

Escuela Superior Politécnica del Litoral

Facultad de Ingeniería en Electricidad y Computación

Calibración de sistema de adquisición de imágenes estereoscópicas para
detección de objetos en cinta transportadora

TECH-345

Proyecto Integrador

Previo a la obtención del Título de:

Ingeniero en Ciencias de la Computación

Presentador por:

Jorge Adrián Vulgarin Punguil

Guayaquil - Ecuador

Año: 2023 - 2024

Dedicatoria

Este proyecto se lo dedico a mis padres (Yolanda Punguil y Ramon Vulgarin) quienes con todo su esfuerzo y arduo trabajo me permitieron enfocarme en mis estudios durante toda mi vida, sin su ayuda no lo hubiese logrado.

Al Dr. Daniel Ochoa quien me brindo la oportunidad de formar parte del laboratorio del CTD y me introdujo al mundo de la investigación.

A mi familia y amigos que siempre me han apoyado.

Jorge Vulgarin

Agradecimiento

Agradezco a ESPOL, por darme la oportunidad de convertirme en ingeniero en Computación.

A mi familia y amigos por acompañarme todo el camino.

Jorge Vulgarin

Declaración Expresa

Yo Vulgarin Punguil Jorge Adrián acuerdo y reconozco que:

La titularidad de los derechos patrimoniales de autor (derechos de autor) del proyecto de graduación corresponderá al autor o autores, sin perjuicio de lo cual la ESPOL recibe en este acto una licencia gratuita de plazo indefinido para el uso no comercial y comercial de la obra con facultad de sublicenciar, incluyendo la autorización para su divulgación, así como para la creación y uso de obras derivadas. En el caso de usos comerciales se respetará el porcentaje de participación en beneficios que corresponda a favor del autor o autores. La titularidad total y exclusiva sobre los derechos patrimoniales de patente de invención, modelo de utilidad, diseño industrial, secreto industrial, software o información no divulgada que corresponda o pueda corresponder respecto de cualquier investigación, desarrollo tecnológico o invención realizada por mí durante el desarrollo del proyecto de graduación, pertenecerán de forma total, exclusiva e indivisible a la ESPOL, sin perjuicio del porcentaje que me corresponda de los beneficios económicos que la ESPOL reciba por la explotación de mi innovación, de ser el caso.

En los casos donde la Oficina de Transferencia de Resultados de Investigación (OTRI) de la ESPOL comunique al autor que existe una innovación potencialmente patentable sobre los resultados del proyecto de graduación, no se realizará publicación o divulgación alguna, sin la autorización expresa y previa de la ESPOL.

Guayaquil, 03 de febrero del 2024.

Jorge Vulgarin

Evaluadores

Ing. Ronald Criollo

Profesor de la materia

Ph.D. Daniel Ochoa

Tutor del proyecto

Resumen

La implementación de líneas de empaquetado automático requiere datos de profundidad, la estéreo-visión se presenta como la opción más costo-efectiva para obtener estos datos. Esta tecnología ofrece a las pequeñas y medianas empresas la posibilidad de acceder a soluciones de automatización, mejorando así su competitividad y fomentando su crecimiento. En este proyecto, se propone la implementación de un sistema de calibración estereoscópica para adquirir datos de profundidad en una cinta transportadora mediante estéreo-visión. Se empleó la cámara ELP Dual Lens 4mp, la librería OpenCV para la calibración estereoscópica y el algoritmo StereoSGBM para la generación de mapas de profundidad. Los resultados incluyen una calibración estereoscópica adecuada con un error inferior a 1px, la identificación de la combinación óptima de parámetros para el algoritmo StereoSGBM con un error por debajo de 3mm, y la determinación del tiempo de ejecución del algoritmo en función del tamaño de las imágenes estereoscópicas. Finalmente, se demuestra que el protocolo de calibración propuesto es adecuado para la cámara utilizada, y se logra la ejecución en tiempo real de la generación de mapas de profundidad. y se implementan interfaces gráficas que facilitan el uso de los sistemas de calibración y generación de mapas de profundidad propuesto.

Palabras claves: Estéreo-visión, Mapas de profundidad, calibración estereoscópica, StereoSGBM

Abstract

The implementation of automatic packaging lines requires depth data, and stereovision emerges as the most cost-effective option to obtain such information. This technology provides small and medium-sized enterprises with the opportunity to access automation solutions, thereby enhancing their competitiveness and fostering growth. In this project, the implementation of a stereoscopic calibration system is proposed to acquire depth data on a conveyor belt using stereovision. The ELP Dual Lens 4mp camera, the OpenCV library for stereoscopic calibration, and the StereoSGBM algorithm for depth map generation were utilized. The results include a proper stereoscopic calibration with an error of less than 1px, the identification of the optimal combination of parameters for the StereoSGBM algorithm with an error below 3mm, and the determination of the algorithm's runtime based on the size of stereoscopic images. Finally, it is demonstrated that the proposed calibration protocol is suitable for the stereovision system used, achieving real-time execution of depth map generation. Additionally, graphical interfaces are implemented to facilitate the use of the proposed calibration and depth map generation systems.

Key words: Stereo vision, depth maps, stereo calibration, StereoSGBM

Índice general

Resumen	i
Abstract	ii
Índice general	iii
Simbología	v
Índice de figuras	vi
Índice de tablas	vii
Capítulo 1	1
1 Introducción	2
1.1 Descripción del problema	2
1.2 Justificación del problema	2
1.3 Objetivos	3
1.3.1 Objetivos generales	3
1.3.2 Objetivos específicos	3
1.4 Marco teórico	3
1.4.1 Parámetros ópticos de cámara	3
1.4.2 Estéreo visión	5
1.4.3 Calibración de un sistema de estéreo visión	7
1.4.4 Mapas de disparidad	12
Capítulo 2	16
2 Metodología	17
2.1 Análisis	17
2.1.1 Requerimientos funcionales	17
2.1.2 Requerimientos no funcionales	17
2.1.3 Alcance de la solución	17

2.2	Sistema de estereo visión	18
2.3	Diseño de la solución	20
2.3.1	Configuración de parámetros	22
Capítulo 3	29
3	Resultados y Análisis	30
3.1	Resultados de calibración estereoscópica	30
3.2	Selección de parámetros óptimos para la generación de mapas de profundidad .	36
3.3	Tiempo de cómputo para la generación de mapa de profundidad	38
Capítulo 4	42
4	Conclusiones y Recomendaciones	43
4.1	Conclusiones	43
4.2	Recomendaciones	43

Simbología

CTD	Centro de Transformación Digital Industrial
SGBM	Semi-Global Block Matching
FPS	Fotogramas por segundo
WLS	Weighted Least Squares

Índice de figuras

Figura 1: Ángulos de orientación de cámara	5
Figura 2: Diagrama estéreo-visión	6
Figura 3: Tipos de patrón de calibración	7
Figura 4: Proceso de calibración estereