot, ya que la conexión usuario-robot se encuentra en una red local. Lo cual a su vez permite que los comandos enviados al robot se ejecuten de manera inmediata (con muy baja latencia).

Ali Ashraf et al. (2011) en "Use of Machine Vision to Sort Tomato Seedlings for Grafting Robots", proponen el desarrollo de un algoritmo de visión por computador capaz de determinar la curvatura, nodos de las hojas y diámetro del tallo de plantines para su posterior

clasificación, siendo este proceso la tarea inicial de un robot de injerto. Se empezó por seleccionar los dispositivos para el proceso de adquisición de imágenes. Se eligió una cámara monocromática de 1628x1236 pixeles, una iluminación LED como iluminación frontal, un panel LCD como iluminación posterior y un filtro de luz para reducir el efecto halo. El ordenador en el que se implementaron los algoritmos utiliza un procesador Intel Core2 Quad CPU. En la etapa de adquisición de imágenes se recolectaron 200 vástagos de plantines (tallo y hojas) y 200 rizomas (tallo inferior y raiz) para determinar la distancia óptima entre la cámara y el plantín. El procesamiento de las imágenes no contiene algoritmos complejos que puedan demandar tiempo de cómputo. Utiliza binarización de imágenes en escala de grises y el conteo de pixeles en los ejes vertical y horizontal. Mediante esta técnica se puede analizar el ángulo de inclinación del plantín, el diámetro del tallo y encontrar el punto de corte para el injerto. Los resultados de la investigación muestran que la iluminación posterior permite establecer un umbral constante para la binarización, el filtro de luz mejora ligeramente la adquisición de imágenes. Finalmente, el proceso de clasificación mostró un 97 % de aciertos. La investigación demuestra que es posible desarrollar un clasificador preciso de plantines, a partir de la obtención de las características morfológicas en imágenes monocromáticas, siendo este su principal aporte.

Tong et al. (2012), en "Development of Automatic System for the Seedling Transplanter Based on Machine Vision Technology" propusieron el uso de un sistema de machine vision para: evaluar la calidad de plantines de tomate en una bandeja de almácigo, detectar vacíos en las bandejas y finalmente guiar el efector final de un robot cartesiano. Para el sistema de machine visión, se escogió un cámara CCD TMCC7DSP, una computadora industrial, un digitalizador de video y 6 lámparas fluorescentes. Para la toma de imágenes, la bandeja es transportada a una estructura cerrada con iluminación controlada. El procesamiento de las imágenes consta de una fase de binarización (eliminación de fondo por diferencia de color), análisis de las regiones extraídas (eliminar regiones pequeñas que correspondan a ruido), extracción del área y el perímetro de las regiones para finalmente clasificar a los plantines de

acuerdo a estas características. Se desarrollaron dos métodos de clasificación, uno ejecutado en los primeros 10-20 dias de crecimiento y el otro después de los 20 dias, resultando un porcentaje de aciertos de entre $96.2 - 100\,\%$. El aporte de esta investigación radica en la metodología desarrollada, la cual utiliza segmentación por color y análisis de regiones (área y perímetro).

Liao y Chuang (2009), en "Vegetable Seedling Sorting Based on Mean-Shift", proponen un algoritmo aplicado a la clasificación de plantines, focalizándose principalmente en la etapa de segmentación de las imágenes de color. En efecto, se propone el uso del algoritmo de desplazamiento medio (mean shift) para obtener un patrón de grupos (clusters), correspondientes a los colores dominantes en la imagen. El proceso de clasificación en general, se divide en la etapa de detección de plantines (utilizando la segmentación por color), localización (utilizando análisis de regiones) y la clasificación (basado en el espacio de crecimiento del plantín).La utilización de algoritmos de machine learning, como la segmentación o clústering y el uso del espacio de color CIEluv, para la extracción o detección de plantines, corresponden los principales aportes de esta investigación.

Liu, Xing, Wang, Tian y Jahun (2017) en "Development of machine-vision system for gap inspection of muskmelon grafted seedlings", proponen un sistema de machine-vision aplicado a la inspección de plantines injertados de melón. Se empezó por la selección de dispositivos que conformaron el sistema. La cámara seleccionada es la MER-125-30UC, permitiendo imágenes a color. Adicionalmente se utilizaron lentes de 16 mm y 25 mm y se probaron dos fuentes de luz LED, una blanca y otra verde. Para el desarrollo del algoritmo encargado de inspeccionar los plantines se utilizó el software HALCON 12.0 y su entorno de desarrollo. El algoritmo está basado en la técnica de template-matching en su versión deformable. Para su implementación es necesaria la creación de un modelo (template), lo cual involucra indicar el nivel de pirámide, ángulo y rango de rotación del modelo y el contraste entre el fondo y el modelo. El siguiente

paso consiste en la búsqueda del modelo dentro de una imagen. Los resultados de la búsqueda arrojan valores entre 0 y 1 (donde 0 significa que no hay similitud y 1, igualdad), siendo seleccionado el valor de 0.58 como umbral de calidad. El algoritmo desarrollado permitió analizar 15 plantines por minuto con un porcentaje de aciertos de 98 %. El aporte de esta investigación radica en el uso de técnicas de template-matching en su versión deformable para el análisis de calidad en plantines y el uso del software HALCON 12.0.

2.2. Fundamentación teórica de la investigación

2.2.1. Máquinas CNC (Control Numérico por Computadora)

Las máquinas CNC son robots industriales muy precisas y potentes desarrolladas por John Parsons, IBM y MIT en la década de 1950. La mayoría de estas máquinas usan el lenguaje RS-274D, impuesto por la EIA (Electronics Industry Association). Más conocido como códigos G o códigos G&M, debido a que la mayoría de instrucciones comienzan con esas letras. En efecto, un programa CNC comprende el uso de estos códigos, junto con coordenadas y otros parámetros (Autodesk Inc., 2014). Cabe resaltar que los movimientos de una máquina CNC están basados en el sistema de coordenadas cartesiano. El punto de origen del sistema de coordenadas de la máquina se denomina "home". Es importante que la máquina reconozca su posición de "home" al encenderla debido que desconoce la posición de los ejes en el espacio de trabajo. Bolívar (2012) señala las principales ventajas del uso de máquinas CNC: la disposición de varios lenguajes de programación para la operación de estas máquinas, precisión en sus movimientos, un solo operador puede operar varias máquinas y fácil repetitividad de secuencias de movimiento.

2.2.1.1. Componentes de una máquina CNC

Una máquina CNC se compone principalmente de la arquitectura física (mecánica), el hardware controlador, el software controlador y el software de aplicación.
Según Overby (2011) se pueden identificar dentro de la arquitectura mecánica a:

- Las herramientas montadas en el efector final de la máquina CNC:

  En el mecanizado de piezas se utilizan herramientas para el torneado, fresado y grabado.

- Los sistemas de guía:

  En máquinas CNC se utilizan guías lineales, las cuales deben permitir: un movimiento rectilíneo a lo largo de un determinado eje, movimiento suave con una mínima fricción y orientación rígida ortogonal entre los ejes. Entre las principales guías lineales se tiene a: los rodamientos lineales de bolas (Figura 1a), los rieles perfilados (Figura 1b), rodillos en V (Figura 1c), entre otros.

- Los sistemas de transmisión:

  Según Overby (2011, pp. 35), los sistemas de transmisión en una máquina CNC se encargan de transformar la potencia rotacional del motor en movimientos lineales. Entre los principales sistemas de transmisión se encuentran: el sistema de husillo de bolas (Figura 2a), husillo de rosca trapezoidal (Figura 2b), tuerca rodante (Figura 2c), piñón-cremallera (Figura 2d) y correa de distribución-polea. Los diferentes sistemas de transmisión presentan diferentes características de velocidad, torque, vibración, precisión y capacidad de carga, lo que permite seleccionar el sistema más adecuado para una determinada aplicación.

- Motores:

  De acuerdo a Overby (2011, pp. 57), se identifican dos tipos de motores usados en maquinaria CNC: los servomotores y los steppers (paso a paso). Los motores paso a paso son los más usados en aplicaciones industriales de rango medio. Se clasifican de acuerdo a su tamaño en NEMA 17, NEMA 23, NEMA 34 y NEMA 42, pudiendo utilizarse en modo unipolar o bipolar. Normalmente estos motores se utilizan sin encoder utilizando un control a lazo abierto, sin embargo, en ocasiones es necesario utilizar un encoder para detectar pérdida de pasos. A pesar de la facilidad de uso de estos motores, se deben utilizar dentro de un rango de velocidad, debido a que pasado ese rango el torque del motor empieza a disminuir. Por otro lado, los servomotores (tanto DC como AC), si

bien son generalmente más costosos y requieren el uso de un encoder para el lazo de control realimentado, desarrollan más potencia cuando operan a altas velocidades y tienen tiempos de respuestas más cortos (mayor aceleración y desaceleración).

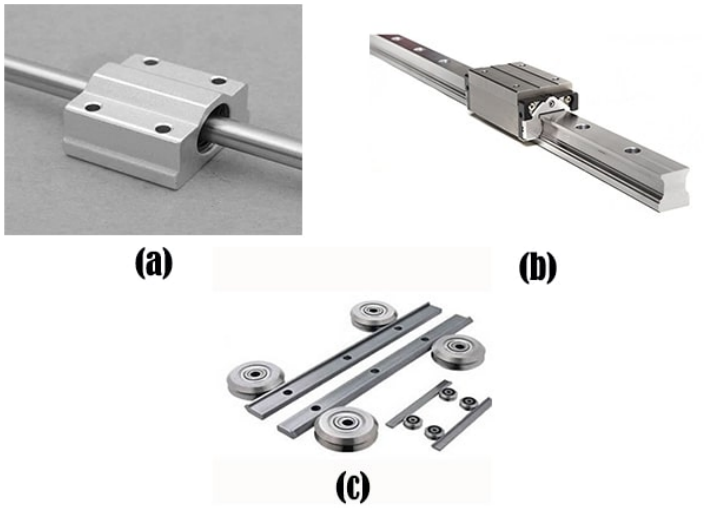


Figura 1: *Guías lineales más utilizadas en máquinas CNC: (a) Rodamientos lineales de bolas; (b) Rieles perfilados; (c) Rodillos en V. Fuente: Elaboración propia*

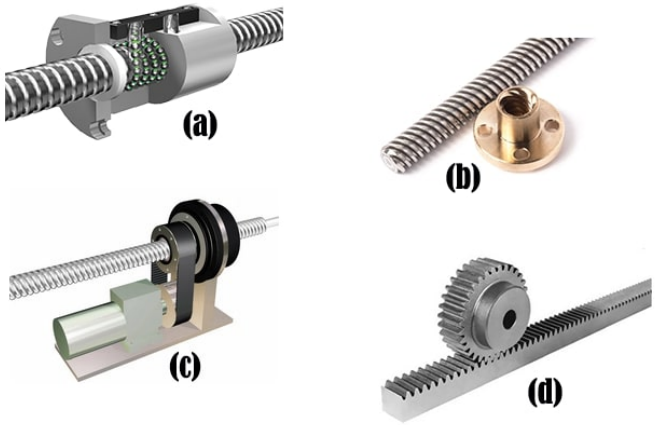


Figura 2: *Sistemas de transmisión más utilizados en máquinas CNC: (a) Husillo de bolas; (b) Husillo de rosca trapezoidal; (c) Tuerca rodante; (d) Piñón-cremallera. Fuente: Elaboración propia*

Dentro de los componentes electrónicos se encuentran:

- Hardware controlador de máquinas CNC:

  Se compone principalmente de una placa de expansión (*breakout board*), la cual le permite acceder a las señales provenientes de una computadora proveyendo además aislamiento galvánico; y los drivers a los que van conectados los motores, ya sean steppers o servomotores. Adicionalmente existen componentes que mejoran el funcionamiento de una máquina CNC, como es el caso de los micro-switches (para la determinación de límites en los ejes y el proceso de "homing").

- Software controlador de máquinas CNC:

  Existen numerosas opciones para el control de una máquina CNC, pero en general la mayoría de softwares utilizan el estándar RD-274 o códigos g para enviar órdenes a la máquina. Los softwares de control pueden ser gratuitos o requerir un pago, así mismo algunos pueden operar diversas máquinas y otros pueden ser cerrados para una sola plataforma. En general, la mayoría de software de control CNC viene integrado dentro de un software de CAD-CAM (e.g. SolidWorks, Fusion 360, etc).

2.2.1.2. Esquema básico de un programa CNC

Según Autodesk Inc. (2014, pp. 59), un programa CNC está compuesto por una lista de instrucciones, las cuales deben ejecutarse en el orden en el que están escritas. Las instrucciones se leen de izquierda a derecha y de arriba a abajo.

Las instrucciones y códigos varían entre las diferentes máquinas CNC. En el caso del robot cartesiano Farmbot Génesis v1.3, los códigos g y f son los que permiten generar movimiento y ejercer operaciones sobre los pines del microcontrolador (Tabla 1).

Block	Description	Purpose
%	Start of program.	Start Program
O0001 (PROJECT1)	Program number (Program Name).	
(T1 0.25 END MILL)	Tool description for operator.	
N1 G17 G20 G40 G49 G80 G90	Safety block to ensure machine is in safe mode.	
N2 T1 M6	Load Tool #1.	Change Tool
N3 S9200 M3	Spindle Speed 9200 RPM, On CW.	
N4 G54	Use Fixture Offset #1.	Move To Position
N5 M8	Coolant On.	
N6 G00 X-0.025 Y-0.275	Rapid above part.	
N7 G43 Z1. H1	Rapid to safe plane, use Tool Length Offset #1.	
N8 Z0.1	Rapid to feed plane.	
N9 G01 Z-0.1 F18.	Line move to cutting depth at 18 IPM.	
N10 G41 Y0.1 D1 F36.	CDC Left, Lead in line, Dia. Offset #1, 36 IPM.	Machine Contour
N11 Y2.025	Line move.	
N12 X2.025	Line move.	
N13 Y-0.025	Line move.	
N14 X-0.025	Line move.	
N15 G40 X-0.4	Turn CDC off with lead-out move.	
N16 G00 Z1.	Rapid to safe plane.	
N17 M5	Spindle Off.	Change Tool
N18 M9	Coolant Off.	
(T2 0.25 DRILL)	Tool description for operator.	
N19 T2 M6	Load Tool #2.	
N20 S3820 M3	Spindle Speed 3820 RPM, On CW.	

Figura 3: *Formato general de un programa CNC. Fuente:Autodesk Inc. (2014, pp. 60)*

2.2.2. Composición mecánica del robot cartesiano

El movimiento del robot es posible mediante el uso de perfiles de aluminio V-Slot, placas de aluminio, correas dentadas (movimientos en X y Y) y husillo de rosca trapezoidal (movimiento en eje Z). Los motores encargados de generar movimiento en los 3 ejes son del tipo paso a paso (NEMA 17) con encoders rotatorios.

La estructura mecánica del robot (Figura 4) se divide en: vías (tracks), pórtico (*gantry*), carro transversal (*cross-slide*) y eje Z. El gantry puede moverse a lo largo de las vías, generando un movimiento en el eje X. El cross-slide puede moverse a lo largo del gantry, generando un movimiento en el eje Y. Y finalmente es posible la generación de movimiento en el eje Z, mediante el motor montado sobre el cross-slide.

Tipo de código	Número	Parámetros	Función
G	00	X Y Z A B C	Movimiento absoluto a una determinada velocidad
G	28		Movimiento de homing en todos los ejes
F	11		Homing en el eje X
F	12		Homing en el eje Y
F	13		Homing en el eje Z
F	21	P	Leer parámetro
F	22	P V	Modificar parámetro
F	22	P V	Modificar parámetro
F	41	P V M	Modificar valor de un pin
F	42	P M	Leer valor de un pin
F	84	X Y Z	Setear posición actual de eje como cero

Tabla 1: *Códigos g y f principales utilizados para operar el robot cartesiano Farmbot. Fuente: Farmbot Inc. (2017)*

El robot tiene un porta-herramientas (UTM: Universal Tool Mount, Figura 5) en el lugar del efector final, permitiendo montar las herramientas propias del fabricante (boquilla de regado, medidor de humedad, removedor de hierba) y otras herramientas desarrolladas, todo esto mediante acoplamiento magnético. Adicionalmente, a un lado del UTM, se encuentra montada una cámara boroscópica (conectada vía USB a la SBC Raspberry Pi 3B), permitiendo la captura de imágenes para su posterior procesamiento y análisis.

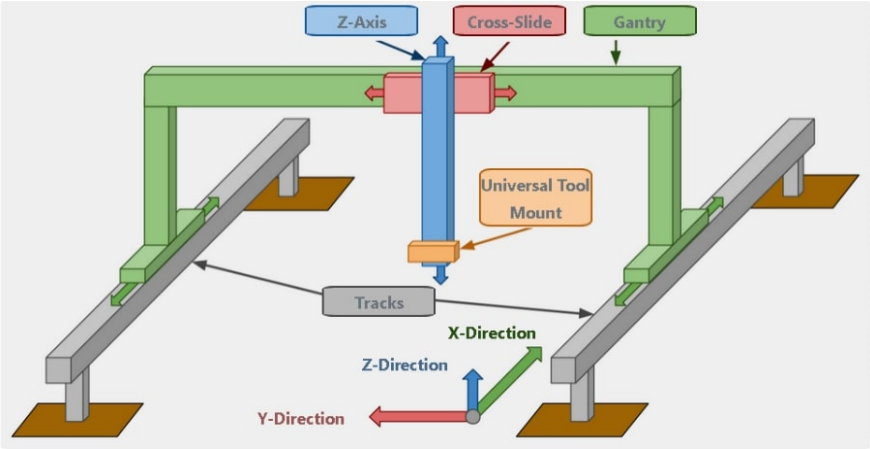


Figura 4: *Elementos de la estructura mecánica del robot. Fuente: Farmbot Inc. (2017)*

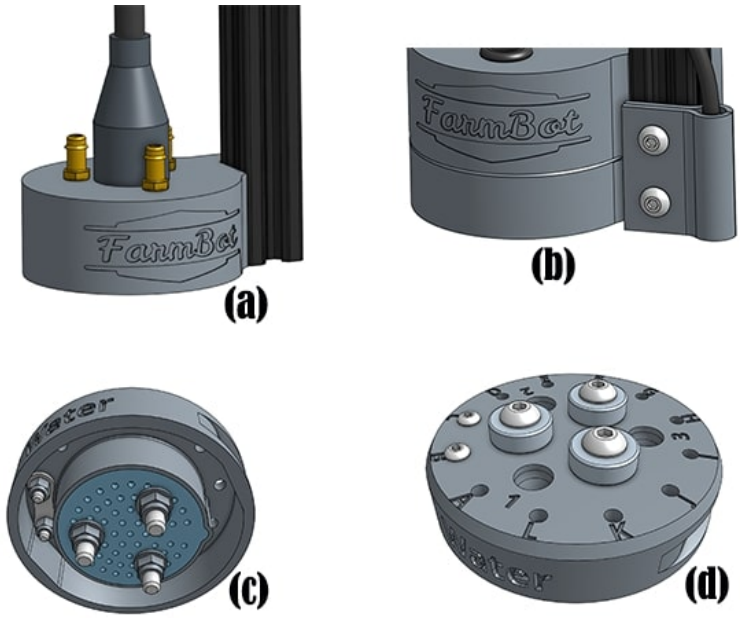


Figura 5: *Elementos del efector final: (a) UTM montado; (b) Cámara montada en el UTM; (c) Boquilla de regado; (d) Disposición de imanes para acoplamiento. Fuente: Farmbot Inc. (2017)*

2.2.3. Composición electrónica del robot cartesiano

La plataforma electrónica del robot, la conforman un Arduino Mega 2560 y un Raspberry Pi 3B, encargados de controlar los movimientos del robot, entre otras funciones.

2.2.3.1. Microcontrolador y firmware

La plataforma basada en microcontrolador está compuesta por un Arduino Mega 2560 y una shield RAMPS (RepRap Arduino Mega Pololu Shield). El firmware está escrito en C++. Se encarga principalmente de interpretar y ejecutar códigos g (propios del control numérico), operar sobre los pines digitales y analógicos del Arduino Mega, entre otros.

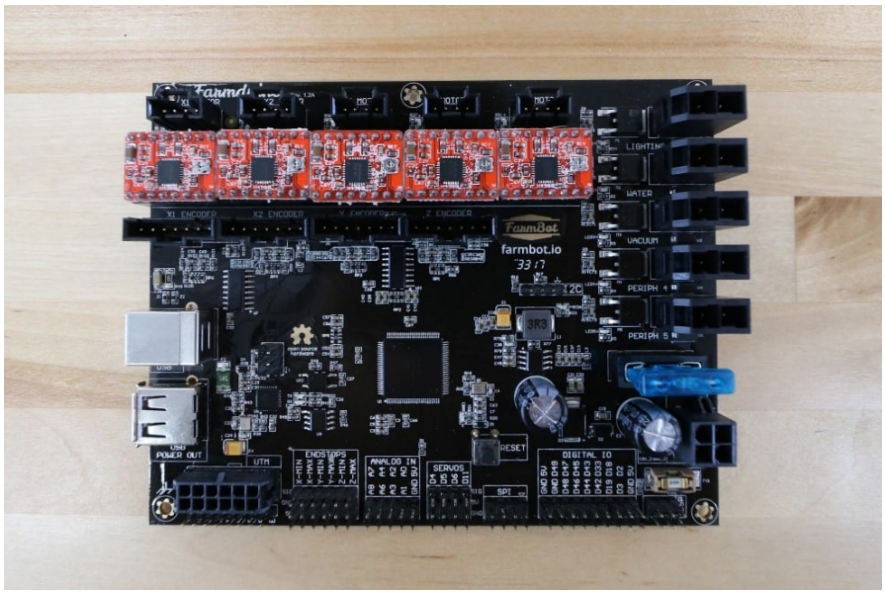


Figura 6: *Placa basada en microcontrolador Farmduino v1.3. Fuente: Farmbot Inc. (2017)*

2.2.3.2. Computadora de placa simple (SBC) y sistema operativo

Según Paunski y Angelov (2019), una computadora de placa simple es una computadora completa implementada en una sola placa de circuito impreso (PCB), incluyendo el microprocesador, memoria, controladores de dispositivos de entrada/salida y otras prestaciones requeridas para una computadora funcional. Entre sus principales ventajas se encuentran: el performance de la CPU y GPU, el tamaño reducido y conexión mediante periféricos (I2C, SPI, UART, etc.).

La SBC utilizada en el robot Farmbot es del tipo Raspberry Pi 3B. El uso de la SBC Raspberry Pi 3B, le permite al sistema robótico la conexión con la aplicación web mediante HTTP y AMQP. De igual manera, permite la ejecución de complejas secuencias que superan las prestaciones ofrecidas por la plataforma del microcontrolador (Arduino Mega 2560), siendo una de ellas el procesamiento de imágenes.

Característica	**Descripción**
SoC	Broadcom BCM2837
CPU	4 x ARM Cortex-A53, 1.2GHz
GPU	Broadcom VideoCore IV
RAM	1GB LPDDR2 (900 MHz)
Conectividad	10/100 Ethernet, 2.4GHz 802.11n wireless
Bluetooth	Bluetooth 4.1 Classic, Bluetooth Low Energy
Almacenamiento	MicroSD
GPIO	Header de 40 pines
Puertos	HDMI, 3.5mm analogue audio-video jack, 4 x USB 2.0, Ethernet, CSI,DSI

Tabla 2: *Especificaciones de la SBC Raspberry Pi 3B*

Debido a la limitación de recursos (principalmente el uso de CPU y tamaño de memoria), es preciso emplear un sistema operativo que no haga uso intensivo de estos. En efecto, el sistema operativo ofrecido por el fabricante está diseñado precisamente para darle las prestaciones

necesarias y suficientes al sistema robótico, permitiendo un uso eficiente de los recursos.

Para el desarrollo del sistema operativo Farmbot OS, se utilizó la estructura Nerves, la cual permite desarrollar sistemas operativos para sistemas embebidos utilizando el lenguaje Elixir. Para el desarrollo de sistemas operativos en esta estructura, es preciso determinar tres factores: plataforma (el hardware donde se ejecutará el sistema operativo), la estructura (librerías en forma de módulos de Elixir) y herramientas (para configurar dispositivos) (Nerves Project, 2020).

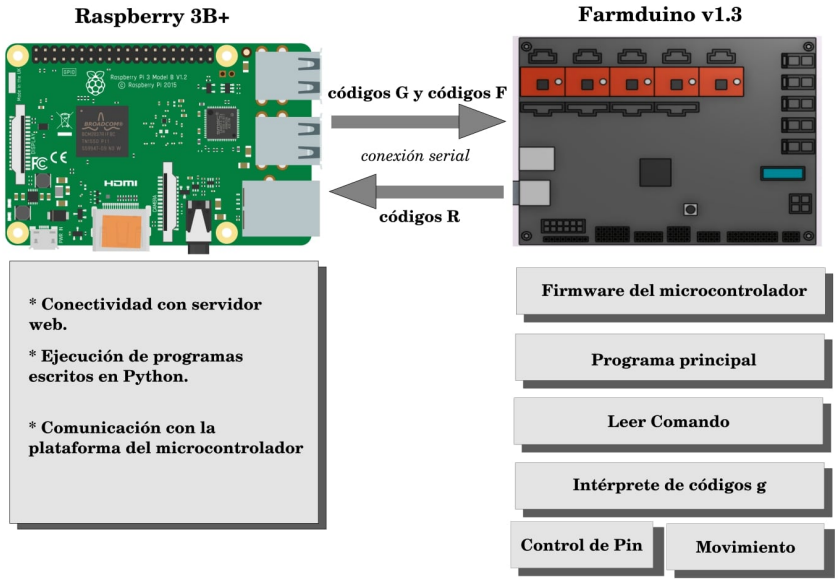


Figura 7: *Comunicación entre Raspberry Pi 3B y el microcontrolador del Farmbot. Fuente: Elaboración propia*

En Figura 7, se observa la estructura de la comunicación entre la computadora del robot y el microcontrolador, indicando que los códigos g (encargados de los movimientos absolutos) y los códigos f (encargados de procesos de homing, calibración y control de los pines adicionales) se envían desde Raspberry Pi 3B hacia el microcontrolador, siendo este último el encargado de enviar los códigos r (encargados de informar los reportes de los comandos ejecutados).

2.2.4. Software de aplicación del robot cartesiano

Para operar el robot cartesiano es preciso contar con un servidor (integrado por una REST API, interfaz de usuario y un *message broker*) y un navegador web instalado en una computadora o un Smartphone.

Si bien la aplicación web (Farmbot Web App) que compone la interfaz de usuario permite una operación gráfica, sencilla e intuitiva, es poco útil para el desarrollo de complejas secuencias de movimiento o generación de trayectorias basadas en algoritmos de visión por computador. A pesar de esto, el uso de la aplicación es útil para configurar algunos parámetros del robot (aceleración, velocidad mínima y máxima, posición cero) e instalar aplicaciones (Farmwares) escritas en lenguaje Python. De igual manera, el message broker integrado en el servidor es indispensable para los procesos de autenticación, identificación de dispositivos y mensajería usuario-robot en tiempo real.

2.2.4.1. Desarrollo de aplicaciones (*Farmwares*)

La aplicación web permite la instalación de proyectos escritos en Python3 y alojados en la plataforma GitHub, a modo de repositorio.

Los módulos o librerías disponibles para el desarrollo de aplicaciones son limitados, en otras palabras, solo se disponen de los módulos propios de Python3, la librería numpy y la librería opencv-python. Adicionalmente, el fabricante pone a disposición la librería *farmware-tools*, la cual permite tener un control sobre las acciones del robot desde una perspectiva de alto nivel. En efecto, esta librería permite al usuario una comunicación con el robot en tiempo real, vía nodos de CeleryScript. Estos nodos, indican la acción a realizar y los parámetros o argumentos de dicha acción.

Lo beneficioso de utilizar la librería *farmware-tools*, es que la construcción y envío de dichos nodos vía MQTT se simplifica, permitiendo al usuario enfocarse en el desarrollo de aplicaciones para generar trayectorias complejas de acuerdo a la información obtenida por los algoritmos de visión por computador.

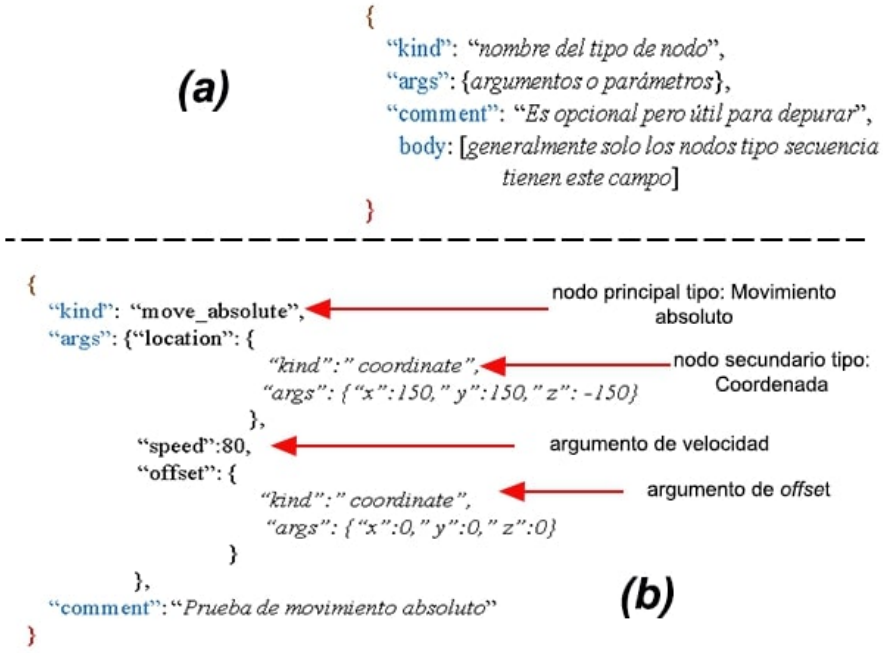


Figura 8: *Notación para envío de órdenes vía CeleryScript nodes: (a) Formato general; (b) Ejemplo de un movimiento absoluto. Fuente: Elaboración Propia*

Adicionalmente, cabe resaltar que la generación de códigos g y f, así como el manejo de códigos r, son operaciones de bajo de nivel para ser administradas por el sistema operativo del robot. El usuario no necesita generar dichos códigos para operar el robot.

En la Figura 8b, se puede visualizar un ejemplo del envío de la orden de movimiento absoluto a las coordenadas X: 150 mm, Y: 150 mm, Z: -150 mm, la cual es enviada usando el protocolo MQTT hacia el Message Broker del servidor, el cual lo publica (publish) y finalmente el robot interpreta la orden mediante un proceso que se ejecuta en segundo plano dentro del sistema operativo del mismo y ejecuta la orden. Cabe recalcar el uso del argumento de velocidad (speed), el cual indica el porcentaje de la velocidad máxima a la que se deben ejecutar los movimientos y el uso del offset para indicar desplazamientos relativos adicionales.

2.2.5. Visión por computador en la evaluación de plantines

2.2.5.1. Adquisición de imágenes digitales

Una imagen digital puede ser definida como una función bidimensional $I(u, v)$, donde u y v son coordenadas espaciales y la amplitud de I es llamada intensidad, siendo todas estas variables discretas y finitas (Gonzalez & Woods, 2001, pp. 1). Además, la intensidad se forma como una combinación de "iluminación" (fuente) y "reflexión" (objeto).

Para la obtención de imágenes digitales en 2D se pueden utilizar sensores de luz dispuestos en arreglos lineales (usados en los escáneres) o arreglos matriciales (usados en las cámaras digitales). El modelo de cámara digital (Figura 9) describe las transformaciones que "sufre" un objeto en 3D para ser proyectado en una imagen digital. En efecto Alegre, Pajares y De la Escalera (2016, pp. 233) describen a la matriz de proyección perspectiva $\mathbf{M}$, la cual relaciona la posición de un punto 3D en un sistema de coordenadas $\mathbf{W}$ con un punto en el plano imagen discretizado $\mathbf{D}$.

$$\mathbf{M} = {}^{d}\mathbf{K}_{p}\ {}^{p}\mathbf{P}_{c}\ {}^{c}\mathbf{T}_{w} \tag{1}$$

En (1) se muestran índices: d, p, c, w :

- d es el sistema de referencia asociado al plano imagen discretizado.
- p es el sistema de referencia asociado al plano imagen continuo.
- c es el sistema de referencia de la cámara.
- w es el sistema de referencia del mundo en 3D.

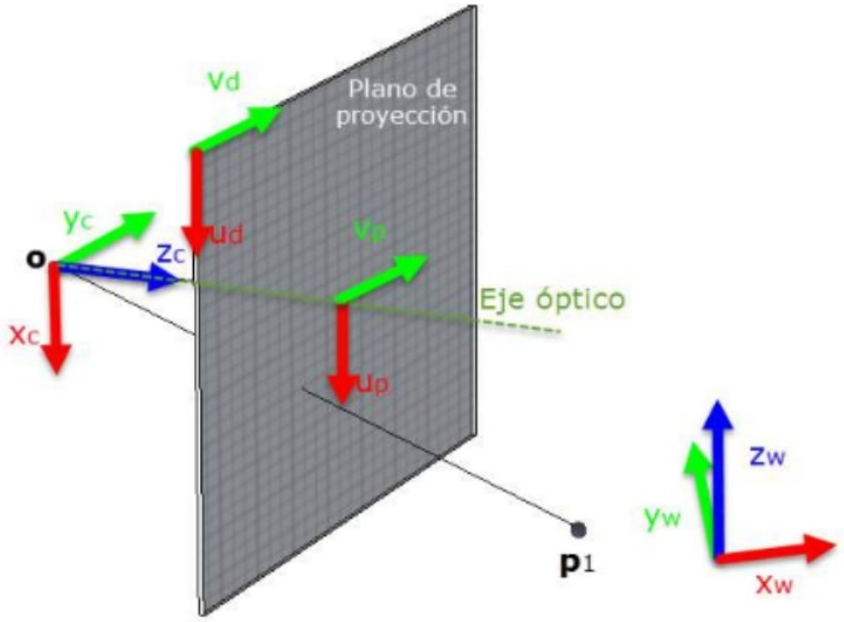


Figura 9: *Localización de los sistemas de referencia del modelo de cámara. Fuente: Alegre, Pajares y De la Escalera (2016, pp. 234)*

En la Figura 10, se pueden observar las relaciones geométricas existentes en el proceso de proyección de un punto 3D. Donde f representa la distancia focal, x_i representa la posición en el plano imagen y X representa la posición del punto 3D en el sistema de referencia de la cámara. De la Figura 10, se pueden derivar las ecuaciones (2) y (3), que describen el proceso de proyección en términos de coordenadas, evidenciando la pérdida de una dimensión: z.

$$\frac{x_i}{f} = \frac{x}{z} \tag{2}$$

$$\frac{y_i}{f} = \frac{y}{z} \tag{3}$$

Si bien el proceso de proyección describe muy bien como un punto en 3D es convertido en un punto en 2D en el plano imagen, aún restan los procesos de muestreo y cuantización que se encargan de transformar una imagen continua en una imagen digital.

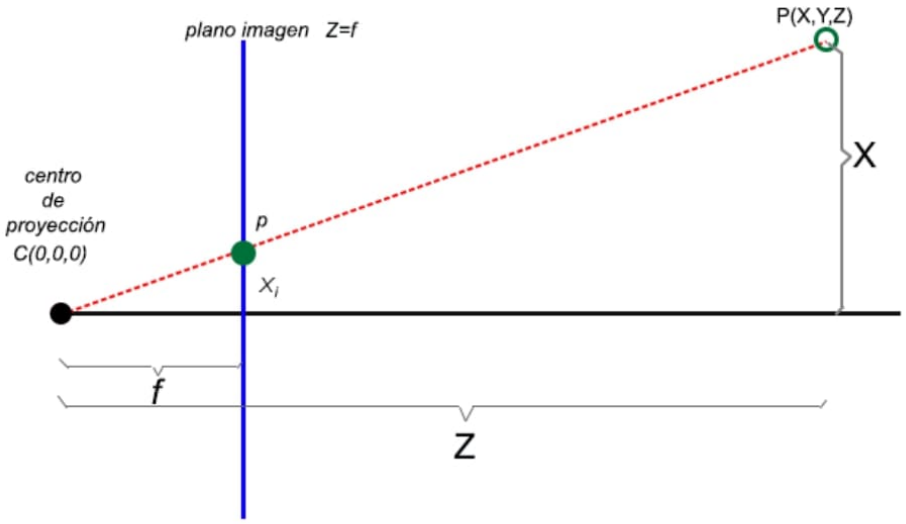


Figura 10: *Relaciones geométricas presentes en el proceso de proyección del modelo de cámara. Fuente: Elaboración Propia*

Según Gonzalez y Woods (2001, pp. 52), el muestreo es la digitalización de los valores de las coordenadas (u, v), mientras que la cuantización es la digitalización de la amplitud o intensidad de $I(u, v)$. Ambas digitalizaciones afectan la calidad de la imagen: la cantidad de filas y columnas, y el número de niveles discretos de intensidad (la profundidad de bit). Los fabricantes de cámaras digitales o sensores de imagen ofrecen variadas resoluciones de imagen (cantidad de filas y columnas): 1280×720, 1200×800, 640×480, 320×240, etc. Sin embargo, no hay mucha variedad en la profundidad de bit. En efecto la mayoría de fabricantes ofrecen una profundidad de 8-bit (256 valores de intensidad). En la Fig. 11 se observa el efecto producido por los procesos de muestreo y cuantización de una imagen continua.

2.2.5.2. Pre-procesamiento de imágenes

Gonzalez y Woods (2001, pp. 75) describen el pre-procesamiento o mejoramiento de la imagen como el proceso de modificar la imagen de modo que el resultado sea más adecuado que la imagen original para una aplicación específica (orientado a un problema). En ese sentido, existen numerosas técnicas para mejorar el contenido de una imagen: transformaciones

de intensidad, procesamiento de histograma, uso de operaciones aritmético-lógicas, filtrado en el dominio del espacio, filtrado en el dominio de la frecuencia, entre otras.

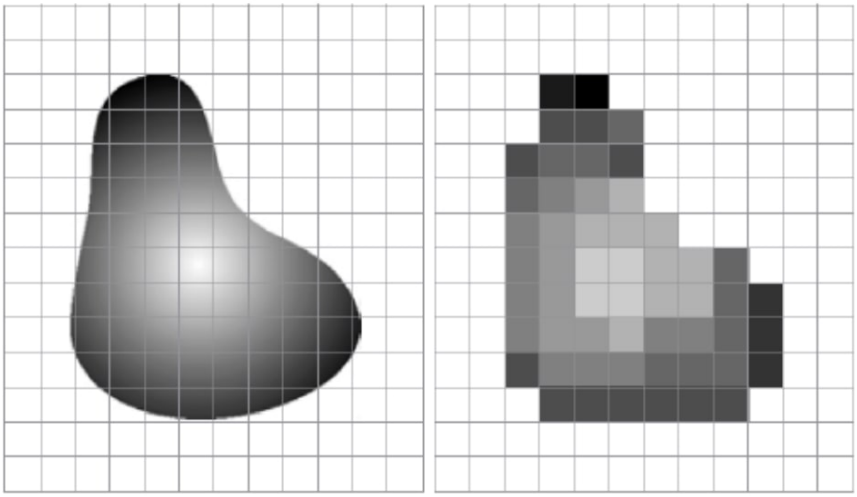


Figura 11: *Procesos de muestreo y cuantización de una imagen continua. Fuente: Gonzalez y Woods (2001, pp. 54)*

- Transformaciones de intensidad:

  Denotando a $r = I_S(u, v)$ como la imagen original y a $s = I_T(u, v)$ como la imagen transformada, se puede definir una transformación de intensidad como una operación que se realiza a cada pixel de manera individual, tal como se muestra en la ecuación (4).

$$s = T(r) \tag{4}$$

  Las transformaciones de intensidad más conocidas son las: Imágenes negativas, logarítmicas, potenciales (modificación de gamma) y lineales por tramos.

  En el caso de las imágenes negativas, tal como se describe en Gonzalez y Woods (2001, pp. 78), una imagen con L intensidades o en el rango de $[0, L-1]$ se puede invertir de acuerdo a la ecuación (5).

$$s = L - 1 - r \tag{5}$$

Las transformaciones logarítmicas se utilizan para ensanchar o comprimir el histograma de los niveles de intensidad en una imagen y su fórmula general se muestra en la ecuación (6).

$$s = clog(1 + r) \tag{6}$$

Las transformaciones potenciales, se utilizan principalmente para corregir variaciones de intensidad entre dispositivos de imagen: cámaras, impresoras, monitores. Su fórmula general se muestra en (7).

$$s = cr^{\gamma} \tag{7}$$

Finalmente, las transformaciones lineales por tramos permiten modificar ciertos rangos de valores de intensidad de manera diferenciada.

Los efectos visuales generados sobre las imágenes al aplicar transformaciones de intensidad, son principalmente la modificación de brillo y contraste.
En la Fig. 12 se muestran las gráficas de las funciones correspondientes a las transformaciones de intensidad. Adicionalmente, en la Fig. 13 se observan los efectos visuales producidos por las transformaciones mencionadas.

- Procesamiento de histograma:

  Según Gonzalez y Woods (2001, pp. 88), el histograma de una imagen digital con niveles $[0, L-1]$ es una función discreta, tal como se muestra en (8), donde r_k es el k-ésimo nivel de intensidad y n_k es el número de pixeles en la imagen que tiene el nivel r_k.

$$h(r_k) = n_k \tag{8}$$

$$p(r_k) = \frac{n_k}{\eta} \tag{9}$$

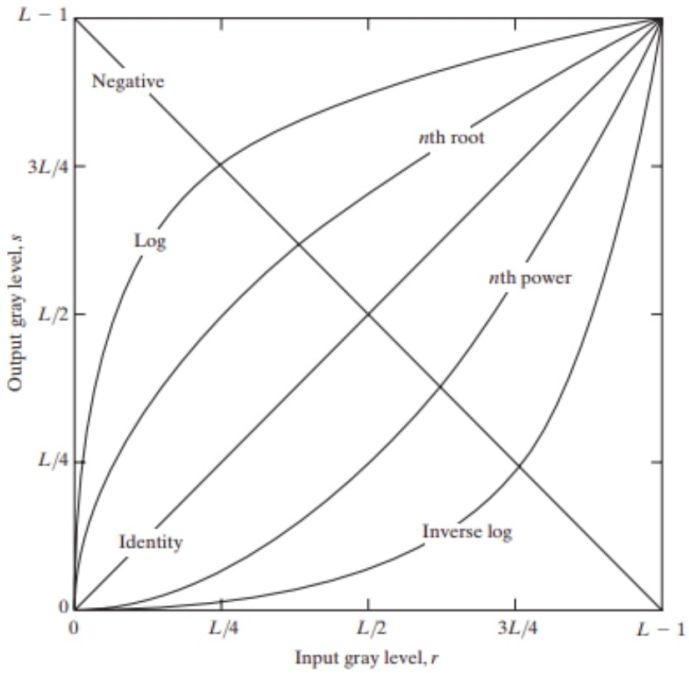


Figura 12: *Gráficas de funciones de transformaciones de intensidad.Fuente: Gonzalez y Woods (2001, pp. 78)*

En (9), se obtiene el histograma normalizado con un rango de $[0, 1]$, siendo η la cantidad total de pixeles. El procesamiento de histograma involucra la modificación del histograma de una imagen mediante la modificación de las intensidades de los pixeles.

En ese sentido el algoritmo de ecualización de histograma busca encontrar una función de transformación de intensidad que permita un obtener un histograma con una distribución homogénea. Esto a su vez permite corregir zonas con mucha intensidad o zonas con poca intensidad.

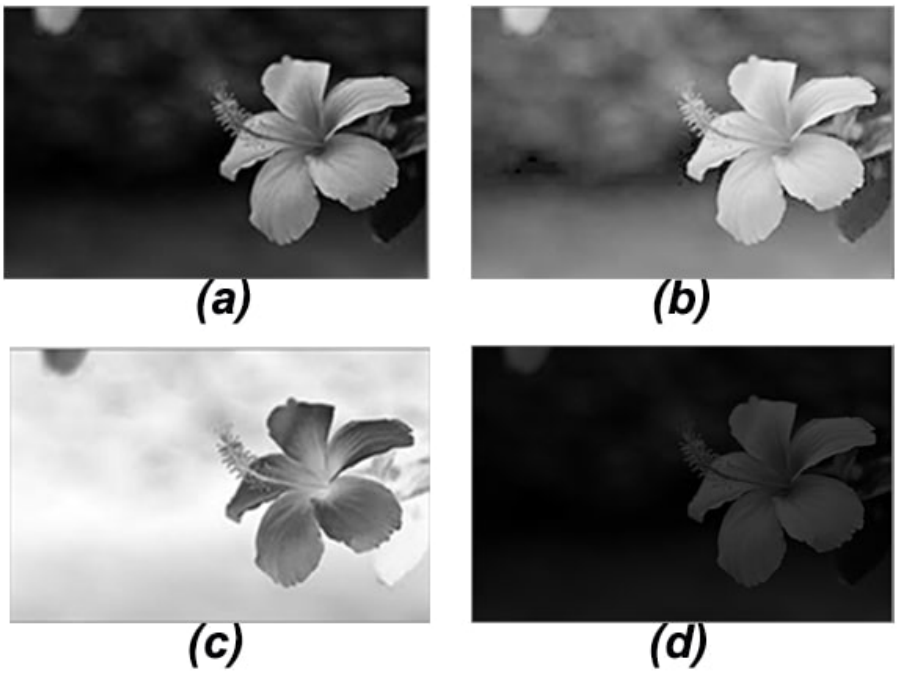


Figura 13: *Transformaciones básicas de intensidad: (a) Imagen original; (b) Transformación potencial* $\gamma = 0.4$*; (c) Inversión de imagen; (d) Transformación logarítmica. Fuente: Elaboración propia*

En la Figura 14 se observa la idealización del algoritmo de ecualización de histograma. La función de transformación de intensidad, involucra analizar el histograma como la función de densidad de probabilidad (PDF) de una variable aleatoria, siendo P_r la PDF del histograma original y Ps, la PDF del histograma ecualizado. La ecuación (10) es el resultado del análisis. Evidenciando que cada valor de la intensidad de la imagen original (r_k) debe ser reemplazado por otro valor específico (s_k), pudiendo utilizarse *look-up tables* para esta tarea.

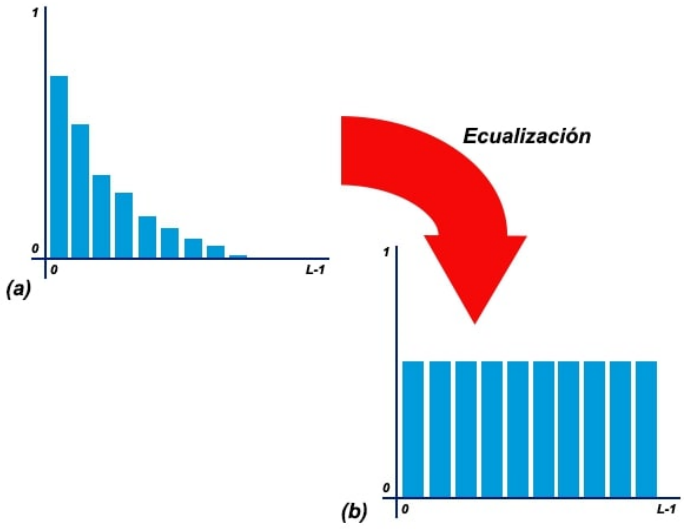


Figura 14: *Resultado ideal de la ecualización de histograma: (a) Histograma original; (b) Histograma ecualizado. Fuente: Elaboración propia*

$$S_k = T(r_k) = \sum_{j=0}^{k} P_r(r_j) \tag{10}$$

- Filtrado en el dominio del espacio:

  En Gonzalez y Woods (2001, pp. 116) se describe al filtrado en el dominio del espacio como la operación de un kernel o filtro sobre la vecindad de los pixeles que conforman la imagen. En el caso de los filtros espaciales lineales, la respuesta a un filtro está determinado por la suma de los productos de los coeficientes del filtro y los correspondientes pixeles en el área abarcada por el filtro. El filtrado lineal corresponde con la operación de convolución, por ello se puede hablar de convolucionar un kernel sobre una imagen. En la Fig. 15 puede observarse un ejemplo de filtrado del tipo promediador con dimensiones 3×3. Es importante resaltar que la operación de filtrado espacial tiene un efecto directo sobre el dominio de la frecuencia de la imagen. Sin embargo, en procesamiento de imágenes se denominan filtros de suavizado a los filtros pasa-bajos y filtros del tipo sharpening a los filtros pasa-altos. Los filtros de suavizado más conocidos son: los promediadores y los de promedio ponderado. Por otro lado, los filtros de sharpening más conocidos son: el laplaciano y el de gradiente. Cabe resaltar que, para el mejoramiento

de una imagen en el dominio del espacio, pueden realizarse operaciones aritméticas a las imágenes después de haberse filtrado, como es el caso del: Unsharp masking y High-boost filtering (Gonzalez & Woods, 2001, pp. 132).

- Filtrado en el dominio de la frecuencia:

  El concepto de frecuencia tiene poco sentido al aplicarse a imágenes, es por ello que se relaciona la frecuencia con la "velocidad de cambio" o patrones de cambio. En ese sentido, si hay pocos detalles (imagen plana) se habla de cambios suaves o bajas frecuencias. En cambio, si hay cantidad de detalles (imagen con textura) se habla de cambios rápidos o altas frecuencias (Gonzalez & Woods, 2001, pp. 160). El filtrado de una imagen en el dominio de la frecuencia, consiste en modificar los coeficientes de la 2D-DFT (transformada discreta de Fourier en 2D) de una imagen. Para esto es preciso primero obtener la DFT de la imagen y luego multiplicarla elemento-por-elemento con una función de filtro. En las ecuaciones (11) y (12) se definen la DFT en dos dimensiones de una imagen $I(u,v)$ (con dimensiones $M \times N$) y su inversa.

$$\Gamma(s,t) \;=\; \frac{1}{MN}\sum_{u=0}^{M-1}\sum_{v=0}^{N-1} f(u,v)e^{-j2\pi\left(\frac{su}{M}+\frac{tv}{N}\right)} \tag{11}$$

$$I(u,v) \;=\; \sum_{s=0}^{M-1}\sum_{t=0}^{N-1} F(s,t)e^{-j2\pi\left(\frac{su}{M}+\frac{tv}{N}\right)} \tag{12}$$

  En (13) se define la operación de filtrado, donde $H(s,t)$ es la función de filtro y $\Gamma(s,t)$ es la DFT de la imagen. Finalmente, para obtener la imagen filtrada, se transforma $G(s,t)$ al dominio del espacio mediante la inversa de la DFT.

$$G(s,t) \;=\; H(s,t)\Gamma(s,t) \tag{13}$$

  Los filtros en el dominio de la frecuencia más conocidos son: Filtro ideal pasa-bajos,

filtro Butterworth pasa-bajos, filtro Gaussiano pasa-bajos y sus respectivas versiones pasa-altos.

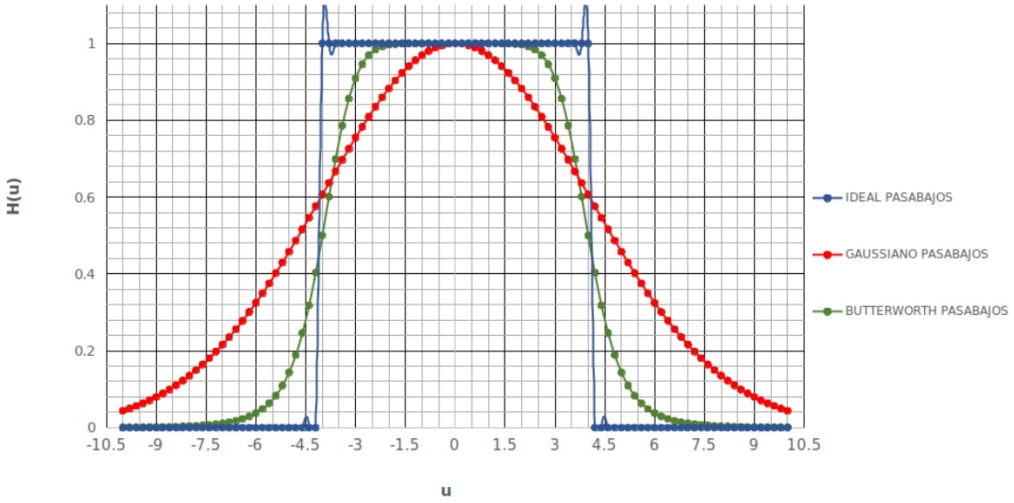


Figura 15: *Respuestas de filtros pasabajos en una dimensión. Fuente: Elaboración propia*

- Filtrado Homomórfico:

  De acuerdo a Gonzalez y Woods (2001, pp.50), Myler, Weeks y Voicu (1995) y Eddins (2013), la intensidad luminosa $I(u, v)$ –registrada en una imagen digital– de un objeto, es el producto de la iluminación en la escena por la reflectancia del objeto, tal como se define en (14), donde $L(u, v)$ representa la iluminación de la escena en el pixel $[u, v]$ y $R(u, v)$ representa la reflectancia del objeto en el pixel $[u, v]$. Debido a que la reflectancia de los objetos tiende a variar de manera brusca dentro de las imágenes, mientras que la iluminación de la escena varía lentamente o es casi constante (Eddins, 2013), se podría solucionar el problema de las variaciones de iluminación entre imágenes si se lograra separar ambos componentes (i.e. eliminar $L(u, v)$ de la imagen).

$$I(u, v) = L(u, v)R(u, v) \tag{14}$$

  Para realizar esta separación es que se aplica logaritmo natural a la imagen, de acuerdo a (15). Luego de aplicado el logaritmo natural, puede notarse que ambos componentes han sido separados.

$$ln\big(I(u,v)\big) = ln\big(L(u,v)\big) \ + ln\big(R(u,v)\big) \tag{15}$$

Como ya se mencionó anteriormente, la iluminación de la escena presenta principalmente componentes de baja frecuencia, mientras que la reflectancia está compuesta por componentes de alta frecuencia en su mayoría. Por lo tanto, se podría realizar un filtrado en el dominio de la frecuencia, para obtener solamente el componente $ln\big(R(u,v)\big)$. Una vez filtrada la imagen $ln\big(I(u,v)\big)$, se realizaría la operación inversa al logaritmo (exponenciación), para obtener la imagen final filtrada. El esquema general del procedimiento para llevar a cabo el filtrado homomórfico es mostrado en la Figura 16, donde $H(s,t)$ representa el filtro paso alto en el dominio de la frecuencia.

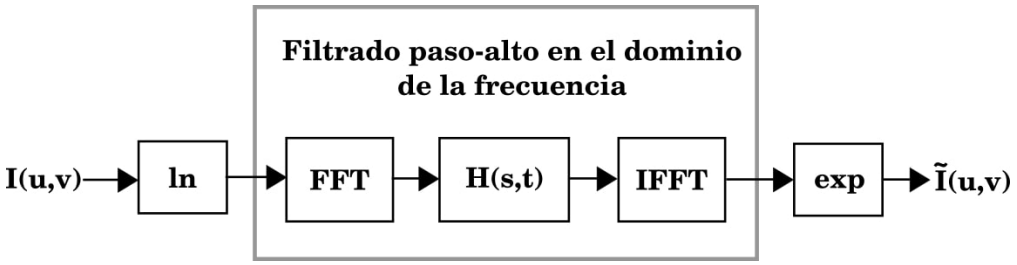


Figura 16: *Esquema del filtrado homomórfico en una imagen. Fuente: Elaboración propia*

2.2.5.3. Espacios de color

De acuerdo a Gonzalez y Woods (2001, pp. 289), el propósito de un espacio de color es facilitar la especificación de colores en algún estándar aceptado. En esencia, un espacio de color es una especificación de un sistema de coordenadas en un sub-espacio dentro del sistema donde cada color está representado por un punto. El uso de los espacios de color está orientado a mantener un estándar entre los diversos hardware de visualización o para editar imágenes. El espacio de color más conocido es el RGB (Figura 17), el cual está basado en el sistema de coordenadas cartesianas. Las imágenes que utilizan este espacio de color están conformadas por tres canales: rojo, verde y azul. Dichos canales tienen una profundidad de bit de 8-bit, conformando un total de 24-bit por pixel, lo cual hace posible un total de

16 777 216 colores diferentes. Otro espacio de color es el CMY-CMYK (Figura 18), aunque su uso está dirigido principalmente a las impresoras. Otro espacio de color muy utilizado es el HSV (*hue-saturation-value*), el cual, a diferencia de otros espacios, es una transformación no lineal del espacio RGB. A diferencia de los espacios RGB o CMYK, que están basados en las relaciones de los colores primarios, el espacio HSV es muy similar a como el humano ve los colores, haciéndolo ideal para los gráficos por computador. La forma de representar el espacio HSV es como un cono invertido en tres dimensiones (Figura 19). Adicionalmente han aparecido nuevos espacios de color que buscan asemejarse a la visión humana: CIEXYZ, CIELAB, CIELUV, entre otros.

Las operaciones más básicas que se pueden realizar sobre imágenes de color, son la segmentación por rangos de color (el espacio HSV es óptimo para esta operación), cambios de tono o matiz y detección de bordes.

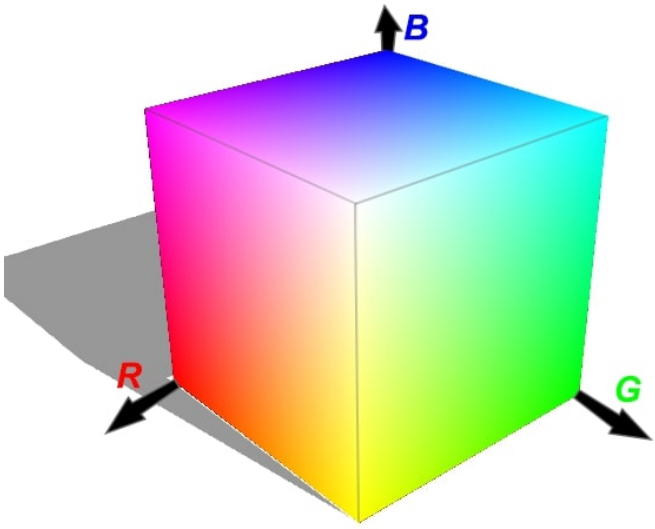


Figura 17: *Espacio de color RGB. Fuente: Elaboración propia*

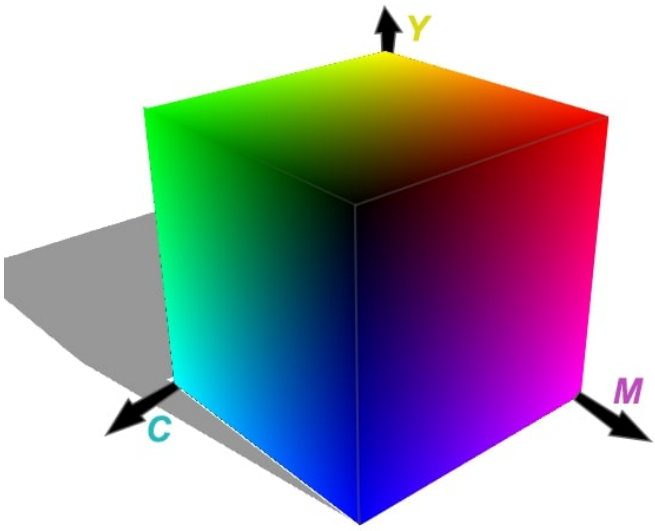


Figura 18: *Espacio de color CMY. Fuente: Elaboración propia*

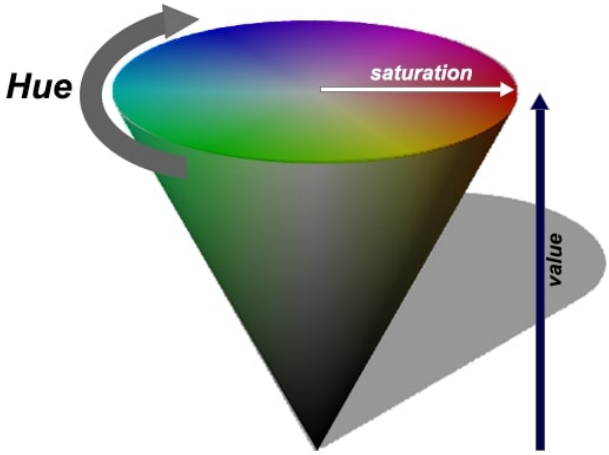


Figura 19: *Espacio de color HSV. Fuente: Elaboración propia*

2.2.5.4. Extracción de componentes conectados por rango de color

Se dice que existe conectividad entre dos pixeles si se cumple que son vecinos o adyacentes y si tienen valores de intensidad similares, o en el caso de imágenes binarizadas, deben tener valores iguales (0 o 1) (Gonzalez & Woods, 2001, pp. 66). La binarización de una imagen permite discriminar información innecesaria de la imagen (fondo) y extraer la mayor información posible de los objetos de interés. Existen varios métodos que permiten binarizar

una imagen, siendo el más conocido el de la binarización por umbral. Alegre et al. (2016, pp. 100) define al umbral como un valor de intensidad a partir del cual los pixeles de una imagen serán considerados como blancos, mientras que el resto serán etiquetados como negros. En la ecuación (16), se define la operación de binarización de una imagen $I(u, v)$ con un umbral T.

$$B(u,v) = \begin{cases} 0; & I(u,v) < T \\ 1; & I(u,v) \geq T \end{cases} \tag{16}$$

Para la obtención de imágenes binarizadas en imágenes de color que correspondan a rangos de matices (hue), es preciso convertir la imagen al espacio de color HSV. Como se precisa un rango de matiz y no un umbral, se debe establecer un valor mínimo H_{min} y un valor máximo H_{max}, quedando la operación de binarización por rango de matiz definida en (17).

$$H_T(u,v) = \begin{cases} 1; & H_{min} \leq H(u,v) \leq H_{max} \\ 0; & otro\ caso \end{cases} \tag{17}$$

La misma operación puede realizarse sobre los 2 canales restantes: saturación y valor, teniendo una imagen binarizada por rangos de matiz, saturación y valor. En la Figura 20, se observa el proceso de binarización por rangos en el espacio de color HSV.

Cabe resaltar que en la Figura 20b se observa la imagen binarizada, dentro de la cual existe una región de componentes conectados, sin embargo, una imagen binarizada puede contener múltiples regiones de componentes conectados, siendo necesaria una operación de etiquetaje (labeling) para extraer las características de cada región.

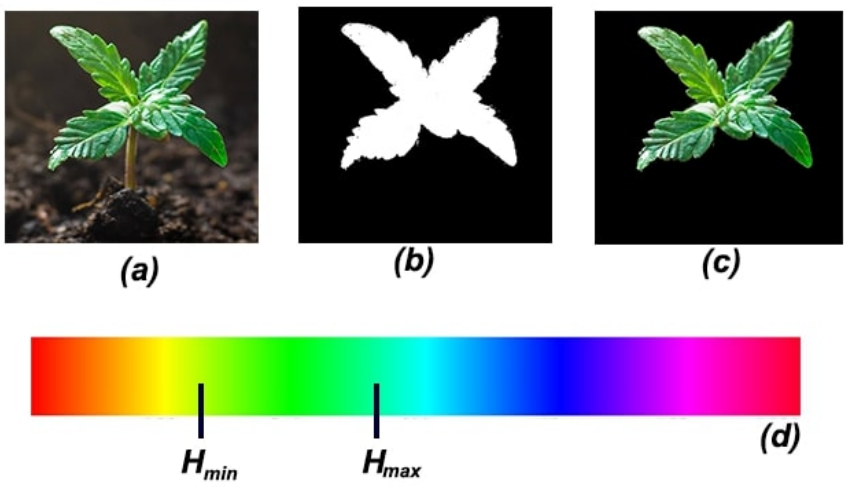


Figura 20: *Segmentación de una imagen por rango de matiz: (a) Imagen original; (b) Imagen binarizada o máscara; (c) Aplicación de máscara a imagen original; (d) Rango de matiz seleccionado. Fuente: Elaboración propia*

2.2.5.5. Operaciones morfológicas

Las operaciones morfológicas analizan imágenes binarizadas basándose en propiedades de forma, empleando teoría de conjuntos. Los conjuntos en una imagen, representan objetos. Un ejemplo se puede ver en la imagen Figura 20b, donde todos los pixeles blancos representan la pertenencia al objeto "planta". Según Gonzalez y Woods (2001, pp. 519), en las imágenes binarizadas, los conjuntos se encuentran incluidos en el espacio $\mathbb{Z}^2$, donde cada elemento es una tupla o vector cuyas coordenadas son (u, v). Los elementos que pertenezcan al conjunto pueden determinarse como 0 o 1 (negro o blanco). Para entender las operaciones morfológicas, es preciso entender las operaciones propias de la teoría de conjuntos. Siendo las operaciones básicas: la unión, intersección, complemento y diferencia. Sin embargo, es preciso agregar 2 operaciones: la reflexión de un conjunto y la traslación. Dichas operaciones se definen en (18) y (19) respectivamente.

$$\hat{B} = \{w \mid w = -b, \ \forall \ b \in B\} \tag{18}$$

$$(A)_z \ = \{c \mid c = a + z, \ \forall \ a \in A\} \tag{19}$$

- Dilatación y erosión:

  Estas operaciones morfológicas son la base para operaciones morfológicas más complejas. Pueden definirse de manera intuitiva como operaciones que eliminan o agregan elementos al conjunto de pixeles, de acuerdo a su vecindad. Formalmente las operaciones de dilatación y erosión se muestran en las ecuaciones (20) y (21).

$$A \oplus B \ = \{z \mid (\hat{B})_z \ \cap A = \varnothing\} \tag{20}$$

$$A \ominus B \ = \{z \mid (B)_z \ \subseteq A = \varnothing\} \tag{21}$$

  En (20) se puede apreciar que el conjunto resultado de dilatar A con B, está conformado por todos los desplazamientos posibles en la que la intersección de la reflexión de B con A tenga por lo menos un elemento. El efecto sobre la imagen binarizada es de un "crecimiento" de la región o dilatación. Por otro lado, en (21) se puede apreciar que el conjunto resultado de erosionar A con B está conformado por todos los desplazamientos de B, en los que todos los elementos de dicho desplazamiento estén incluidos en A. La condición de (21), es más restrictiva que la de la dilatación. De hecho, el efecto producido por esta operación es de una "reducción" de la región o erosión. Cabe resaltar que las operaciones de erosión y dilatación, a pesar de tener efectos contrarios, no son inversas entre sí. Dado que no son inversas entre sí, al operar una después de la otra se obtienen nuevas operaciones: el opening y el closing.

2.2.5.6. Operaciones sobre componentes conectados

Después de segmentar una imagen en regiones de componentes conectados (i.e. regiones), es preciso realizar operaciones que permitan describir la región. Por ejemplo, se puede representar una región en términos de su contorno, la cantidad de pixeles que comprenden la región, entre otros. Estas operaciones dan como resultado un descriptor o representador de la región especificada.

- Representador del tipo código de cadena (chain code):

  Este representador utiliza una secuencia conectada de segmentos de recta de una determinada distancia y dirección (basada en conectividad-4 o conectividad-8). Los códigos de cadena (Figura 21c) permiten sub-muestrear una región de manera que se pueda disminuir la cantidad de elementos que conforman la región. Otra opción es utilizar la primera diferencia del código de cadena para normalizar el factor de rotación. El código de cadena también puede aproximarse mediante polígonos (Figura 21d). La finalidad de estas aproximaciones es capturar la esencia de la forma del contorno de la región utilizando la menor cantidad posible de segmentos poligonales.

- Descriptores de un contorno:

  La longitud del contorno B puede definirse como el número de pixeles que lo conforman o también puede definirse en función del código de cadena según (22).

$$L(B) = S_{HV} + \sqrt{2}S_D \tag{22}$$

  Donde S_{HV} corresponde al número de segmentos horizontales o verticales y S_D corresponde al número de segmentos diagonales. El diámetro de un contorno B está definido en (23) como la máxima distancia entre dos puntos diferentes que pertenezcan al contorno.

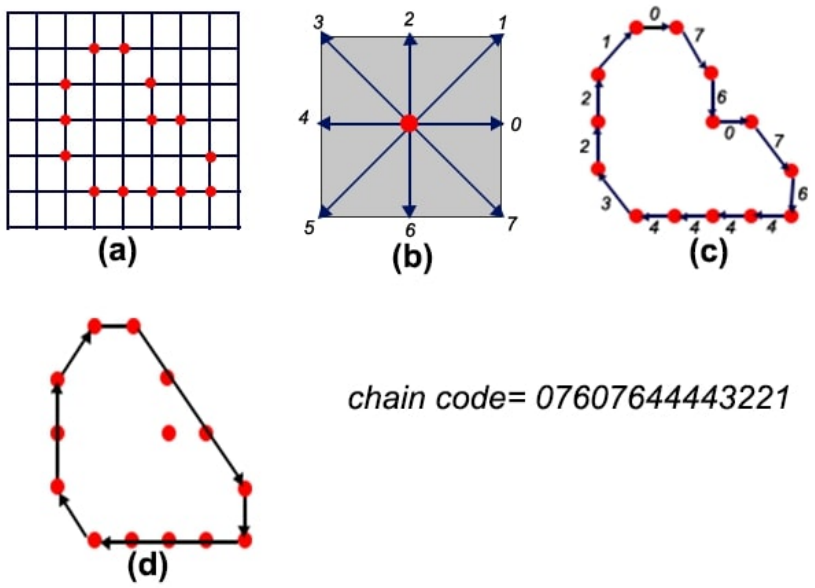


Figura 21: *Creación de un código de cadena: (a) Puntos del contorno; (b) Numeración de las direcciones; (c)Representación del código de cadena; (d) Aproximación poligonal. Fuente: Elaboración propia*

$$Diam(B) = max\big[D(p_i, p_j), p_i \neq p_j\big] \tag{23}$$

El diámetro y su orientación, puede utilizarse para encontrar el eje mayor y el eje menor (el eje perpendicular al eje mayor con una longitud determinada por el mínimo rectángulo circundante). La utilidad de encontrar estos ejes, es la posibilidad de encontrar el rectángulo que abarque toda la región, así como usar las dimensiones de dicho rectángulo para encontrar la excentricidad del contorno.

- Descriptores de Fourier:

Los descriptores de Fourier buscan describir un contorno en 2D como una función compleja en 1D. Para esto, cada elemento compuesto puede definirse como una variable compleja según (24) y en (25) se define la transformada de Fourier de los elementos complejos del contorno.

$$s(k) \;=\; u_k + jv_k \tag{24}$$

$$a(t) = \frac{1}{K}\sum_{k=0}^{K-1} s(k)e^{\frac{-j2\pi tk}{K}} \tag{25}$$

Cabe resaltar que los descriptores de Fourier son útiles para obtener los coeficientes más importantes y determinar la "esencia" de la forma del contorno, pudiendo ser útil para comparar dos contornos. En la Figura 22, se observa la reconstrucción de un contorno a partir de los coeficientes de Fourier, donde M representa el número de coeficientes utilizados para la reconstrucción.

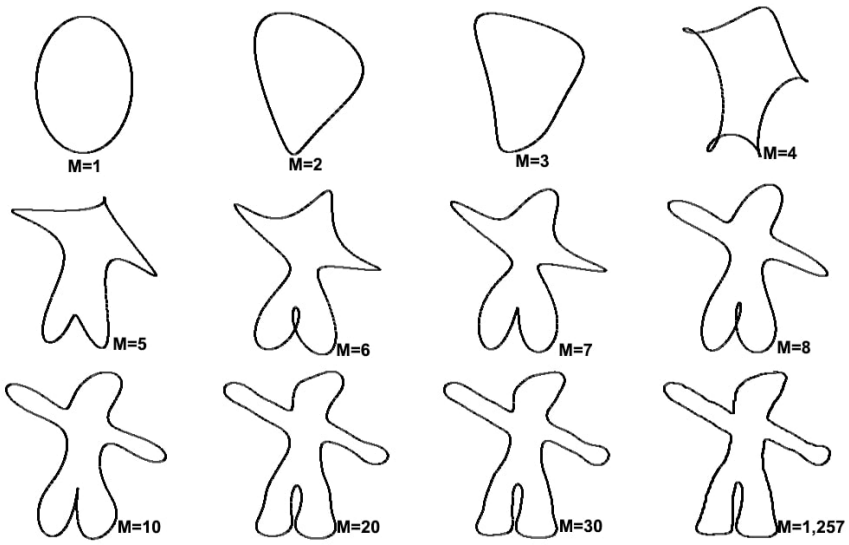


Figura 22: *Reconstrucción de contorno a partir de coeficientes de Fourier. Fuente: Wang (2013)*

- Momentos de un componente conectado: Un momento de imagen es cierto promedio ponderado particular de las intensidades de los píxeles de una imagen binaria; o también una función de tales momentos. El momento M_{pq} de la región $R(u, v)$ dentro de una imagen binarizada queda definida en (26).

$$M_{pq} = \sum_u \sum_v u^p v^q R(u, v) \tag{26}$$

Una región puede ser descrita de acuerdo a sus momentos (Gonzalez & Woods, 2001, pp. 675), sin embargo existen otras utilidades para estos momentos, tales como calcular el área total (en pixeles) de la región (M_{00}) o encontrar el centro de masa o centroide ($[C_u, C_v]$) de la región, el cual es útil para localizar a la región dentro de la imagen binarizada. En las ecuaciones (30) y (31) se define el cálculo del centroide de una región o componente conectado, dentro de una imagen binaria.

$$M_{00} = \sum_u \sum_v R(u, v) \tag{27}$$

$$M_{10} = \sum_u \sum_v uR(u, v) \tag{28}$$

$$M_{01} = \sum_u \sum_v vR(u, v) \tag{29}$$

$$C_u = \frac{M_{10}}{M_{00}} \tag{30}$$

$$C_v = \frac{M_{01}}{M_{00}} \tag{31}$$

2.2.6. Integración del sistema de visión por computador con robot cartesiano

Dado que una imagen digital corresponde a una proyección de puntos en un espacio cartesiano de tres dimensiones a un plano imagen de dos dimensiones, es preciso conocer los parámetros que definen esta proyección de manera que dadas ciertas condiciones (conociendo la distancia de la cámara al punto) podamos convertir pixeles de imagen digital a coordenadas absolutas del robot cartesiano.

2.2.6.1. Calibración de cámara

De acuerdo a Alegre et al. (2016, pp. 238), el proceso de calibración de una cámara es la estimación de los parámetros intrínsecos (propios de la cámara) y extrínsecos (relativo a un sistema referencial) de la misma.

Existen dos métodos principales para la calibración de una cámara. El más sencillo de entender es el método de la transformación linear directa (DLT), en el que un conjunto de puntos en 3D es relacionado con sus correspondientes puntos de proyección en el plano imagen digitalizado. Sin embargo, para llevar a cabo este método es preciso saber la ubicación exacta de al menos 6 puntos no co-planares lo cual dificulta mucho su utilización y puede generar errores debido a fallas en la ubicación.

Por otro lado, existe un método basado en homografías. Una homografía es la proyección de perspectiva que se puede utilizar cuando el escenario (puntos en 3D) corresponde a un plano. En otras palabras, la homografía es la proyección de un plano sobre el plano imagen del sensor de la cámara.

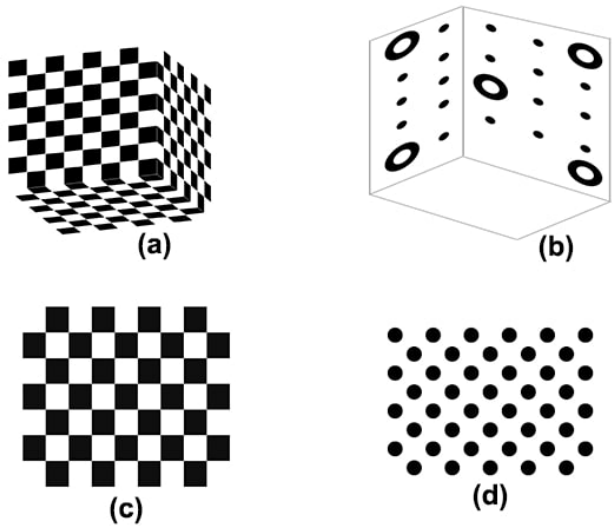


Figura 23: *Patrones de calibración: (a) Patrón ajedrez 3D; (b) Patrón circular 3D; (c) Patrón de ajedrez planar; (d) Patrón circular planar. Fuente: Elaboración propia*

Para simplificar el proceso de calibración, cada método necesita al menos un patrón conocido. En el caso del método DLT, se suele utilizar el patrón de Tsai (Figura 23a y Figura 23b) y en el caso del método por homografías se suelen utilizar el patrón de ajedrez (Figura 23c) o el patrón de círculos (Figura 23d), siendo el primero el más utilizado.

2.2.6.2. Parámetros intrínsecos

Permiten conocer las características propias de la cámara (independientemente a la ubicación de la cámara). En la ecuación (32), se define la matriz de parámetros intrínsecos, donde s_x y s_y son las escalas pixeles-milímetros (en el plano imagen) de los ejes X y Y respectivamente. La ubicación del centro del plano imagen en pixeles es $\mathbf{u_c} = (u_c, v_c)$ y f es la distancia focal de la cámara.

$$\mathbf{A} = \begin{bmatrix} fs_x & fs_\theta & u_c \\ 0 & fs_y & v_c \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \alpha & \beta & u_c \\ 0 & \gamma & v_c \\ 0 & 0 & 1 \end{bmatrix} \tag{32}$$

2.2.6.3. Parámetros extrínsecos

Permiten conocer las transformaciones de traslación ($\mathbf{t}$) y rotación ($\mathbf{R}$) de la cámara respecto al sistema de coordenadas del robot cartesiano. En la ecuación (33) se define la matríz $\mathbf{W}$ que integra la matríz de rotación y traslación en una sola. En las ecuaciones (34) y (35) se detalla la composición de cada matríz. La matríz de rotación se define en (36) en función de sus rotaciones alrededor de cada eje: θ_x, θ_y y θ_z.

$$\mathbf{W} = \begin{bmatrix} \mathbf{R_{3\times 3}} & \mathbf{t_{3\times 1}} \\ \mathbf{0_{1\times 3}} & 1 \end{bmatrix} \tag{33}$$

$$\mathbf{t} = \begin{bmatrix} t_x & t_y & t_z \end{bmatrix}^\top \tag{34}$$

$$\mathbf{R} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}^\top \tag{35}$$

$$\mathbf{R} = \begin{bmatrix} cos(\theta_y)cos(\theta_z) & -cos(\theta_y)sin(\theta_z) & sin(\theta_y) \\ cos(\theta_x)sin(\theta_z) + sin(\theta_x)sin(\theta_y)sin(\theta_z) & cos(\theta_x)cos(\theta_z) - sin(\theta_x)sin(\theta_y)sin(\theta_z) & -sin(\theta_x)cos(\theta_y) \\ sin(\theta_x)sin(\theta_z) - cos(\theta_x)sin(\theta_y)cos(\theta_z) & sin(\theta_x)cos(\theta_z) + cos(\theta_x)sin(\theta_y)sin(\theta_z) & cos(\theta_x)cos(\theta_y) \end{bmatrix} \tag{36}$$

2.2.6.4. Calibración por patrón planar: Método de Zhang

Para este tipo de calibración se recurre al uso de homografías en conjunto con el análisis DLT. Es preciso conocer los puntos de referencia del mundo real (X_j) y los puntos observados (u_{ij}) por la cámara. En las ecuaciones (37), (38) y (39) se detalla el proceso de construcción de la matriz de homografía H_i.

$$s\ \mathbf{u_{ij}^{\cdot}}\ =\ \mathbf{AWX_j} \tag{37}$$

$$s \begin{bmatrix} u_{ij}^{\cdot} \\ v_{ij}^{\cdot} \\ 1 \end{bmatrix} = A \begin{bmatrix} r_{11} & r_{12} & t_x \\ r_{21} & r_{22} & t_y \\ r_{31} & r_{32} & t_z \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_j \\ Y_j \\ 1 \end{bmatrix} \tag{38}$$

$$\mathbf{H_i} = \lambda \mathbf{A} \begin{bmatrix} r_{11} & r_{12} & t_x \\ r_{21} & r_{22} & t_y \\ r_{31} & r_{32} & t_z \\ 0 & 0 & 1 \end{bmatrix} \tag{39}$$

Para encontrar la matriz $\mathbf{H_i}$, se requiere reducir el problema a un sistema homogéneo de ecuaciones lineales $\mathbf{Mh} = 0$, donde $\mathbf{h}$ corresponde a un vector formado con los elementos de $\mathbf{H}$; y $\mathbf{M}$ corresponde a un arreglo conformado por todos los puntos de referencia y los puntos observados. La resolución de este sistema parte por definir la matriz $\mathbf{M}$ como una matriz singular. Burger (2016) detalla los pasos para llevar a cabo la calibración de cámara por el método de Zhang, incluyendo la descomposición en valores singulares (SVD) de la matriz $\mathbf{M}$. El propósito de la calibración no es encontrar la matriz de homografía en una imagen, sino obtener los parámetros intrínsecos de la cámara. Es por ello que a partir de todas las matrices de homografía: $\mathbf{H_0}$,...,$\mathbf{H_{M-1}}$, se debe encontrar la matriz de $\mathbf{A}$. Como ya se mencionó anteriormente, si el sistema está sobre determinado, se debe utilizar SVD para resolver el sistema. El proceso empieza por definir los vectores de rotación de la matriz como ortonormales, tal como se define en las ecuaciones (41) y (42).

$$\tilde{\mathbf{W}} = \begin{bmatrix} r_{11} & r_{12} & t_x \\ r_{21} & r_{22} & t_y \\ r_{31} & r_{32} & t_z \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} | & | & | \\ \mathbf{r_0} & \mathbf{r_1} & \mathbf{t} \\ | & | & | \end{bmatrix} \tag{40}$$

$$\mathbf{r_0}^\top \, \mathbf{r_1} = \mathbf{r_1}^\top \, \mathbf{r_0} = 0 \tag{41}$$

$$\mathbf{r_0}^\top \, \mathbf{r_0} = \mathbf{r_1}^\top \, \mathbf{r_1} = 1 \tag{42}$$

De las ecuaciones (39), (41) y (42) se puede definir un sistema lineal según (43) y (44).

$$\mathbf{h_0}^\top \mathbf{B} \, \mathbf{h_1} = 0 \tag{43}$$

$$\mathbf{h_0}^\top \mathbf{B} \, \mathbf{h_0} - \mathbf{h_1}^\top \mathbf{B} \, \mathbf{h_1} = 0 \tag{44}$$

Donde $\mathbf{B} = (\mathbf{A}^{-1})^\top \mathbf{A}^{-1}$ y $\mathbf{h_p} = \begin{bmatrix} H_{p,0} & H_{p,1} & H_{p,2} \end{bmatrix}$; $\forall p \in \{0, 1, 2\}$.

Hasta el momento se ha considerado a la cámara como ideal (sin distorsión óptica), sin embargo, para obtener un modelo completo de la relación entre los puntos de referencia y los puntos observados es necesario estimar los coeficientes de distorsión. Estos coeficientes corresponden a una distorsión radial de la lente de la cámara, es decir que la distorsión depende de r (radio) definido como la distancia de un pixel al pixel central o centro de proyección (u_c, v_c). En la ecuación (45) se define la operación de distorsión o "warping".

$$\tilde{\mathbf{u}} = warp(\mathbf{u}, \mathbf{k}) = \mathbf{u}\big[1 + D(r, \mathbf{k})\big] \tag{45}$$

El objetivo principal es obtener $\mathbf{k}$ que es el vector de coeficientes de distorsión radial. Para esto se parte del conjunto de puntos proyectados $\mathbf{u_{i,j}}$ (puntos calculados) y los puntos observados $\mathbf{\dot{u}_{i,j}}$ puntos que realmente ve el sensor), de manera que se pueda calcular su desviación propia $\mathbf{d_{i,j}} = [\mathbf{u_{i,j}} - \mathbf{u_c}]D(\mathbf{r_{i,j}}, \mathbf{k})$ y su desviación observada $\mathbf{\dot{d}_{i,j}} = \mathbf{\dot{u}_{i,j}} - \mathbf{u_{i,j}}$. El objetivo es minimizar la diferencia entre ambas desviaciones tal como se define en la ecuación (46).

$$\sum_{i,j} \|\mathbf{d_{i,j}} - \mathbf{\dot{d}_{i,j}}\| \longrightarrow min \tag{46}$$

Al final el problema queda definido de la siguiente manera: Dado un sistema sobredeterminado $\mathbf{d_{i,j}} = \mathbf{\dot{d}_{i,j}}$, determinar $\mathbf{k}$ utilizando mínimos cuadrados.

2.2.6.5. Uso de parámetros extrínsecos de la cámara

Como se mencionó anteriormente, los parámetros extrínsecos de una cámara digital permiten localizar un sistema de coordenadas específico (puede ser el sistema de coordenadas de un robot) con respecto al sistema de coordenadas de la cámara. En otras palabras, si mediante la imagen capturada con una cámara digital se puede determinar la ubicación de un objeto respecto a la cámara, los parámetros extrínsecos permitirían determinar la ubicación del mismo objeto, pero respecto a otro sistema de coordenadas. A efecto de generar trayectorias para un robot cartesiano mediante visión por computador, se debe determinar la matriz de parámetros extrínsecos que relaciona el sistema de coordenadas de la cámara con el sistema de coordenadas del robot cartesiano.

2.3. Definición de términos básicos

- Algoritmos de visión por computador de bajo nivel: Es el conjunto de algoritmos básicos en los cuales se transforma una imagen para obtener otra, de manera que la imagen obtenida presente características mejoradas. Dentro de este conjunto se encuentran los

algoritmos de filtrado (en espacio y en frecuencia), ecualización, operaciones geométricas, operaciones morfológicas y segmentación.

- Algoritmos de visión por computador de mediano nivel: Es el conjunto de algoritmos básicos en los cuales se procesa una imagen para obtener ciertas características o descriptores. Dentro de este conjunto se encuentran los algoritmos de análisis de componentes conectados, identificación de contornos, obtención de puntos de interés, etc.

- Cámara digital: La cámara es el dispositivo que, utilizando un objetivo formado por un juego de lentes y un diafragma, construye una imagen sobre el plano del sensor compuesto por elementos fotosensibles, la digitaliza y la transmite hacia la tarjeta de adquisición del procesador. Están compuestas por un sensor y la electrónica asociada.

- Componente conectado: Es aquel conjunto de pixeles (región) de una imagen que tienen el mismo valor y entre los cuales siempre existe un camino de pixeles con el mismo valor, de forma que se pueda ir de un pixel a otro sin abandonar la región que los contiene.

- Homografía: Una homografía es una transformación proyectiva en perspectiva que determina una correspondencia biyectiva entre los puntos de dos planos. En otras palabras, es la proyección de los puntos que componen un plano, sobre el plano imagen de una cámara digital.

- Imagen digital: Una imagen digital es aquella matriz obtenida mediante una cámara digital, cuyos elementos toman valores discretos y representan la intensidad luminosa de los objetos que componen la escena capturada por la cámara.

- Operaciones morfológicas: Son operaciones matemáticas basadas en la teoría de conjuntos, las cuales operan sobre una imagen binarizada (aquella cuyos pixeles solo pueden tomar dos valores) de manera que pueden modificar la forma de los componentes conectados que conforman la imagen binarizada.

- Parámetros extrínsecos: Son los elementos de una matriz, la cual permite conocer las

operaciones de rotación y traslación realizadas al sistema de coordenadas de la cámara, con la finalidad de relacionarlo con el sistema de coordenadas de un robot cartesiano.

- Parámetros intrínsecos: Son aquellos valores constantes y propios de una cámara digital, los cuales definen la operación de proyección de los puntos que conforman los objetos presentes dentro de la escena capturada por la cámara.
- Plano imagen: Es el plano asociado al sensor fotosensible propio de una cámara digital.
- Robot cartesiano: Es un robot de 3 grados de libertad, en el cual solo se realizan movimientos lineales para los 3 ejes ortogonales del robot.
- Visión por computador: Es una rama científica interdisciplinaria cuyo objetivo es interpretar la escena captada en una imagen digital, utilizando un ordenador.

Capítulo III

MATERIAL Y MÉTODOS

3.1. Material

3.1.1. Población

Plantines de alcachofa Imperial Star con un tiempo de 20 dias después de la siembra.

3.1.2. Muestra

123 plantines de alcachofa Imperial Star germinados en el vivero Hortifrut Plant, los cuales estarán distribuidos en tres calidades.

3.1.3. Unidad de análisis

Plantín de alcachofa Imperial Star

3.2. Métodos

3.2.1. Nivel de investigación

La presente investigación es de nivel aplicativo ya que se busca desarrollar un sistema de visión por computador y evaluar la influencia del uso de un conjunto de algoritmos sobre los

movimientos del robot cartesiano durante la actividad de repique de plantines de alcachofa. Por otro lado, no se busca explicar la razón por la cual los algoritmos puedan inducir movimientos erróneos en el robot, si no que se busca validar la utilización de estos algoritmos para llevar a cabo el repique y a la vez evaluar su performance.

3.2.2. Diseño de investigación

El diseño de investigación utilizado para la presente investigación es del tipo preexperimental con una sola medición, dado que no se manipularán los algoritmos de visión por computador si no que se evaluará su aplicación sobre la generación de trayectorias para el robot cartesiano. Además, no se tendrá un grupo de contrastación sobre el cual no se apliquen los algoritmos, dado que es preciso realizar la aplicación de estos para poder generar movimientos en el robot, ya sean erróneos o acertados. Por otra parte, solo se realizará una medición de las variables independientes y dependiente, la cual se llevará a cabo cuando se hayan generado las trayectorias para el sistema robótico.

3.2.3. Procedimientos

- Determinar las etapas de procesamiento y análisis de las imágenes (Figura 24), necesarias para la clasificación de los plantines y para generar trayectorias al sistema robótico. Para esto se realizará una búsqueda bibliográfica en la cual se resuman las técnicas de procesamiento de imágenes (algoritmos de bajo nivel), de manera que se puedan identificar los beneficios y desventajas de usar cierto algoritmo. El producto de esta etapa será un diagrama de bloques en el que se detalle el flujo de datos que ocurrirá desde la captura de las imágenes de los plantines hasta la generación de trayectorias.

- Identificadas las etapas de procesamiento y análisis de las imágenes obtenidas con el sistema de visión por computador del robot, se procedió a llevar a cabo la etapa inicial: La calibración de la cámara. Para esto, se procedió de acuerdo a OpenCV (2021), capturándose 10 imágenes de un patrón planar de ajedrez cuyas dimensiones eran cono-

cidas. En la Figura 25 se puede observar como gracias a la librería OpenCV, es posible obtener los vértices presentes en el patrón de ajedrez. En efecto gracias a la función *cv2.findChessboardCorners()* es posible obtener un listado con las ubicaciones (en pixeles) de cada vértice dentro de la imagen.

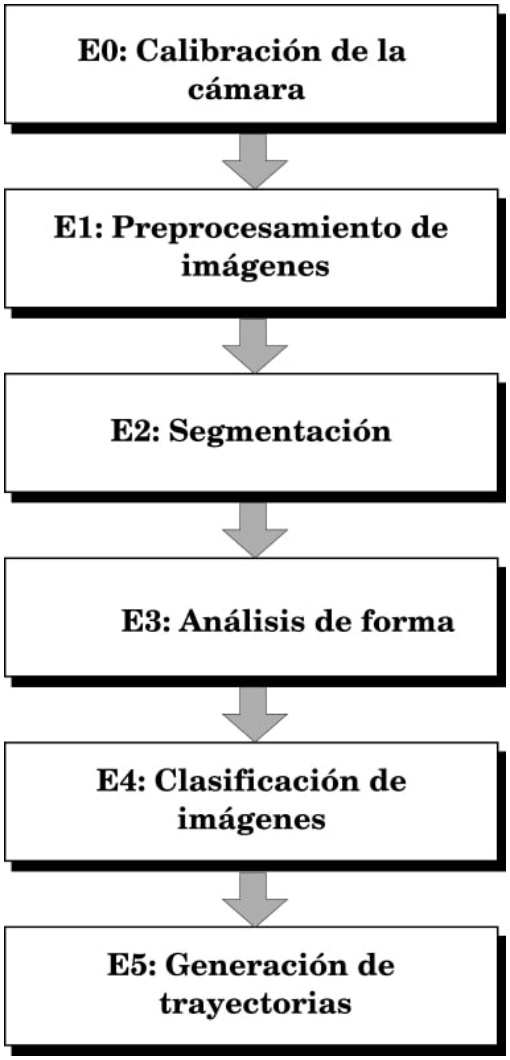


Figura 24: *Etapas de procesamiento y análisis de imágenes. Fuente: Elaboración propia*

El objetivo de la primera parte de esta etapa fue obtener la matriz de parámetros intrínsecos **A** de la cámara. Una vez obtenida **A** se procedió a obtener los parámetros extrínsecos de la cámara que relacionan el sistema de referencia de la cámara con el del efector final.

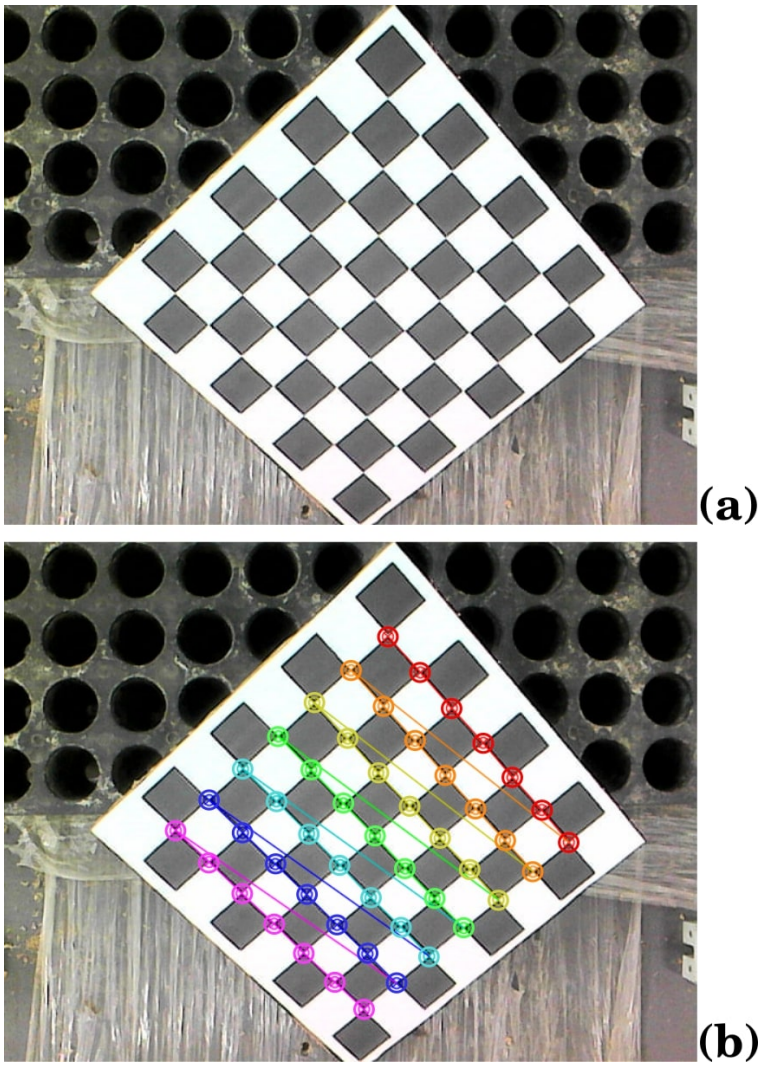


Figura 25: *Obtención de vértices en patrón de ajedrez: (a) Imagen original de patrón de ajedrez; (b) Imagen con vértices identificados. Fuente: Elaboración propia*

Para esta segunda parte, se utilizó el mismo patrón planar de ajedrez, con la diferencia que en esta ocasión, las localizaciones (respecto al efector final) de los puntos (o esquinas) del patrón de ajedrez eran conocidas. Para esto se midió la distancia que había desde la cámara al patrón de ajedrez (Figura 26) y se localizó cada vértice del patrón dentro del sistema de referencia del espacio de trabajo del robot (Figura 27).

De acuerdo a la Figura 28, cada vertice del patrón de ajedrez (colocado dentro del espacio de trabajo del robot) tiene asociado una posición absoluta $\mathbf{P_n}$, la cual representa la posición del n-ésimo punto del patrón respecto al sistema de referencia del robot.

Además, ese mismo vértice queda representado como el pixel $\mathbf{u_n}$ dentro de la imagen digital del patrón. Adicionalmente, como puede observarse en Figura 28, existe un vector $\mathbf{T}$, el cual representa la posición del efector final; y el vector $\tilde{\mathbf{P}}_\mathbf{n}$, el cual representa la ubicación del n-ésimo punto del patrón respecto al sistema de referencia del efector final.

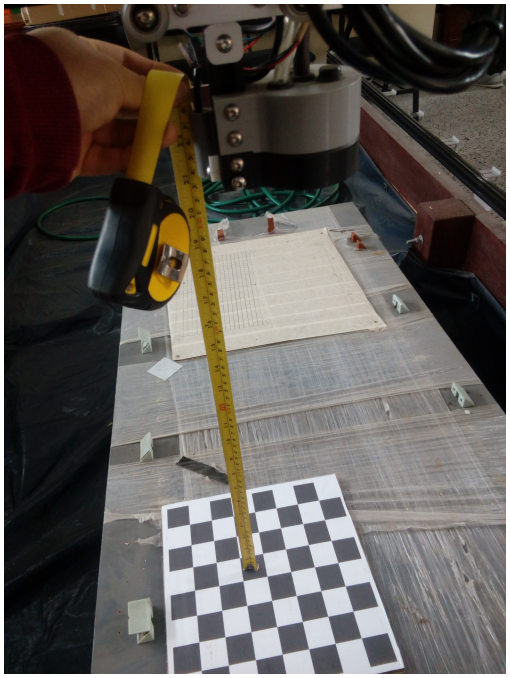


Figura 26: *Medición de la distancia cámara - patrón. Fuente: Elaboración propia*

Dados estos datos (posiciones de un mismo punto desde diferentes sistemas de referencia), es preciso relacionarlos de manera que podamos obtener los parámetros extrínsecos de la cámara. Este procedimiento puede llevarse a cabo con la función *cv2.calibrateCamera()*, la cual al ingresarle las ubicaciones de los vértices en pixeles ($\mathbf{u_n}$) y las ubicaciones respecto al efector final $\tilde{\mathbf{P}}_\mathbf{n}$, arroja la matriz de rotación $\mathbf{R}$ (en forma de vector de Rodriguez) y el vector de traslación $\mathbf{t}$; ambos detallados en (33) , (34) y (35).

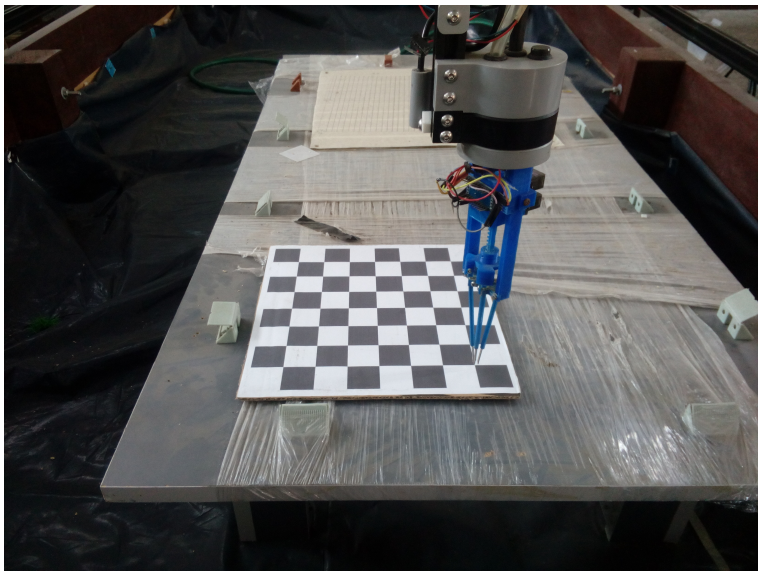


Figura 27: *Posicionamiento del efector final sobre un vértice del patrón de ajedrez. Fuente: Elaboración propia*

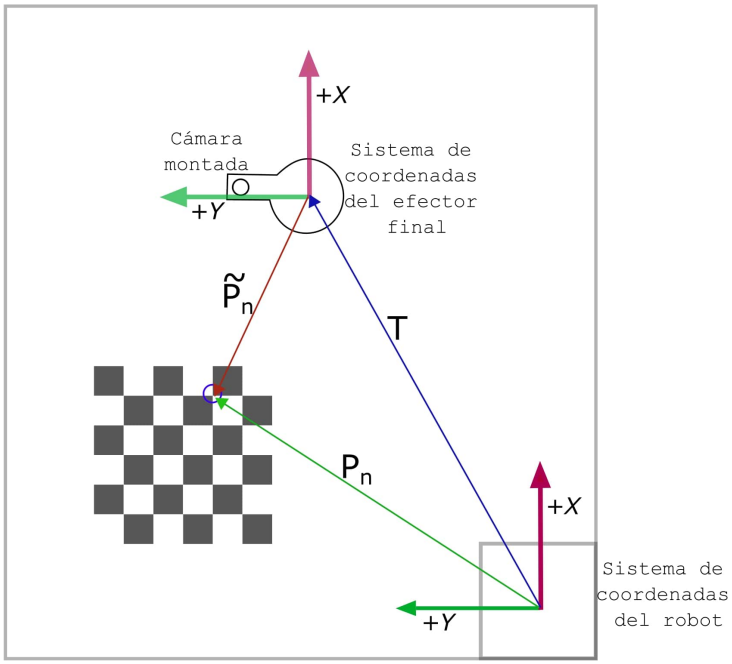


Figura 28: *Colocación del patrón de ajedrez dentro del espacio de trabajo del robot. Fuente: Elaboración propia*

- Como se mencionó anteriormente, el pre-procesamiento de imágenes es importante para facilitar tareas mas complejas que requieran tratar con imágenes en condiciones de iluminación diferente o con ruido. Como primera medida para reducir los problemas ocasionados por la iluminación en las imágenes capturadas, se optó por variar los parámetros de captura propios de la cámara mediante el método *VideoCapture().set(property,value)* de la librería *OpenCV*, tal como se observa en la Tabla 3 y se realizaron capturas con cada conjunto de parámetros de la Tabla 3. Los resultados de estas variaciones se muestran en la Figura 29, evidenciando que los valores de parámetros correspondientes a P_2 son los óptimos, debido a que esta combinación resalta los colores y disminuye el excesivo brillo mostrado en la Figura 29a.

Propiedad de la cámara	**Valor de fábrica** P_0	**Valor modificado** P_1	**Valor modificado** P_2	**Valor modificado** P_3
Width	640	640	640	640
Height	480	480	480	480
Brightness	0.5	0.4	0.35	0.2
Contrast	0.733	0.8	0.73	0.85
Saturation	0.3543	0.35	0.5	0.5
Hue	0.5	0.5	0.5	0.5

Tabla 3: *Cambios en los parámetros de captura de la cámara.Fuente: Elaboración Propia*

Como segunda medida para mitigar las condiciones de iluminación variable, se optó por utilizar dos algoritmos ampliamente conocidos en visión por computador y detallados en el Capítulo II: la ecualización de histograma y el filtrado homomórfico; para observar

sus resultados y definir cual de los dos puede robustecer el proceso de segmentación. De acuerdo a la investigación de Myler et al. (1995), el uso del filtro paso-altos de tipo "Butterworth", arrojó buenos resultados al aplicar el filtrado homomórfico. En la presente investigación, se evaluará el performance de la utilización de tres tipos de filtros paso alto: "Ideal" (Figura 30),"Gaussiano" (Figura 31) y "Butterworth" (Figura 32), con sus correspondientes parámetros (Tabla 4).

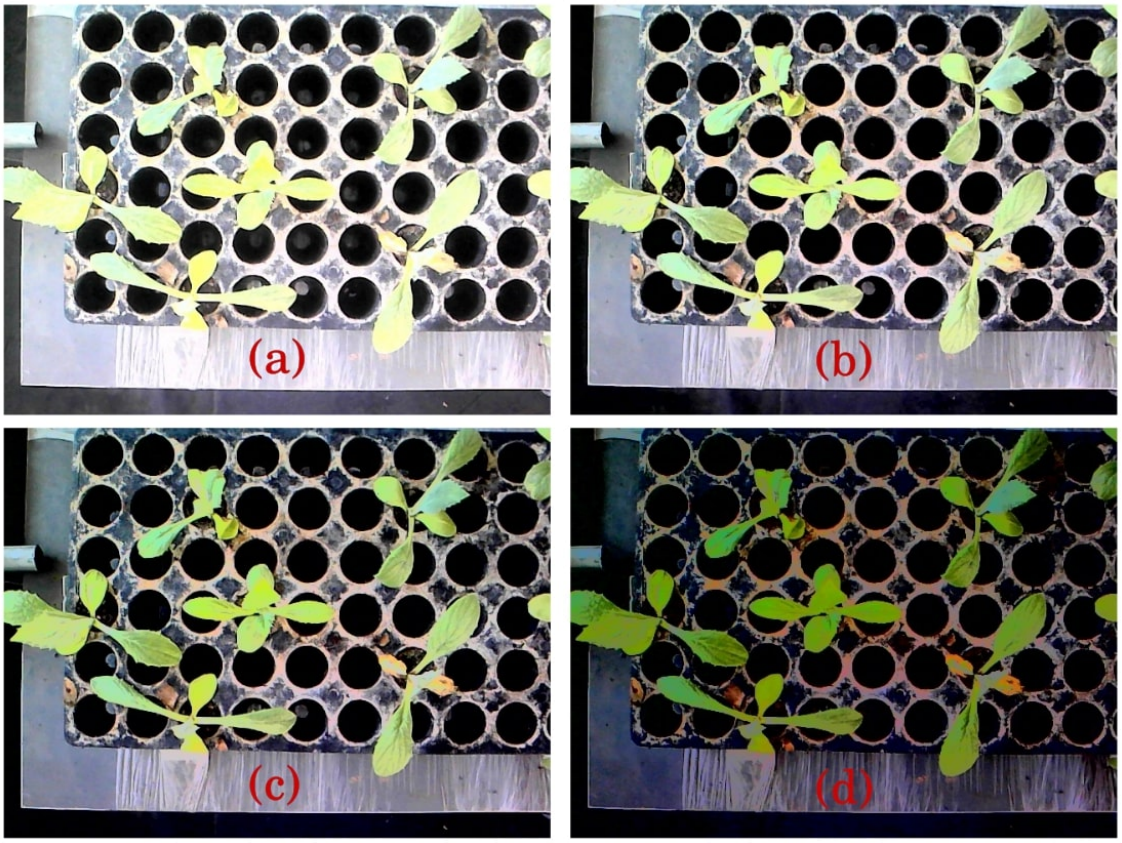


Figura 29: *Resultados de la variación en los parámetros de captura de la cámara: (a) Imagen con parámetros iniciales P_0; (b) Imagen con parámetros modificados P_1; (c) Imagen con parámetros modificados P_2; (d) Imagen con parámetros modificados P_3. Fuente: Elaboración propia*

Filtro	**Parámetros**	**Descripción**	**Valores**
Filtro ideal Paso alto	r_c	Radio de corte	$r_c < min(M, N)$
	γ_l	Ganancia en radios bajos	$0 < \gamma_l < \gamma_h$
	γ_h	Ganancia en radios altos	$\gamma_l < \gamma_h$
Filtro Gaussiano Paso alto	σ	Desviación estándar	$0 \leq \sigma$
	γ_l	Ganancia en radios bajos	$0 < \gamma_l < \gamma_h$
	γ_h	Ganancia en radios altos	$\gamma_l < \gamma_h$
Filtro Butterworth Paso alto	n	Orden	$1 \leq n$
	r_c	Radio de corte	$r_c < min(M, N)$
	γ_l	Ganancia en radios bajos	$0 < \gamma_l < \gamma_h$
	γ_h	Ganancia en radios altos	$\gamma_l < \gamma_h$

Tabla 4: *Parámetros de los filtros paso altos a utilizar sobre una imagen de dimensiones $M \times N$. Fuente: Elaboración propia*

A continuación se expresarán las ecuaciones que definen las funciones de filtro de tres filtros en el dominio de la frecuencia. Para esto es preciso resaltar que los conceptos de radio $r(s,t)$ y radio de corte r_c (mencionado en la Tabla 4) están relacionados con la distancia euclidiana existente entre un punto $[s,t]$ de la función que define al filtro en frecuencia (Función de filtro) y el centro de este $[s_c, t_c]$. En el caso del radio, este queda

definido en (47).

$$r(s,t) = \sqrt{(s - s_c)^2 + (t - t_c)^2} \tag{47}$$

El filtro "Gaussiano" paso alto queda definido en

$$H_{gauss}(s,t) = \gamma_h - (\gamma_h - \gamma_l) e^{\frac{(s-s_c)^2 + (t-t_c)^2}{2\sigma^2}} \tag{48}$$

El filtro "Ideal" paso alto queda definido en

$$H_{ideal}(s,t) = \begin{cases} \gamma_h; \ r \leq r_c \\ \\ \gamma_l; \ r > r_c \end{cases} \tag{49}$$

El filtro de "Butterworth" paso alto queda definido en

$$H_{butt}(s,t) = \gamma_h + \frac{\gamma_l - \gamma_h}{1 + 2.415(\frac{r}{r_c})^{2n}} \tag{50}$$

Como ya se menciónó anteriormente el objetivo del filtrado homomórfico es evitar que las variaciones en iluminación afecten el resto de etapas del sistema de visión por computador, es por ello que se escogieron los parámetros que mejor cumplieran con este objetivo. Para lograr esto, se introdujeron dos imágenes (Figura 33) de una misma escena (i.e. bandeja con plantines dentro del espacio de trabajo del robot) con iluminaciones distintas. Debido a que no existe una forma determinística de encontrar los valores óptimos de los parámetros que definen a los filtros a utilizar, se optó por utilizar el método de ensayo y error, utilizando valores cercanos a los propuestos por Myler et al. (1995), los cuales son: $\gamma_l = 0.8$, $\gamma_h = 1.8$, $r_c = 100$.

En la Tabla 5 se muestran todos los valores utilizados durante el proceso de ensayo y error, colocando en la columna final una observación de los resultados obtenidos.

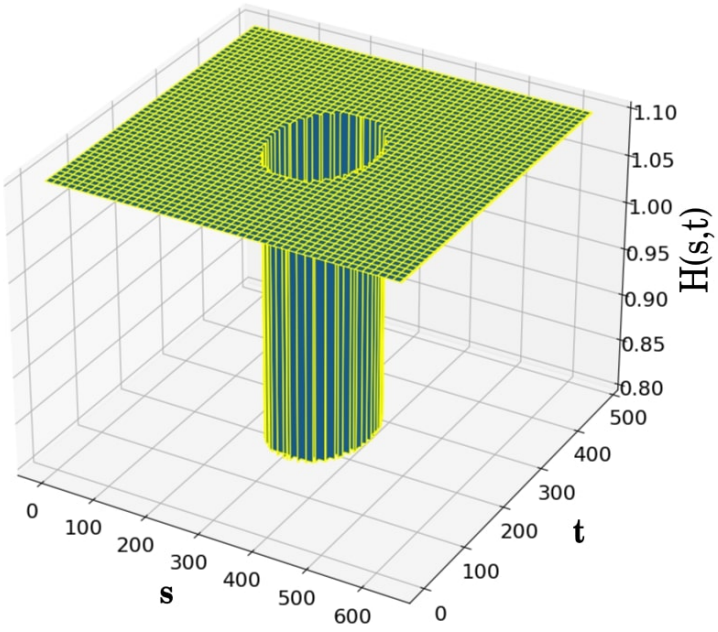


Figura 30: *Diagrama de Función de filtro paso altos ideal en el dominio de la frecuencia ($r_c = 90$, $\gamma_l = 0.8$, $\gamma_h = 1.1$). Fuente: Elaboración propia*

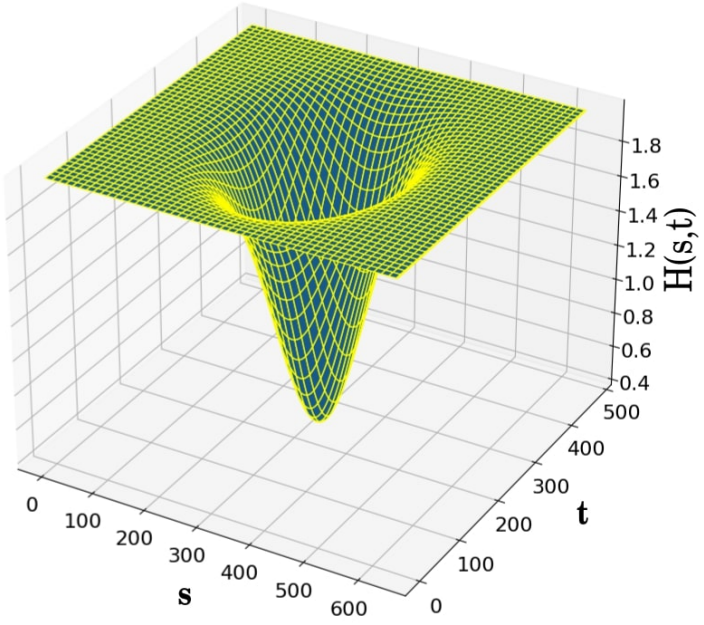


Figura 31: *Diagrama de Función de filtro paso altos Gaussiano en el dominio de la frecuencia. ($\sigma = 70$, $\gamma_l = 0.4$, $\gamma_h = 2$). Fuente: Elaboración propia*

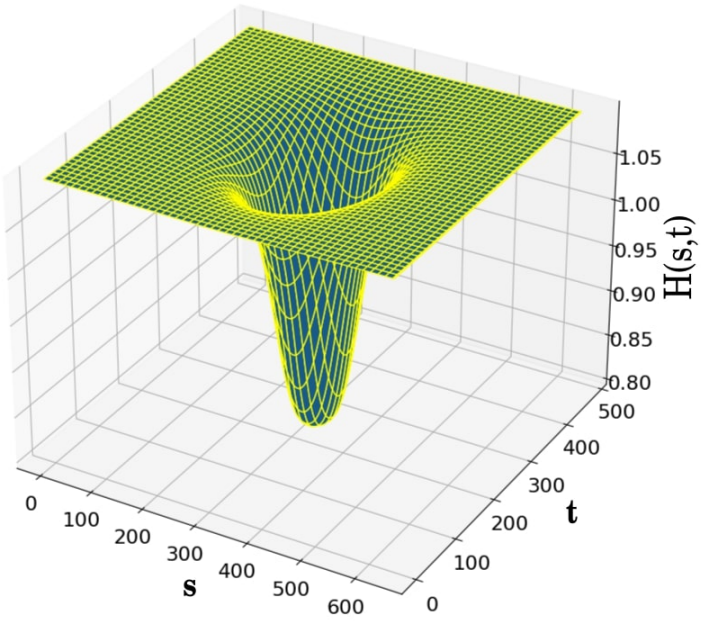


Figura 32: *Diagrama de Función de filtro paso altos ideal Butterworth en el dominio de la frecuencia. ($r_c = 90$, $\gamma_l = 0.8$, $\gamma_h = 1.1$, $n = 2$). Fuente: Elaboración propia*

Después de aplicar el procedimiento de ensayo y error a los parámetros de los filtros mencionados, se consideraron las imágenes obtenidas después de aplicar el filtrado homomórfico (Figura 34, Figura 35, Figura 36) y la Tabla 5 para decidir que el filtro Butterworth con $\gamma_l = 0.8$,$\gamma_h = 1$, $r_c = 30$,$n = 3$ es el más indicado para este tipo de imágenes (i.e. imágenes que contienen plantines colocados sobre una bandeja).

- Para la tarea de segmentación se recurrió al uso de rangos de valores, en otras palabras establecer un valor mínimo y máximo permitido, para cada canal. Primero la imagen se convirtió del espacio de color BGR a HSV. Luego se estableció un valor mínimo y máximo para cada canal (i.e. H, S, V) con la finalidad de binarizar la imagen $I_{HSV}(u, v)$. La operación de binarización queda definida en la ecuación (51), donde $H(u, v)$, $S(u, v)$, $V(u, v)$ representan los tres canales de la imagen $I_{HSV}(u, v)$.

Filtro	Parámetros					Observaciones
	γ_l	γ_h	r_c	n	σ	
Filtro Butterworth paso alto	0.4	1	30	3	-	Existen regiones con colores muy diferentes al original.
	0.6	1	30	3	-	Los colores de la imagen aún están distorsionados pero en menor magnitud.
	0.8	1	30	3	-	Se resaltan los colores, especialmente en la región de los plantines.
	0.8	1	30	2	-	No hay mucha variación respecto al anterior.
Filtro Ideal paso alto	0.4	1	40	-	-	Regiones de la imagen con colores distorsionados.
	0.6	1	40	-	-	No existe mucha variación respecto al anterior.
	0.8	1	40	-	-	Se resaltan los colores, pero algunas partes de la bandeja se confunden con los plantines.
Filtro Gaussiano paso alto	0.4	1	-	-	60	Se resaltan los colores, pero algunas regiones varían de color notablemente.
	0.6	1	-	-	60	Se resaltan los colores, pero algunas partes de la bandeja se confunden con los plantines.
	0.8	1	-	-	30	Se reduce ligeramente la distorsión de colores.

Tabla 5: *Valores de parámetros de los filtros paso altos utilizados. Fuente: Elaboración propia*

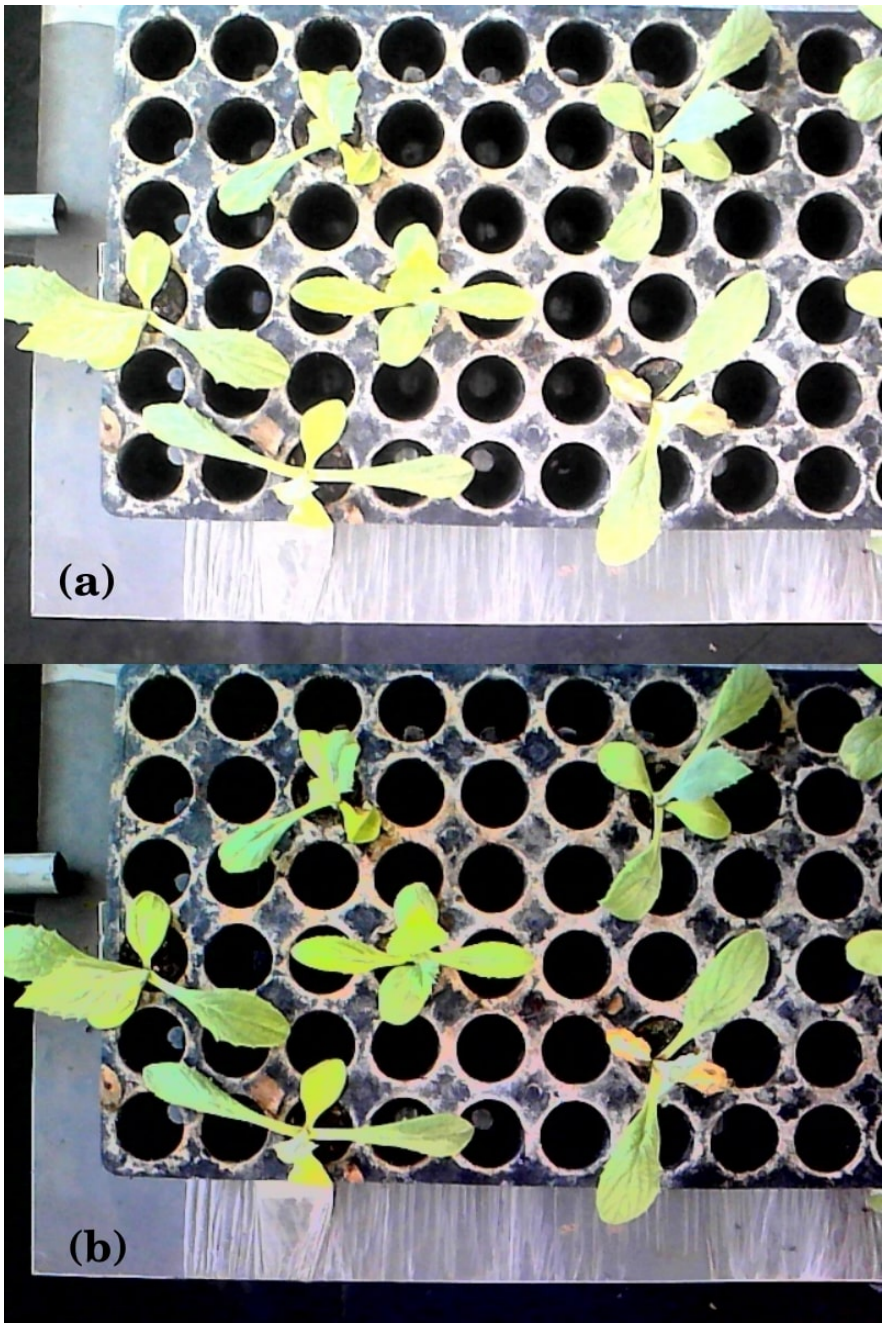


Figura 33: *Imágenes utilizadas para evaluar el filtrado homomórfico.(a) Imagen tomada durante la mañana con luz natural. (b) Imagen tomada durante la tarde con luz artificial. Fuente: Elaboración propia*

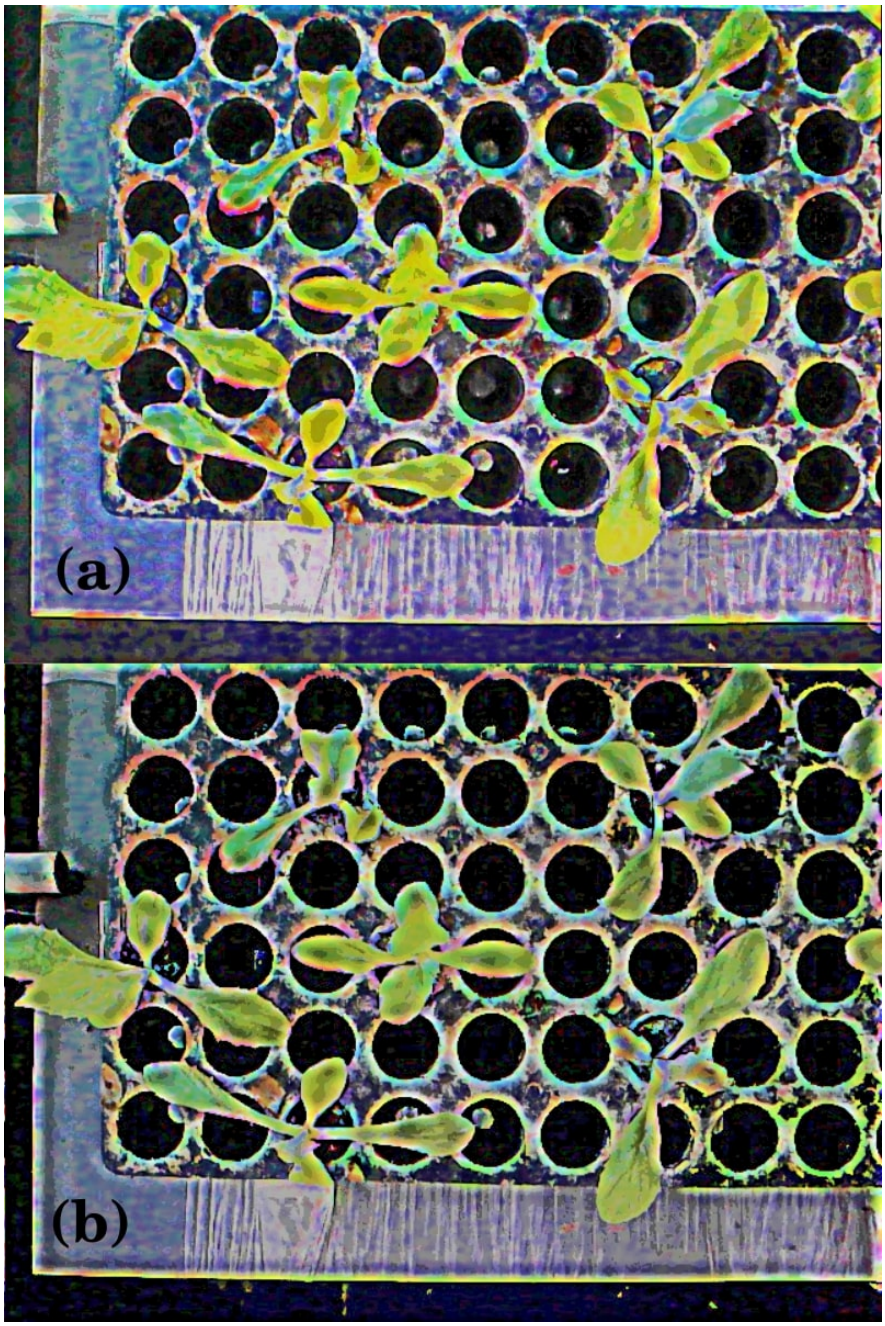


Figura 34: *Imágenes producidas al aplicar el filtro Butterworth con* $\gamma_l = 0.4$,$\gamma_h = 1$, $r_c = 30$,$n = 3$ *.(a) Imagen generada a partir de la Figura 33a. (b) Imagen generada a partir de la Figura 33b. Fuente: Elaboración propia*

Figura 35: *Imágenes producidas al aplicar el filtro Butterworth con* $\gamma_l = 0.8$,$\gamma_h = 1$, $r_c = 30$,$n = 3$ *.(a) Imagen generada a partir de la Figura 33a. (b) Imagen generada a partir de la Figura 33b. Fuente: Elaboración propia*

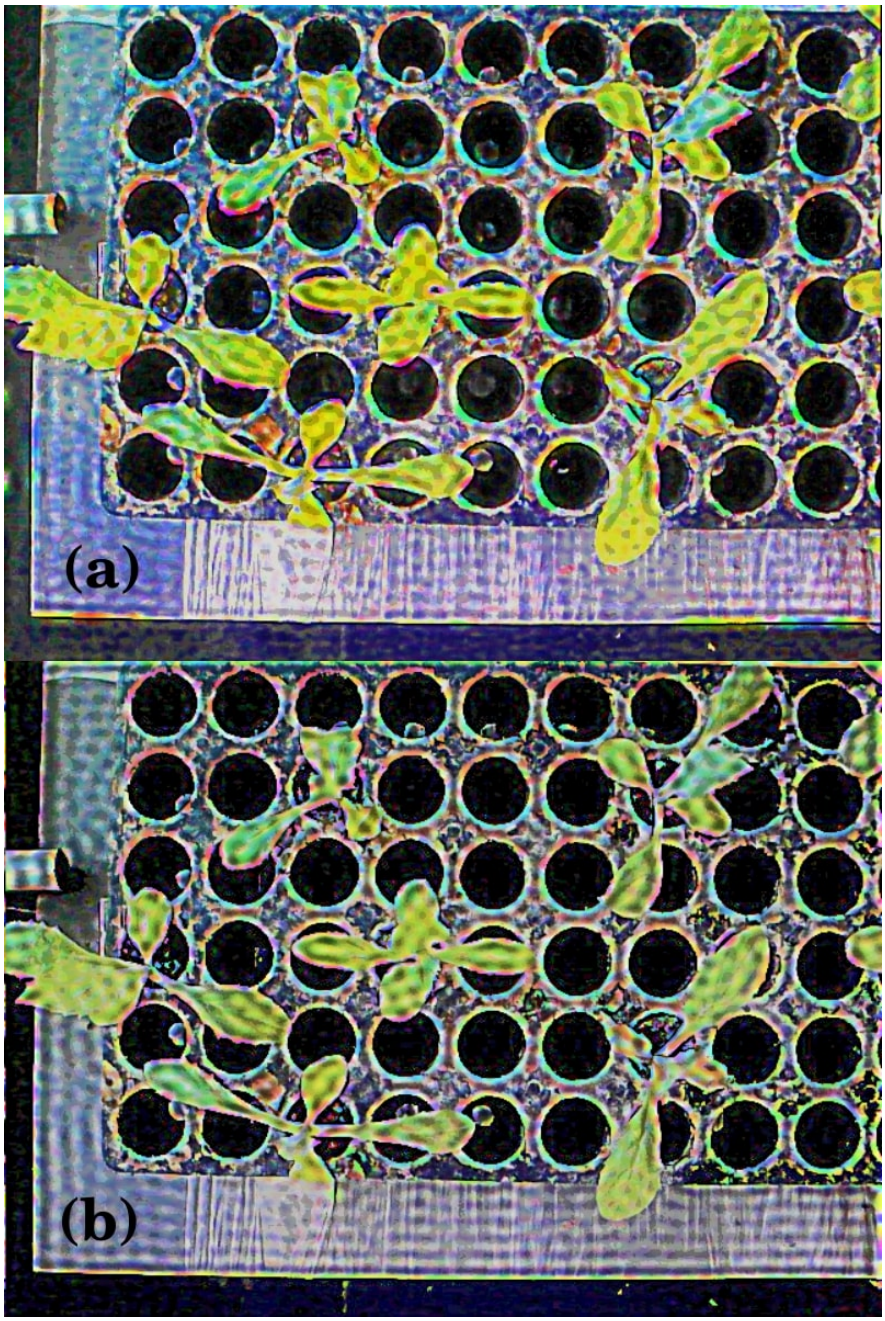


Figura 36: *Imágenes producidas al aplicar el filtro Ideal con* $\gamma_l = 0.4$,$\gamma_h = 1$, $r_c = 40$ *.(a) Imagen generada a partir de la Figura 33a. (b) Imagen generada a partir de la Figura 33b. Fuente: Elaboración propia*

$$B(u,v) = \begin{cases} 1; & H_{min} \leq H(u,v) \leq H_{max}, S_{min} \leq S(u,v) \leq S_{max}, V_{min} \leq V(u,v) \leq V_{max} \\ 0; & otro\ caso \end{cases} \tag{51}$$

- En el caso del análisis de formas (i.e. contornos) se optó por utilizar a los descriptores de Fourier, los cuales ya fueron definidos en el Capítulo II. Sin embargo como se menciona en Gonzalez y Woods (2001, pp.655 - 659) y Zhang y Lu (2001), al utilizar los descriptores de Fourier tal como se definen en (24) y (25) para comparar la similitud entre dos contornos, esta comparación puede resultar errónea debido a que los descriptores de Fourier son afectados por operaciones de rotación, escala o punto inicial del contorno. Es por ello que se optó por utilizar otro tipo de descriptores de Fourier (Zhang & Lu, 2001), los cuales se definen en (53).

$$r_k = \left[\left(u(k) - u_c \right)^2 + \left(v(k) - v_c \right)^2 \right]^{\frac{1}{2}} \tag{52}$$

$$a(t) = \frac{1}{K} \sum_{k=0}^{K-1} r(k) e^{\frac{-j2\pi tk}{K}} \tag{53}$$

Donde $[u(k), v(k)]$ son las tuplas que definen la ubicación de los pixeles que conforman un contorno compuesto por K pixeles. En la ecuación (52) se utilizan u_c y v_c para representar la ubicación del centroide del contorno. Por otro lado, $r(k)$ representa la distancia que existe entre el k-ésimo punto del contorno respecto al centroide del contorno.

Para validar el uso de los descriptores de Fourier se escogió una silueta modelo (i.e. *template*) y se calculó sus descriptores de acuerdo a (53). Luego se compararon los descriptores del modelo ($a_{modelo}(t)$) con los de otras siluetas ($a_{test}(t)$) utilizando el valor del error RMS tal como se define en (54).

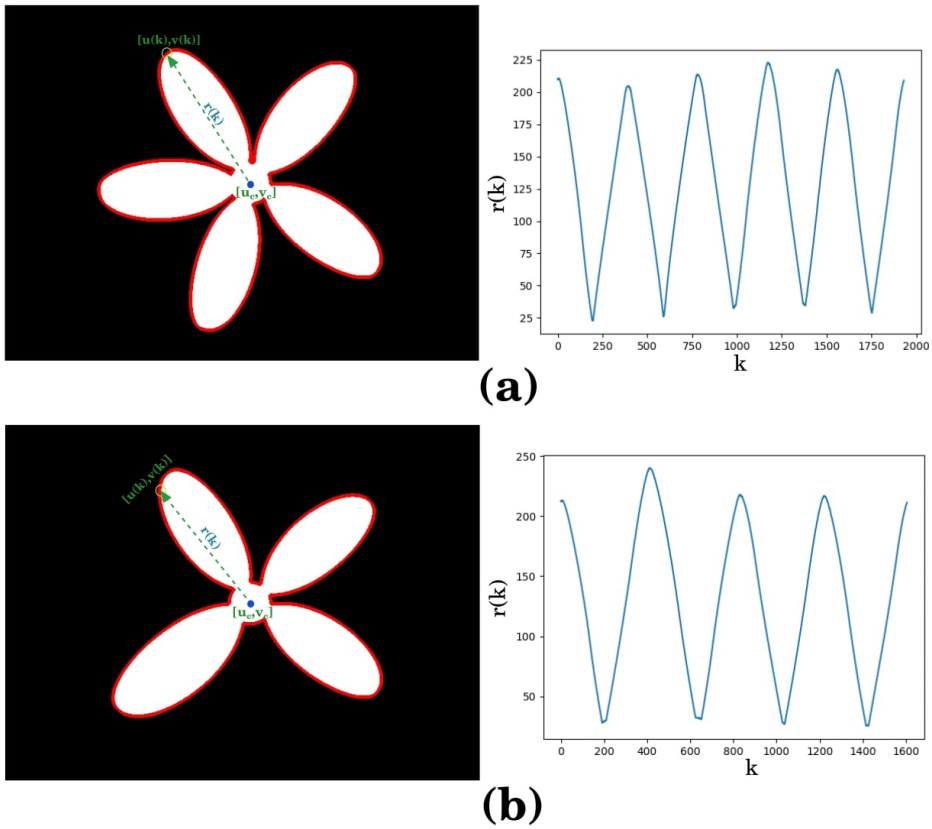


Figura 37: *Representación de contornos y la gráfica de su función $r(k)$:(a) Contorno y $r(k)$ de la silueta de una planta de 5 hojas; (b) Contorno y $r(k)$ de la silueta de una planta de 4 hojas. Fuente: Elaboración propia*

$$error_{RMS} = \sqrt{\frac{1}{M}\sum_{t=1}^{M}\Big[a_{modelo}(t) - a_{test}(t)\Big]^2} \tag{54}$$

En la Figura 38 pueden observarse todas las siluetas utilizadas para validar esta técnica. Se utilizaron los descriptores de Fourier de las siluetas de la Figura 37 como modelos y se compararon con los descriptores de todas las siluetas presentes en la Figura 38. La comparación se realizó utilizando la ecuación (54) y de acuerdo a los resultados se agregó un contorno de color azul a todas las siluetas que demostraron una alta similitud con la silueta de la Figura 37a ("Planta de 5 hojas"), un contorno verde a todas las siluetas que demostraron una alta similitud con la silueta de la Figura 37b ("Planta de 4 hojas") y un contorno rojo a todas las siluetas con un alto grado de disimilitud a ambas siluetas modelo ("Desconocido"). Adicionalmente, en la Tabla 6 se resumen los

resultados de las comparaciones de silueta en términos del error rms.

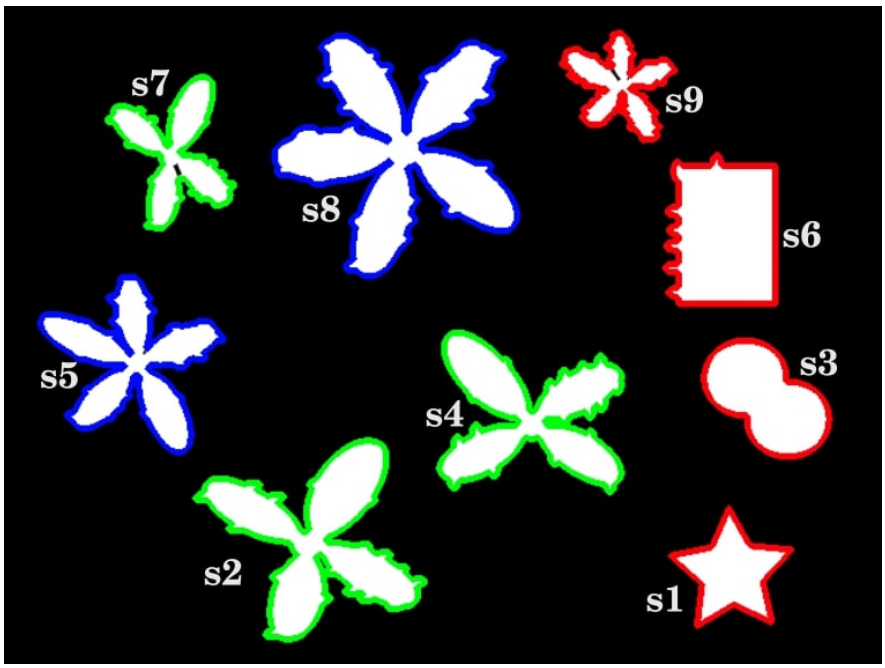


Figura 38: *Representación gráfica de los resultados de aplicar descriptores de Fourier. Fuente: Elaboración propia*

- Una vez seleccionado y validado el método de análisis y comparación de formas, se procedió a generar una base de datos con los descriptores de Fourier de múltiples plantines. Para realizar esto, primero se realizaron capturas de imágenes (utilizando los parámetros de captura ya definidos anteriormente) de plantines colocados en bandejas, dentro del espacio de trabajo del robot. Luego se preprocesaron y segmentaron las imágenes, de acuerdo a lo expuesto anteriormente, para obtener imágenes binarizadas, las cuales luego se procesaron para obtener sus contornos y finalmente sus descriptores de Fourier. Estos descriptores se guardaron como archivos tipo **.npy* para posteriormente poder acceder ellos.

Silueta	Error rms respecto a silueta de Figura 37a	Error rms respecto a silueta de Figura 37b	Clasificación
s1	2.11	4.28	"Desconocido"
s2	2.24	0.41	"Planta de 4 hojas"
s3	17.01	16.51	"Desconocido"
s4	2.23	0.29	"Planta de 4 hojas"
s5	0.41	2.22	"Planta de 5 hojas"
s6	1.86	1.34	"Desconocido"
s7	2.09	0.71	"Planta de 4 hojas"
s8	0.25	2.24	"Planta de 5 hojas"
s9	1.30	1.50	"Desconocido"

Tabla 6: *Resultados de la comparación de siluetas por descriptores de Fourier.Fuente: Elaboración Propia*

- Para llevar a cabo la clasificación, se realizaron dos tareas. La primera fue identificar si la región u objeto obtenido después de la segmentación, correspondía a un plantín o si solo se trata de regiones erróneas producidas durante la segmentación. La segunda tarea fue la de clasificar solo aquellas regiones que fueron identificadas como plantines en la tarea anterior, utilizando como descriptores el radio de la mínima circunferencia de la región y la cantidad de pixeles que conforman la región (i.e. plantín en la imagen binarizada).

- Para la generación de trayectorias se consideraron tres partes: primero la localización de un plantín clasificado dentro de la imagen digital, luego la localización del plantín dentro del espacio de trabajo del robot y finalmente un algoritmo que asegure que la herramienta de transplante no colisione con la bandeja (*coordinate matching*). Para llevar a cabo la primera parte, solo se requiere obtener el centroide (de acuerdo a las ecuaciones (30) y (31)) de la región que representa un plantín dentro de la imagen segmentada. Calculado el centroide de todos las regiones que representan plantines, se

procedió a convertir las coordenadas del centroide (C_u, C_v), de tuplas fila-columna en la imagen digital a coordenadas x_c, y_c en milímetros respecto al sistema de referencia de la cámara. Para esto se utilizaron los parámetros intrínsecos de la cámara (obtenidos en la etapa de calibración) y la distancia plantín-cámara z_c, tal como se define en (55). Luego se procedió a convertir estas coordenadas respecto al sistema de referencia de la cámara $[x_c, y_c, z_c]$ en coordenadas respecto al sistema de referencia del robot cartesiano $[x_r, y_r, z_r]$ utilizando la matriz de parámetros extrínsecos (matriz de rotación $\mathbf{R}$ y vector de traslación $\mathbf{t}$) y la posición actual del efector final del robot $[R_x, R_y, R_z]$, de acuerdo a (56).

$$\mathbf{x_c} = \begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix} = \begin{bmatrix} z_c\left(\frac{C_v - v_c}{f_x s_x}\right) \\ z_c\left(\frac{C_u - u_c}{f_y s_y}\right) \\ z_c \end{bmatrix} \tag{55}$$

$$\mathbf{x_r} = \begin{bmatrix} x_r \\ y_r \\ z_r \end{bmatrix} = \begin{bmatrix} R_x \\ R_y \\ R_z \end{bmatrix} + \mathbf{R}^{-1}(\mathbf{x_c} - \mathbf{t}) \tag{56}$$

Finalmente, para evitar que la herramienta (i.e *gripper*) montada en el efector final, colisione con la bandeja en la que se encuentran los plantines, se desarrolló un algoritmo que permite encontrar el hoyo de la bandeja más cercano a una determinada ubicación $\mathbf{x_r}$. Para esto, las ubicaciones (x, y) del centro de cada hoyo de la bandeja fueron determinadas de antemano y se dispusieron en una matriz $\mathbf{Bn}$ de dimensiones $\mathbb{R}^{6\times12\times2}$ (esto debido a que la bandeja cuenta con 72 hoyos).

La ubicación del hoyo fue determinado utilizando la ecuación (57), en la cual se buscó el elemento de $\mathbf{Bn}$ que minimice la expresión (57).

$$||\mathbf{x_r} - \mathbf{Bn}(i,j)|| \longrightarrow min;\ \forall 1 \leq i \leq 6,\ 1 \leq j \leq 12 \tag{57}$$

3.2.4. Variables de estudio y definición operacional

Variable Independiente	**Definición conceptual**	**Definición operacional**	**Indicadores**	**Unidades**	**Instrumentos de investigación**
Algoritmos de visión por computador de bajo y mediano nivel	Conjunto de algoritmos que permiten conocer ciertas características presentes en una escena del mundo real representada en una imagen digital.	Se establecieron los algoritmos de visión por computador y se estimará su performance en términos de: tiempo computacional y exactitud	Tiempo computacional	Segundos (s)	Guía de observación
			Exactitud	Porcentaje (%)	
El modelo de cámara digital	Conjunto de definiciones matemáticas que describen el proceso de proyección de una escena del mundo real sobre el sensor de una cámara digital.	Mediante la captura fotográfica de patrones conocidos se establecieron los parámetros intrínsecos y extrínsecos que conforman el modelo de cámara.	Parámetros intrínsecos	—	Guia de observación
			Parámetros extrínsecos	—	

Tabla 7: *Variables independientes y operacionalización. Fuente: Elaboración Propia*

Variable Dependiente	Definición conceptual	Definición operacional	Indicadores	Unidades	Instrumentos de investigación
Movimientos del robot cartesiano para llevar a cabo el proceso de repique de plantines.	Conjunto de trayectorias que realiza robot cartesiano para llevar a cabo el traslado de una bandeja de plantines no clasificada a otra clasificada.	Se evaluará la validez de las trayectorias generadas por el sistema de visión por computador y llevadas a cabo por el robot así como el tiempo total de repique.	Tiempo de ejecución de repique	Segundos (s)	Guía de observación
			Exactitud de las trayectorias	Porcentaje (%)	

Tabla 8: *Variable dependiente y operacionalización. Fuente: Elaboración Propia*

3.2.5. Técnicas e instrumentos de recolección de datos

3.2.5.1. Técnicas

- **Revisión bibliográfica:** Esta técnica comprende la revisión de documentos que contengan la descripción de: el funcionamiento de cada algoritmo de visión por computador utilizado, los métodos utilizados para calibrar una cámara digital y los comandos necesarios para generar movimientos al robot cartesiano. El producto deesta revisión bibliográfica es un resumen bibliográfico, el cual sirvió para diseñar e implementar la secuencia de repique.

- **Observación directa:** Esta técnica comprende la documentación del comportamiento del robot cartesiano con el sistema de visión por computador integrado. Mediante esta técnica se pretende obtener indicadores que permitan estimar el performance del sistema propuesto en términos de tiempos de ejecución y exactitud de las trayectorias generadas al robot.

3.2.5.2. Instrumentos

- **Compendio de las librerías utilizadas:** Aquí se resumieron las clases y funciones propias de las librerías del software a utilizar. Para la presente investigación se utilizó la librería de visión por computador OpenCV (Tabla 10) para Python3, la librería NumPy (Tabla 11) y la librería farmware-tools (Tabla 9) proveída por el fabricante Farmbot Inc. para manipular el robot cartesiano.

Función	**Variables de entrada**	**Variables de salida**	**Descripción**
home()	Ejes	–	Realiza la acción de *homming* en los ejes especificados
move_absolute()	posición, velocidad, offset	–	Realiza un movimiento absoluto a una velocidad determinada
read_pin()	número de pin , tipo de pin	valor del pin	Permite leer el valor de un pin del microcontrolador ya sea digital o analógico
set_pin_io_mode()	numero de pin, tipo de pin	–	Permite configurar un pin del microcontrolador como digital o analógico.
wait()	número de milise-gundos	–	Permite generar retardo temporal.
write_pin()	numero de pin, tipo de pin, valor	–	Permite modificar el valor de un pin.
get_current_position()	ejes	posición en los ejes ingresados	Permite conocer la ubicación del robot.

Tabla 9: *Funciones/Clases de la librería farmware-tools utilizados. Fuente: Elaboración propia*

Función / Clase	**Variables de entrada**	**Variables de salida**	**Descripción**
VideoCapture()	Número de cámara USB conectada	Imagen capturada en formato matriz de Numpy	Permite realizar una captura fotográfica
imwrite()	Título de la imagen, nombre de la Permite guardar una imagen	–	Permite guardar una imagen
cvtColor()	Imagen a convertir, Código de conversión	Imagen con espacio de color convertido	Permite convertir el espacio de color de una imagen
getStructuringElement()	Forma del kernel, tamaño del kernel	Kernel generado	Permite obtener el kernel o núcleo necesario para realizar las operaciones morfológicas (e.g. dilatación, erosión)
inRange()	Imagen a procesar, rangos de color	Imagen binarizada	Permite binarizar una imagen de acuerdo a un rango de color
morphologyEx()	Imagen binarizada de entrada, tipo de operación morfológica, número de iteraciones	Imagen binarizada de salida	Permite realizar una operación morfológica a una imagen binarizada.
bitwise_and()	Imagen a color, máscara	Imagen a color enmascarada	Permite realizar la operación de enmascaramiento a una imagen.

Tabla 10: *Funciones/Clases de la librería OpenCV utilizados. Fuente: Elaboración propia*

Función / Clase	Variables de entrada	Variables de salida	Descripción
fft()	Arreglo de una dimensión	FFT del arreglo	Permite calcular la FFT de un arreglo
fft2()	Arreglo bidimensional	FFT bidimensional	Permite calcular la FFT bidimensional de un arreglo bidimensional
fftshift()	FFT de un arreglo	FFT desplazado	Desplaza el componente cero en frecuencia al centro del espectro
ifft2()	FFT de un arreglo	Arreglo bidimensional	Permite calcular la inversa de una FFT bidimensional.
ifftshift()	FFT de un arreglo	FFT desplazado	Realiza la operación inversa a fftshift()

Tabla 11: *Funciones/Clases de la librería Numpy utilizados. Fuente: Elaboración propia*

- **Guía de observación:** Mediante este instrumento (ANEXO 01) se midió el tiempo computacional que aporta cada algoritmo utilizado, el tiempo total de procesamiento de la imagen, el tiempo de ejecución del repique y el número de aciertos y desaciertos al ejecutar el repique.

3.2.6. Técnicas de procesamiento y análisis de datos

3.2.6.1. Técnicas de procesamiento de datos

- **Matriz de confusión:** Esta técnica utilizó los datos obtenidos por medio de la guía de observación (ANEXO 01). Específicamente los verdaderos positivos, falsos positivos, verdaderos negativos y falsos negativos propios del clasificador. Se utilizó una matriz de confusión multiclase para un evaluar el performance del clasificador de tres clases de plantines a una población de 123 plantines.

		VALORES REALES		
		Clase A	Clase B	Clase C
VALORES PREDICHOS	Clase A	157	4	2
	Clase B	16	43	3
	Clase C	2	3	20

Tabla 12: *Matriz de confusión multiclase utilizando el clasificador con una población de 250 plantines.Fuente: Elaboración propia*

En la Figura 39 se hace mención a los siguientes términos:

- TPA: Verdaderos positivos de clase A
- TNA: Verdaderos negativos de clase A
- FPA: Falsos positivos de clase A
- FNA: Falsos negativos de clase A
- TPB: Verdaderos positivos de clase B
- TNB: Verdaderos negativos de clase B
- FPB: Falsos positivos de clase B
- FNB: Falsos negativos de clase B
- TPC: Verdaderos positivos de clase C
- TNC: Verdaderos negativos de clase C
- FPC: Falsos positivos de clase C
- FNC: Falsos negativos de clase C

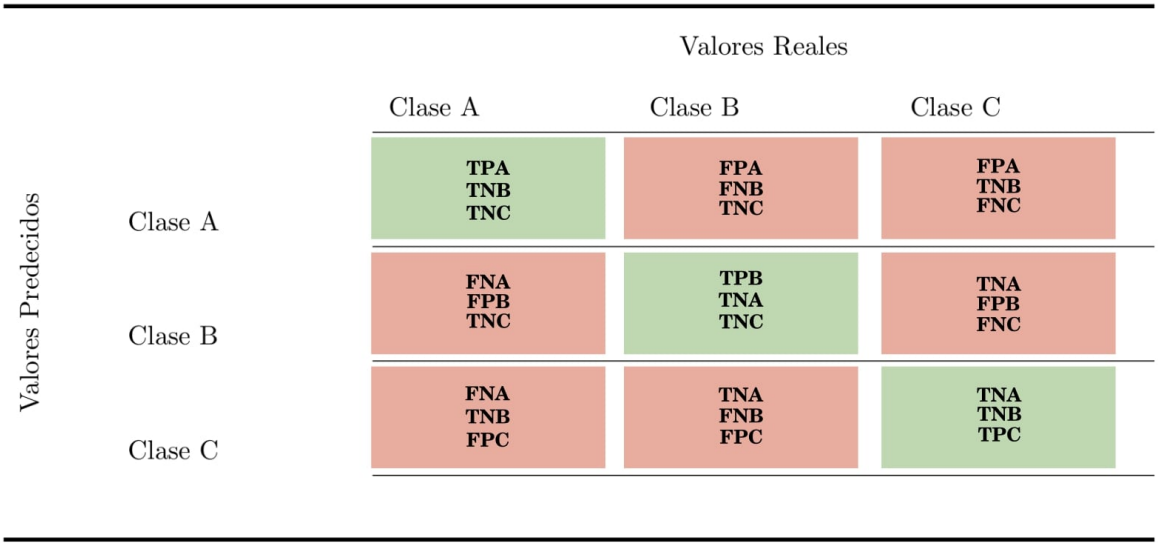


Figura 39: *Especificación de elementos dentro de la matriz de confusión multiclase Fuente: Elaboración propia*

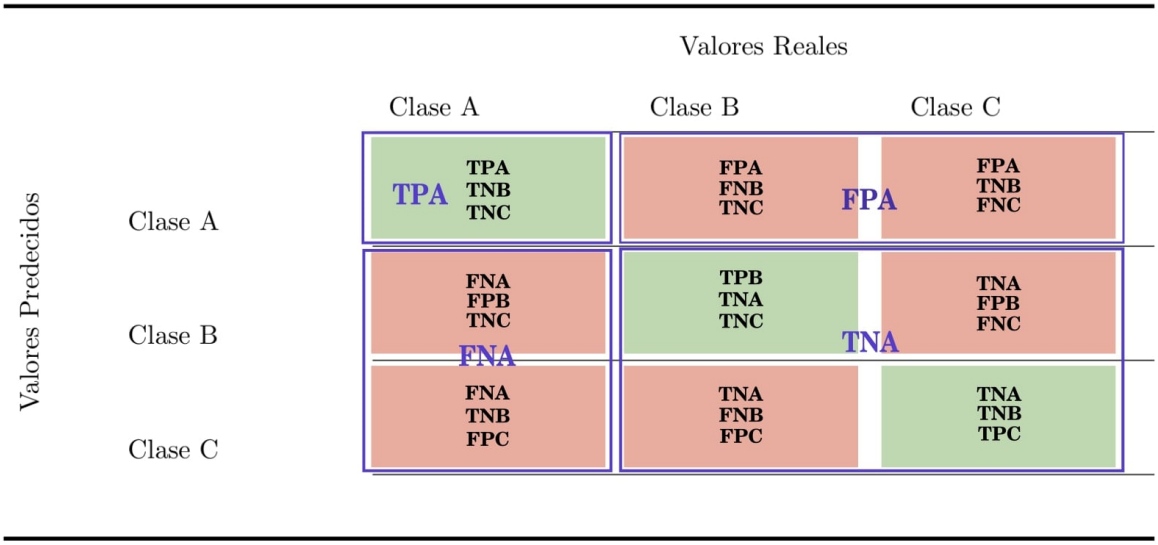


Figura 40: *Agrupación de elementos dentro de la matriz de confusión multiclase para el análisis de la clase A. Fuente: Elaboración propia*

Cabe resaltar que los elementos (i.e. celdas) que componen la matriz de confusión describen los desaciertos y aciertos o performance del clasificador para cada clase. Por ejemplo, si se desea analizar el performance del clasificador para la "clase A" es preciso analizar la matriz de confusión tal como se observa en la Figura 40, donde puede observarse que los TPA, FPA, FNA y TNA han sido agrupados.

- **Tablas estadísticas:** Aquí se indicaron los valores de tiempo computacional por algoritmo, tiempo computacional total y tiempo de ejecución de repique, obtenidos mediante el uso de la guía de observación (ANEXO 01).

3.2.6.2. Técnicas de análisis de datos

- **Análisis de la matriz de confusión:** Consistió en obtener los valores de exactitud, precisión, sensibilidad (recall), especificidad y score-F1 de la matriz de confusión. Estos valores o métricas permitieron evaluar el performance del clasificador desarrollado para el repique de plantines para cada clase. Las métricas se definen a continuación:

$$\text{Exactitud (Clase } X) = \frac{\text{TP}X + \text{TN}X}{\text{TP}X + \text{TN}X + \text{FP}X + \text{FN}X} \tag{58}$$

$$\text{Precisión (Clase } X) = \frac{\text{TP}X}{\text{TP}X + \text{FP}X} \tag{59}$$

$$\text{Sensibilidad (Clase } X) = \frac{\text{TP}X}{\text{TP}X + \text{FN}X} \tag{60}$$

$$\text{Especificidad (Clase } X) = \frac{\text{TN}X}{\text{TN}X + \text{FP}X} \tag{61}$$

$$\text{F-1 Score (Clase } X) = \frac{2 \times \text{Especificidad (Clase } X) \times \text{Precisión (Clase } X)}{\text{Especificidad (Clase } X) + \text{Precisión (Clase } X)} \tag{62}$$

Donde TPX significa verdadero positivo de la clase X (un plantín fue clasificado correctamente a la clase X), TNX significa verdadero negativo (un plantín fue excluido correctamente de la clase X), FPX significa falso positivo (un plantín fue clasificado incorrectamente a la clase X) y FNX significa falso negativo (un plantín fue excluido

incorrectamente de la clase X). Además es preciso saber que significa cada métrica. La exactitud nos da un indicador del porcentaje de predicciones acertadas respecto al total de predicciones. La precisión nos da un indicador del porcentaje de predicciones positivas acertadas respecto al total de predicciones positivas (se dice que una predicción fue positiva si el clasificador asignó a un plantín una determinada clase, ya sea de manera acertada o errónea). La sensibilidad nos da un indicador de cuan bueno es el clasificador al predecir correctamente la calidad (i.e. clase) de un plantín de una determinada clase entre todos los plantines de dicha clase. La especificidad es un indicador de cuan bueno es el clasificador al predecir que un plantín no pertenece a una determinada clase entre todos los plantines que no pertenecen a dicha clase. El score-F1 nos da una combinación de los resultados de precisión y sensibilidad, de manera que es útil cuando la población de plantines es desbalanceada (existen más plantines de una determinada clase que de otra).

- **Gráficas de resultados:** La información obtenida mediante las tablas estadísticas se dispusieron en gráficas, para facilitar su interpretación. En estas gráficas se evaluó el comportamiento del sistema robótico (tiempo de procesamiento, tiempo de repique y exactitud de posicionamiento) al ejecutarse la secuencia principal de repique en 25 ocasiones.

Capítulo IV

RESULTADOS

Como se mencionó en el primer capítulo, el objetivo general de esta investigación involucró el desarrollo de un clasificador de plantines a partir imágenes a color, la localización de los plantines dentro del espacio de trabajo del robot cartesiano y finalmente la generación de trayectorias para llevar a cabo la tarea de repique de plantines. Es por ello que se organizaron los resultados obtenidos durante cada etapa, de manera que se logre verificar el logro de cada uno de los objetivos propuestos.

- **Resultados correspondientes al desarrollo del clasificador de plantines.**

  El clasificador desarrollado está compuesto por dos sub-bloques: el primero es un identificador de plantines (permite discriminar objetos diferentes a un plantín) y el segundo es el clasificador de plantines como tal. De acuerdo a lo mencionado anteriormente, el clasificador toma como características al radio de la mínima circunferencia que ocupa el plantin dentro de la imagen y la cantidad de pixeles que representan al plantín dentro de la imagen, para determinar si el plantín es de clase A, B o C. Por otra parte es preciso resaltar que para llegar a clasificar los plantines a partir de una imagen a color, fueron necesarias 3 etapas previas: pre-procesamiento, procesamiento y clasificación.

  Para analizar la complejidad computacional de los algoritmos utilizados en las etapas

previas a la clasificación y los de la etapa de clasificación, se utilizaron los datos obtenidos en las guias de observación detalladas anteriormente para disponerlas en la Tabla 13. En la Tabla 13 se resume los resultados de tiempo computacional obtenidos después de ejecutar la secuencia de repique en 25 ocasiones, haciendo uso de 123 plantines. Adicionalmente en las Figuras 41, 42, 43 pueden visualizarse gráficas de Tiempo computacional vs. Algoritmo para las etapas de pre-procesamiento, procesamiento y clasificación respectivamente.

Etapa	**Algoritmo**	**Tiempo computacional (segundos)**			
		Valor mínimo	**Valor máximo**	**Promedio**	**Desviación(σ)**
Pre-procesamiento	Configuración de cámara	0.01745	0.01758	0.01753	0.00004
	Filtrado Homomórfico	1.79957	1.87634	1.80851	0.05510
Procesamiento	Segmentación por color	0.00550	0.00578	0.00555	0.00016
	Remoción de ruido	11.05401	11.65301	11.18693	0.12504
	Operaciones morfológicas	0.80007	0.82329	0.80987	0.00938
	Extracción de contornos	0.00875	0.00997	0.00981	0.00019
Clasificación	Detección de plantines	0.01318	0.02011	0.01412	0.00977
	Clasificación de plantines detectados	0.00595	0.00701	0.00607	0.00086

Tabla 13: *Resultados de tiempo computacional detallado para el clasificador de plantines. Fuente: Elaboración propia*

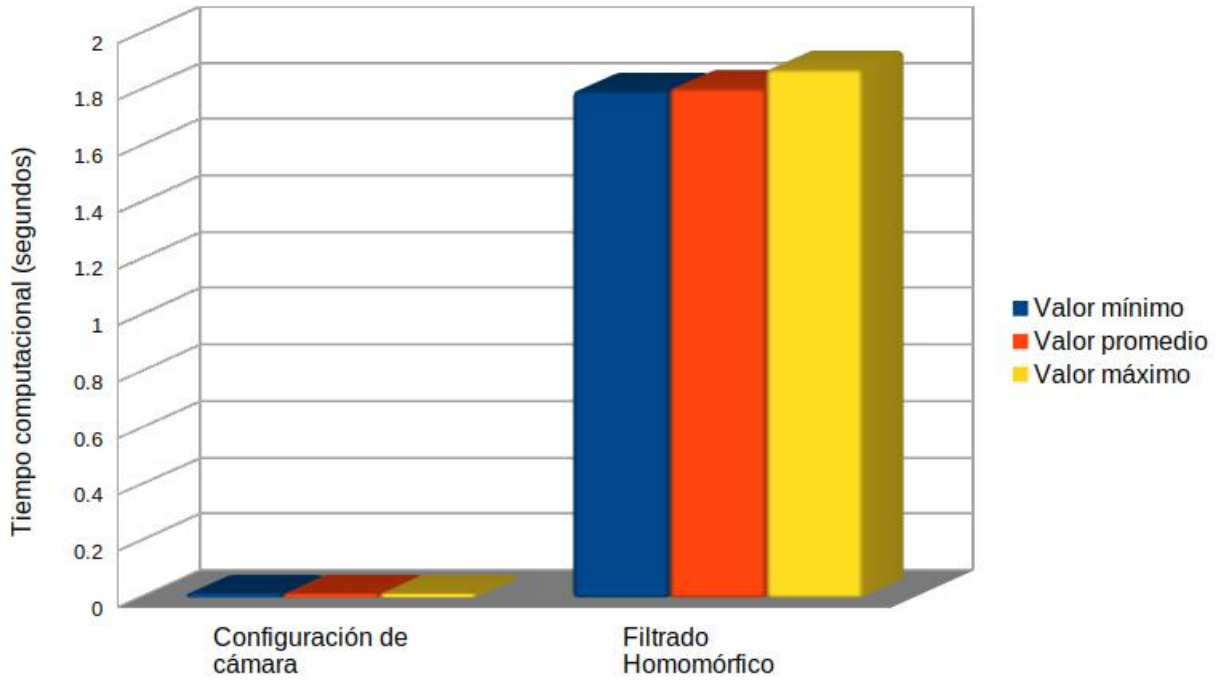


Figura 41: *Tiempo computacional de los algoritmos de la etapa de pre-procesamiento. Fuente: Elaboración propia*

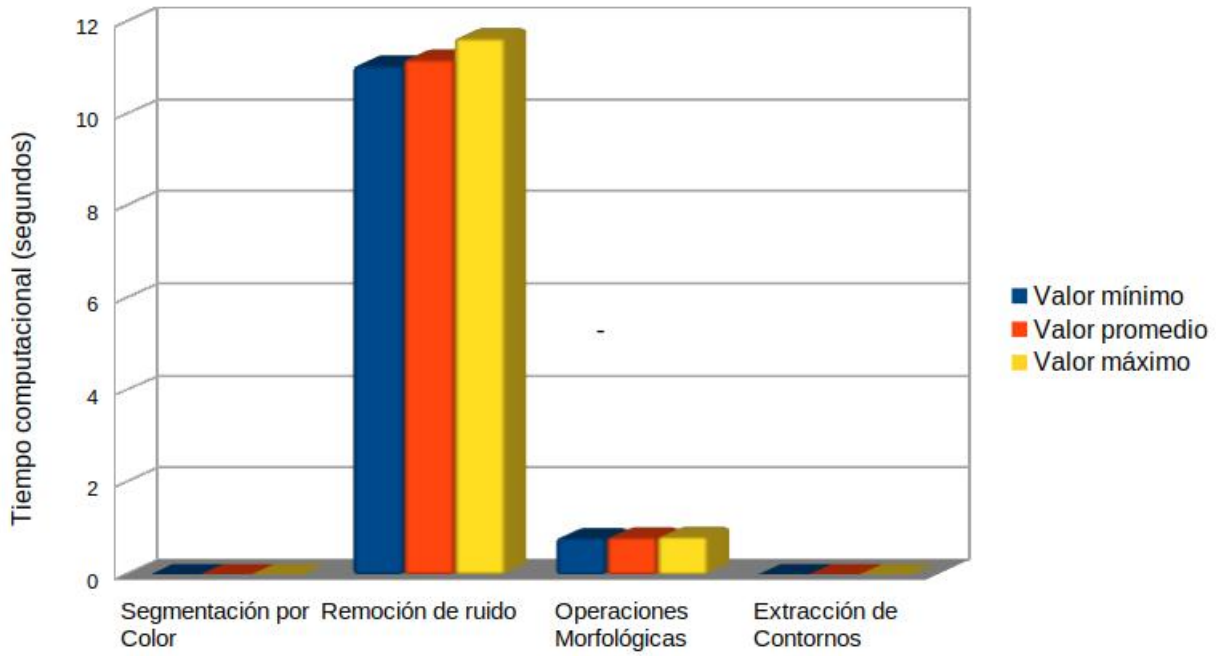


Figura 42: *Tiempo computacional de los algoritmos de procesamiento. Fuente: Elaboración propia*

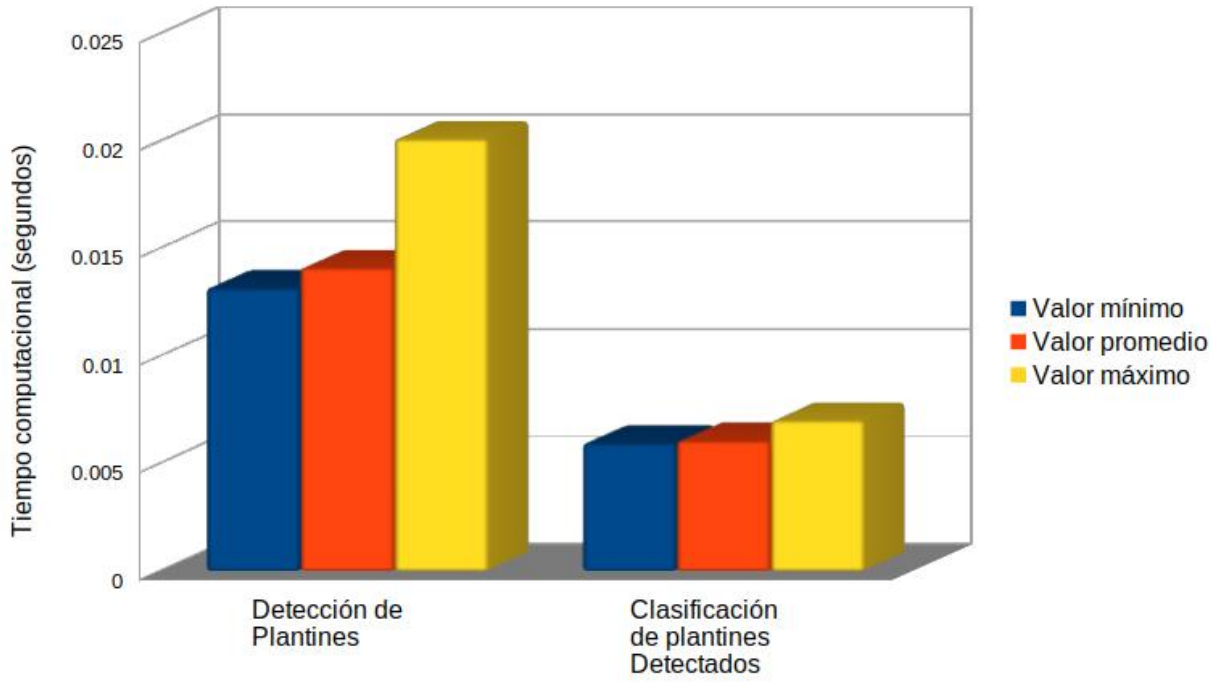


Figura 43: *Tiempo computacional de los algoritmos de la etapa de clasificación. Fuente: Elaboración propia*

Una de las etapas fundamentales para analizar los plantines presentes en una imagen a color es la segmentación de la imagen. En la Figura 44 se observa que si solamente se segmenta la imagen original (Figura 44a) y se utilizan operaciones morfológicas, el resultado es una imagen (Figura 44b) con una gran cantidad de regiones (i.e. agrupaciones de pixeles) que no corresponden a plantines y que por lo tanto pueden ser consideradas como ruido o regiones no deseadas. Sin embargo, al utilizar el algoritmo de remoción de ruido, cuya función es identificar y eliminar regiones pequeñas dentro de la imagen, se obtuvo una imagen con menor cantidad de estas regiones no deseadas (Figura 44c), en la cual solamente se pudo observar una región que no corresponda a plantines. Este algoritmo de remoción de ruido, a pesar de tener un alto tiempo computacional, pudo facilitar la siguiente tarea de extracción de contornos (Figura 44d).

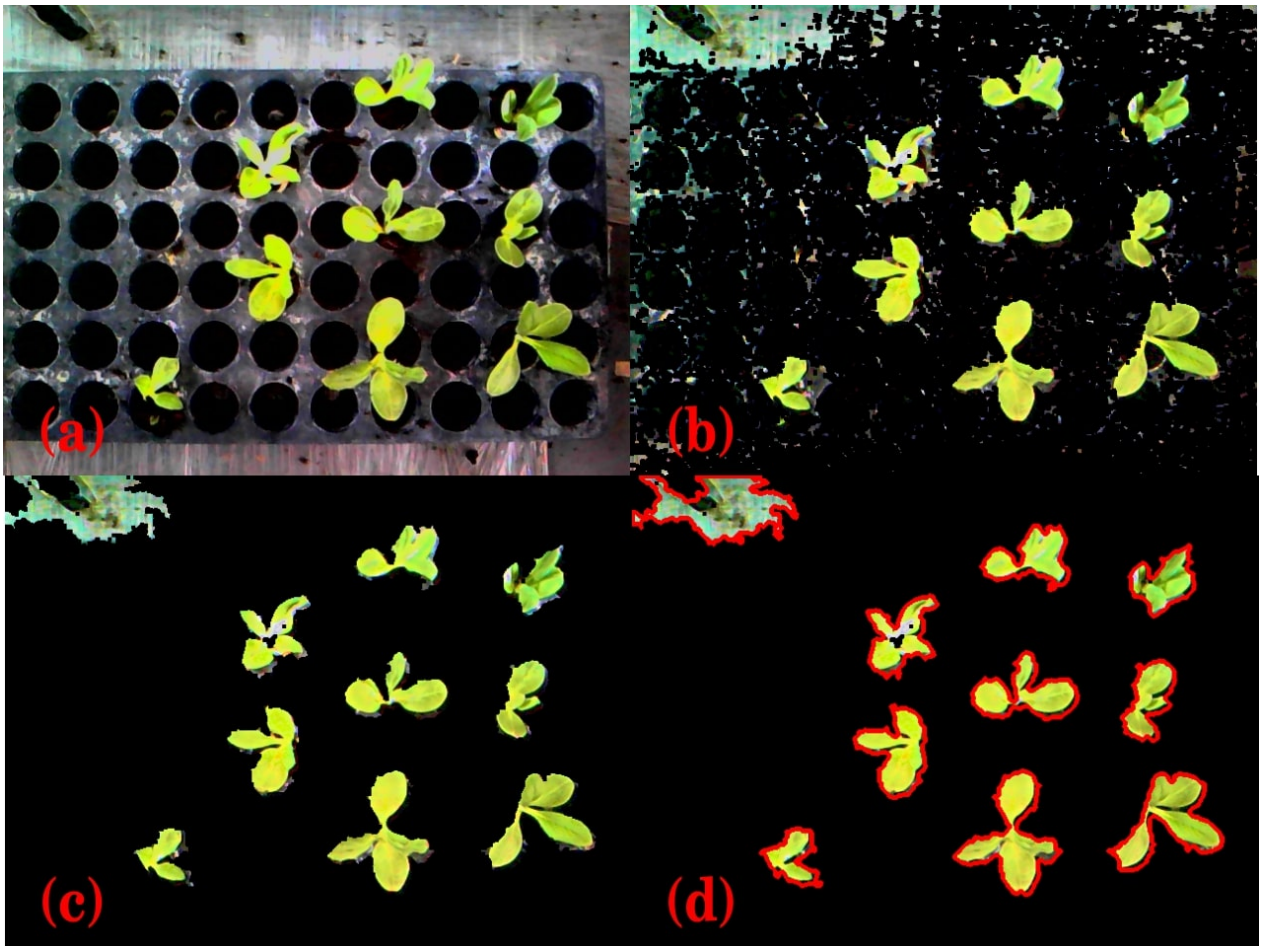


Figura 44: *Resultados de la segmentación y la importancia de la remoción de ruido: (a) Imagen original; (b) Imagen segmentada sin remoción de ruido; (c) Imagen segmentada con remoción de ruido; (d) Imagen segmentada y con extracción de contornos. Fuente: Elaboración propia*

Como parte del clasificador, el identificador de plantines permitió identificar que regiones segmentadas pertenecen a un plantín y cuales no, para posteriormente ser clasificados. Finalmente, para evaluar el performance del clasificador de acuerdo a su matriz de confusión (Tabla 12) se procedió a calcular las métricas de exactitud, precisión, sensibilidad, especificidad y score F-1 de acuerdo a las definiciones de (58), (59), (60), (61) y (62) respectivamente. Los resultados fueron tabulados en la Tabla 14.

Clase	Métricas del clasificador				
	Exactitud	**Precisión**	**Sensibilidad**	**Especificidad**	**F-1 Score**
Clase A	0.904	0.963	0.897	0.920	0.941
Clase B	0.896	0.694	0.860	0.905	0.785
Clase C	0.960	0.800	0.800	0.978	0.880

Tabla 14: *Resultados de las métricas del clásificador de plantines. Fuente: Elaboración propia*

- **Resultados correspondientes a la localización de plantines**

  La etapa fundamental para localizar plantines dentro del espacio de trabajo del robot (i.e. integrar el sistema de visión por computador al robot) y por lo tanto generar trayectorias, fue la calibración de cámara.

Parámetro	**Descripción**	**Valor**
fs_x	Distancia focal de x en pixeles	642.983
fs_θ	Oblicuidad o *skewness*	0.0
fs_y	Distancia focal de y en pixeles	640.326
u_c	Fila del centro de proyección	220.343
v_c	Columna del centro de proyección	296.406
k_0	Primer coeficiente de distorsión	0.057
k_1	Segundo coeficiente de distorsión	-0.435
k_2	Tercer coeficiente de distorsión	0.0017
k_3	Cuarto coeficiente de distorsión	-0.0011
k_4	Quinto coeficiente de distorsión	0.5077

Tabla 15: *Parámetros intrínsecos de la cámara utilizada. Fuente: Elaboración propia*

Parámetro	Descripción	Valor
r_{11}	Elemento de la matriz de rotación	0.002841
r_{12}	Elemento de la matriz de rotación	-0.99995
r_{13}	Elemento de la matriz de rotación	0.00882
r_{21}	Elemento de la matriz de rotación	-0.9999
r_{22}	Elemento de la matriz de rotación	-0.0028
r_{23}	Elemento de la matriz de rotación	0.00463
r_{31}	Elemento de la matriz de rotación	-0.0046
r_{32}	Elemento de la matriz de rotación	-0.00883
r_{33}	Elemento de la matriz de rotación	-0.9999
t_x	Traslación en X (en mm)	80.865
t_y	Traslación en Y (en mm)	29.358
t_z	Traslación en Z (en mm)	113.05542

Tabla 16: *Parámetros extrínsecos de la cámara respecto al efector final. Fuente: Elaboración propia*

En la Tabla 15 se especifican los parámetros intrínsecos obtenidos después de calibrar la cámara. De igual manera, en la Tabla 16 se especifican los parámetros extrínsecos respecto al efector final del robot (i.e. *gripper*). Adicionalmente, en la Tabla 17, se muestra la cantidad de plantines que fueron localizados correctamente dentro del espacio de trabajo del robot.

- **Resultados correspondientes al sistema durante el proceso de repique**

En la Tabla 17 se muestran los resultados de ejecutar el proceso de repique en 25 ocasiones. Como puede observarse, se obtuvo un porcentaje mínimo de identificación correcta de plantines de 33 % (en $N = 5$) y un máximo de 80 % (en $N = 9$). Así mismo, se

obtuvo un porcentaje mínimo de clasificación correcta de plantines de 0 % (en $N = 11$) y un máximo de 66.7 % (en $N = 20$). Por otro lado, el porcentaje de plantines transplantados (i.e. repicados) correctamente tuvo un valor mínimo de 0 % (en $N = 10$) y un máximo de 80 % (en $N = 3$), sin embargo en la mayoría de ocasiones dicho porcentaje no superó el 20 %.

Por otra parte, los resultados obtenidos correspondientes a los tiempos requeridos para la ejecución de cada etapa (Tabla 18) mostraron que el tiempo de procesamiento total (i.e. captura de imagen, pre-procesamiento de imagen, procesamiento de imagen, localización de plantines) tuvo un valor mínimo de 14.12 segundos y un valor máximo de 15.99 segundos. Así mismo, el tiempo de repique por plantín tuvo un valor mínimo de 24.13 segundos y un máximo 32.41 segundos.

En la Figura 45 se puede observar el proceso de repique, en el cual se utilizó la herramienta *gripper* (garra) para manipular los plantines. Como puede observarse, primero se posicionó al gripper sobre el plantin (Figuras 45a y 45b), luego se movieron las agujas del gripper (Figura 45c) de manera que estas agujas penetren el sustrato en el que está el plantín y por lo tanto este pueda ser asido por el gripper. Luego se levantó el plantín (Figura 45d) para poder ser traslado a su correspondiente bandeja de acuerdo a la clasificación (Figura 45e). Finalmente el plantín es dejado en su bandeja correspondiente (Figura 45f).

N	Plantines totales	Plantines Identificados	Plantines Clasificados correctamente	Plantines ubicados correctamente	Plantines transplantados correctamente
1	4	2	2	2	2
2	5	3	2	2	2
3	5	5	3	2	2
4	5	2	2	2	1
5	6	2	2	2	1
6	5	3	2	2	2
7	5	3	3	3	1
8	4	2	1	1	1
9	5	4	2	2	2
10	5	2	2	2	0
11	5	2	0	0	0
12	5	3	2	2	1
13	5	3	3	3	3
14	6	3	2	2	1
15	4	2	1	0	0
16	5	3	3	3	1
17	4	2	2	2	0
18	4	2	2	2	1
19	6	3	3	3	1
20	6	4	4	4	3
21	5	3	3	2	2
22	5	2	2	2	0
23	4	3	2	2	2
24	5	2	2	2	0
25	5	2	1	1	1

Tabla 17: *Resultados de identificación, clasificación,localización y transplante durante el proceso de repique. Fuente: Elaboración propia*

N	Tiempo de procesamiento total (s)	Tiempo de repique/plantín (s)			Tiempo de repique total (s)
		Min.	Max.	Medio	
1	14.59	24.84	26.7	25.75	119.38
2	15.87	27.16	**32.41**	30.2	227.35
3	15.84	26.09	32.03	28.57	246.10
4	14.67	27.04	28.3	27.67	101.67
5	14.44	25.46	26.92	26.19	96.37
6	15.32	**24.13**	25.64	24.46	105.26
7	14.87	25.87	26.22	26.1	126.32
8	14.21	25.63	26.42	26.07	95.87
9	14.22	25.13	26.37	25.78	248.64
10	14.28	26.87	27.18	26.92	170.87
11	14.63	24.99	25.93	25.43	164.53
12	15.99	23.98	24.51	24.15	210.63
13	**14.12**	26.31	27.61	25.77	215.23
14	14.27	25.88	26.81	25.97	218.53
15	14.38	25.74	26.97	25.87	95.13
16	14.53	24.65	26.03	25.74	215.63
17	14.87	24.82	25.84	25.47	90.36
18	15.96	25.46	26.82	25.87	113.86
19	**15.99**	25.18	26.07	25.81	210.88
20	14.38	24.22	25.31	24.75	**298.34**
21	14.44	25.33	26.94	25.55	119.54
22	15.87	24.57	26.08	25.12	97.68
23	15.63	24.15	25.09	24.87	123.45
24	15.45	25.87	26.84	25.91	**94.87**
25	15.97	26.99	27.52	27.31	109.49

Tabla 18: *Resultados de tiempos de procesamiento y transplante durante el proceso de repique. Fuente: Elaboración propia*

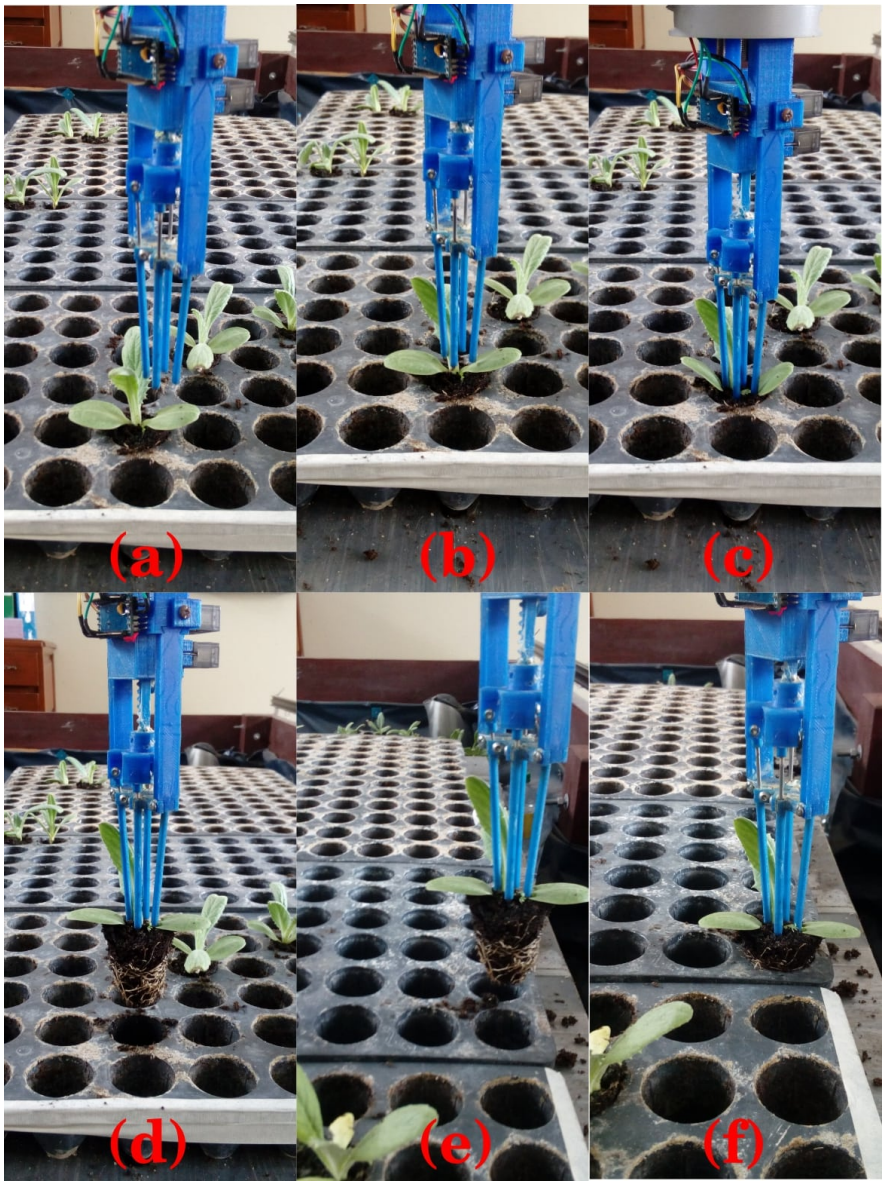


Figura 45: *Uso de la herramienta gripper durante el proceso de repique: (a) Gripper durante posicionamiento; (b) Gripper posicionado; (c) Gripper cogiendo plantín; (d) Plantín levantado; (e) Plantín trasladado a bandeja final; (f) Plantín repicado. Fuente: Elaboración propia*

Capítulo V

DISCUSIÓN DE RESULTADOS

- Para el desarrollo del clasificador de plantines se consideraron secuencias de algoritmos base, utilizadas en investigaciones relacionadas a sistemas de visión por computador aplicados a la agroindustria. Sin embargo, se agregaron algoritmos poco usuales (e.g. filtrado homomórfico, Remoción de ruido, Análisis de contornos) en investigaciones pasadas, con la finalidad de robustecer el sistema de visión por computador planteado, obteniendo mejores resultados.

- Los algoritmos de filtrado homomórfico y Remoción de ruido presentaron un elevado tiempo computacional en comparación con el resto de algoritmos utilizados. Esto se debió a que dichos algoritmos fueron implementados en su totalidad en lenguaje Python (i.e. no pertenecen a una librería). Sin embargo, ambos algoritmos robustecieron el sistema de visión por computador.

- A pesar de que en ciertas métricas obtenidas de la matriz de confusión del clasificador parecía que el clasificador tenía un mejor performance al identificar los plantines de clase C que al identificar plantines de clase A, la métrica F-1 score permitió determinar que en realidad el clasificador se desempeñaba mejor al identificar plantines de clase A. Este efecto es normal, debido a que la población está desbalanceada.

Capítulo VI

CONCLUSIONES

- Al término de la revisión bibliográfica se logró determinar que los parámetros morfológicos de un plantín (i.e. dimensiones) y el color del mismo, son relevantes durante la tarea de clasificación de plantines.
- Para implementar la secuencia base del sistema de visión por computador se desarrollaron etapas de: pre-procesamiento (configuración de cámara y filtrado homomórfico) y procesamiento (segmentación por rangos de color,remoción de ruido, operaciones morfológicas, extracción de contornos y análisis de contornos), siendo esta última etapa la que más tiempo computacional tiene (aproximadamente 12 segundos) respecto a la etapa de pre-procesamiento (aproximadamente 2 segundos).
- Se desarrolló un clasificador de plantines de 3 clases, obteniendo el mejor performance durante la clasificación de plantines de clase A (F-1 score = 0.941). En segundo lugar en performance está la clasificación de plantines de clase C (F-1 score = 0.880) y finalmente, la clasificación de plantines de clase B (F-1 score = 0.785).
- Se logró automatizar el proceso de repique de plantines de alcachofa al integrar el sistema de visión por computador (captura de imágenes , clasificador) con el robot cartesiano, mediante la utilización del modelo de cámara digital y de algoritmos de calibración de cámara.

Capítulo VII

RECOMENDACIONES

- Hasta el término de la presente investigación, los plantines se colocaron de forma separada dentro de la bandeja, con la finalidad de evitar problemas con la segmentación por rangos de color. Se sugiere robustecer el algoritmo de segmentación, utilizando algoritmos propios del aprendizaje automático (i.e. *machine learning*).

- Debido a que las imágenes RGB son proyecciones (en 2D) de un espacio en tres dimensiones, se recomienda utilizar cámaras RGB-D para la extracción de los parámetros morfológicos reales (i.e. no en pixeles, si no en milímetros).

- Se recomienda la mejora de la herramienta manipuladora de plantines (i.e. *gripper*), ya que el porcentaje de plantines transplantados correctamente (respecto a la cantidad de plantines clasificados correctamente) tuvo un valor de $41.93\,\% \pm 30.0\,\%$ debido principalmente a una inadecuada manipulación de plantines por parte del *gripper*.

REFERENCIAS BIBLIOGRÁFICAS

Agencia Peruana de Noticias. (2019, junio). "Perú registra los mejores rendimientos en agricultura intensiva a nivel mundial ". Recuperado desde https://www.americaeconomia.com/negocios-industrias/peru-registra-los-mejores-rendimientos-en-agricultura-intensiva-nivel-mundial

Alegre, E., Pajares, G. & De la Escalera, A. (2016). *Conceptos y métodos en visión por computador*. España: Comité español de automática.

Ali Ashraf, M., Kondo, N. & Shiigi, T. (2011). Use of Machine Vision to Sort Tomato Seedlings for Grafting Robot. *Engineering in Agriculture, Environment and Food*, *4*(4), 119-125. doi:10.1016/S1881-8366(11)80011-X

Autodesk Inc. (2014). *Fundamentals of CNC Machining*. USA.

Balza, J. (2019, octubre). "Tecnología de punta para preservar frutas y hortalizas de exportación." Friopacking cumple 10 años. Recuperado desde http://www.redagricola.com/pe/tecnologia-punta-preservar-frutas-hortalizas-exportacion/

Bolívar, F. (2012). *Módulo control numérico computarizado*. Boyacá-Colombia: Universidad Nacional Abierta y a Distancia CCAV NEIVA.

Burger, W. (2016, mayo). *Zhang's Camera Calibration Algorithm: In-Depth Tutorial and Implementation*. University of Applied Sciences Upper Austria. Austria.

Chirinos, O. (2019, enero). "Balance y nuevos retos del agro en el Perú". Recuperado desde https://www.inei.gob.pe/media/MenuRecursivo/publicaciones_digitales/Est/Lib1537/libro.pdf

Choque, C. J. (2018). *"Desarrollo de un robot cnc tipo cartesiano de la marca farmbot como soporte tecnológico para el proceso de control de calidad de plantines"* (Tesis de pregrado, Universidad Privada Antenor Orrego (UPAO), Trujillo-Perú). Recuperado desde http://repositorio.upao.edu.pe/handle/upaorep/5975

Damiani, O. (2000). "El Estado y la agricultura no tradicional de exportación en América Latina: Resultados y lecciones de tres estudios de casos".

De Clercq, M., Vats, A. & Biel, A. (2018, febrero). *Agriculture 4.0 – The Future Of Farming Technology.*

Edan, Y., Han, S. & Kondo, N. (2009). Automation in Agriculture. En Springer (Ed.), *Springer Handbook of Automation* (Cap. 63).

Eddins, S. (2013). Homomorphic filtering – part 1. Recuperado desde https://blogs.mathworks.com/steve/2013/06/25/homomorphic-filtering-part-1/

Encyclopedia Britannica. (2020). Computational complexity. Recuperado desde https://www.britannica.com/topic/computational-complexity

Faridi, H. & Aboonajmi, M. (2017). Application of machine vision in agricultural products. En *4th Iranian International NDT Conference (IRNDT).*

Farmbot Inc. (2017). Farmbot Genesis V1.3. Recuperado desde https://genesis.farm.bot/docs/

Flores, D. A., Saito, C., Paredes, J. A. & Trujillano, F. (2017). "Multispectral imaging of crops in the Peruvian Highlands through a fixed-wing UAV system". En *2017 IEEE International Conference on Mechatronics (ICM)* (pp. 399-403). doi:10.1109/ICMECH.2017.7921139

Gandini, L. (2012, abril). "Skilled-Worker Mobility and Development in Latin America and the Caribbean: Between Brain Drain and Brain Waste". *Journal of the Office of Latino/Latin-American Studies 2012.*

GLOBALG.A.P. (2020). Material de Propagación Vegetal - Mejores Prácticas para los Viveros (PPM). Recuperado desde https://www.globalgap.org/es/for-producers/ppm/

Gonzalez, R. C. & Woods, R. E. (2001). *Digital Image Processing* (2.ª ed.). USA: Addison-Wesley Longman Publishing Co.

Infomercado. (2018, julio). "Perú: Indicadores de Empleo e Ingreso por departamento." Recuperado desde https://www.inei.gob.pe/media/MenuRecursivo/publicaciones_digitales/Est/Lib1537/libro.pdf

Infomercado. (2020, enero). "Agricultura: Segunda actividad generadora de divisas en el Perú". Recuperado desde https://infomercado.pe/agricultura-segunda-actividad-generadora-de-divisas-en-el-peru/

Al-Kindi, G. & Zughaer, H. (2012). An Approach to Improved CNC Machining Using Vision-Based System. *Materials and Manufacturing Processes*, *27*(7), 765-774. doi:10.1080/10426914.2011.648249

Lazar, C., Panescu, D., Laczko, A. & Braescu, C. (2003). Vision-Based Integrated System for Robotic Assembly. *IFAC Proceedings Volumes*, *36*(23), 113-118. IFAC Workshop on Intelligent Assembly and Disassembly (IAD'03), Bucharest, Romania, 9-11 October 2003. doi:10.1016/S1474-6670(17)37671-1

Liao, Y. & Chuang, Y. (2009). Vegetable seedling sorting based on mean-shift. En *2009 Joint Conferences on Pervasive Computing (JCPC)* (pp. 191-196). doi:10.1109/JCPC.2009.5420192

Liu, S., Xing, Z., Wang, Z., Tian, S. & Jahun, F. (2017). Development of machine-vision system for gap inspection of muskmelon grafted seedlings. *PLoS One*, *12*. doi:10.1371/journal.pone.0189732doi

Matuska, S., Hudec, R. & Benco, M. (2012). The comparison of CPU time consumption for image processing algorithm in Matlab and OpenCV. En *2012 ELEKTRO* (pp. 75-78). doi:10.1109/ELEKTRO.2012.6225575

McCarthy, C., Hancock, N. & Raine, S. (2010). Applied machine vision of plants: A review with implications for field deployment in automated farming operations. *Intelligent Service Robotics*. doi:10.1007/s11370-010-0075-2

MINCETUR. (2009, agosto). *"Informe Final: Mejora de las técnicas y procesos en la producción, cosecha y acopio de la alcachofa, Lambayeque"*. MINCETUR. Lambayeque-Perú.

Myler, H. R., Weeks, A. R. & Voicu, L. (1995). RGB color enhancement using homomorphic filtering. En R. L. Stevenson & S. A. Rajala (Eds.), *Image and Video Processing III* (Vol. 2421, pp. 43-50). International Society for Optics y Photonics. SPIE. doi:10.1117/12.205491

Nerves Project. (2020). Nerves - Getting Started. Recuperado desde https://hexdocs.pm/nerves/getting-started.html

Oliva, M., Vacalla, F., Pérez, D. & Tucto, A. (2014). *Manual vivero forestal para producción de plantones de especies forestales nativas: Experiencia en Molinopampa, Amazonas – Perú.* Chachapoyas-Perú: Instituto de Investigación de la Amazonía Peruana (IIAP).

OpenCV. (2021). Camera Calibration. Recuperado desde https://docs.opencv.org/master/dc/dbb/tutorial_py_calibration.html

Overby, A. (2011). *CNC Machining Handbook: Building,Programming and Implementation* (M.-H. Inc., Ed.). McGraw-Hill Inc.

Patrício, D. I. & Rieder, R. (2018). Computer vision and artificial intelligence in precision agriculture for grain crops: A systematic review. *Computers and Electronics in Agriculture*, *153*, 69-81. doi:10.1016/j.compag.2018.08.001

Paunski, Y. K. & Angelov, G. T. (2019). Performance and power consumption analysis of low-cost single board computers in educational robotics. *IFAC-PapersOnLine*, *52*(25), 424-428. 19th IFAC Conference on Technology, Culture and International Stability TECIS 2019. doi:10.1016/j.ifacol.2019.12.575

Poudel, P. & Shirvaikar, M. (2010). Optimization of computer vision algorithms for real time platforms. En *2010 42nd Southeastern Symposium on System Theory (SSST)* (pp. 51-55). doi:10.1109/SSST.2010.5442803

Rui, W., Huawei, Q., Yuyang, S. & Jianliang, L. (2017). Calibration of Cartesian robot based on machine vision. En *2017 IEEE 3rd Information Technology and Mechatronics Engineering Conference (ITOEC)* (pp. 1103-1108). doi:10.1109/ITOEC.2017.8122525

Tangmongkhonsuk, J., Meena, A., Sasithong, P., Wijayasakra, S., Phanomchoeng, G., Phongphanphanee, C., ... Wuttisittikulkij, L. (2018). Development of a Software Program for Automatic Cartesian Farming Robot. En *2nd Intenational Conference on Advanced Information Technologies(ICAIT)* (pp. 6-9).

Thompson, B. (1985). "Seedlings morphological evaluation– What you can tell by looking". En *"Evaluating Seedling Quality: Principles, Procedures, and Predictive Abilities of Major Tests"* (Cap. 6).

Tian, H., Wang, T., Liu, Y., Qiao, X. & Li, Y. (2020). Computer vision technology in agricultural automation —A review. *Information Processing in Agriculture*, *7*(1), 1-19. doi:doi.org/10.1016/j.inpa.2019.09.006

Tong, J., Jiang, H. & Zhou, W. (2012). Development of automatic system for the seedling transplanter based on machine vision technology. En *2012 IEEE International Conference on Computer Science and Automation Engineering (CSAE)* (Vol. 2, pp. 742-746). doi:10.1109/CSAE.2012.6272873

Velasco, J. (2017, noviembre). "El avance de la automatización en la agricultura ": Tendencias tecnológicas en la industria frutícola. Recuperado desde http://www.redagricola.com/cl/el-avance-de-la-automatizacion-en-la-agricultura/

Velazco, J. [Jackeline] & Velazco, J. [Julia]. (2012). EMPLEO Y PROTECCIÓN SOCIAL. En C. Garavito & I. Muñoz (Eds.), *"Características del empleo agrícola en el Perú"*: (1.ª ed., Cap. 5, pp. 161-211). Fondo Editorial - Pontificia Universidad Católica del Perú.

Wang, R. (2013). Fourier Descriptor. Recuperado desde http://fourier.eng.hmc.edu/e161/lectures/fd/fd.html

Zhang, D. & Lu, G. (2001). A Comparison of Shape Retrieval Using Fourier Descriptors and Short-Time Fourier Descriptors. En H.-Y. Shum, M. Liao & S.-F. Chang (Eds.), *Advances in Multimedia Information Processing PCM 2001* (pp. 855-860). Springer Berlin Heidelberg. doi:10.1007/3-540-45453-5_111

ANEXOS

Anexo 01

Guía de observación utilizada durante la evaluación del sistema durante el proceso de repique

Universidad Privada Antenor Orrego

Vicerrectorado de Investigación

LABINM – Línea de investigación de robótica y automatización

"Guia de observación del robot cartesiano integrado al sistema de visión por computador durante el proceso de repique de plantines de alcachofa"

Fecha: 08/03/2021 Hora de observación: 10:35 Número de plantines a analizar: 6 ID: 14

I. Tiempo computacional por algoritmos

N	Nombre de algoritmo	Tiempo de ejecución (segundos)	Observaciones
1	Configuración de cámara	0.01754	
2	Filtrado Homomórfico	1.83634	
3	Segmentación por color	0.00563	
4	Remoción de ruido	11.1345	
5	Operaciones morfológicas	0.81473	
6	Extracciones de contornos	0.00984	
7	Detección de plantines	0.01517	
8	Clasificación de plantines detectados	0.00691	

Tiempo computacional total:

II. Conteo de plantines clasificados

N° de Plantín	Calidad real	Calidad de acuerdo al clasificador
1	B	B
2	B	B
3	B	B
4		
5		
6		

III. Performance del sistema durante el repique

N° de Plantín	Tiempo de repique	¿Transplantado correctamente?	Observaciones
1	25.97	No	Fallo de gripper
2	26.54	Si	
3	25.89	Si	
4			
5			
6			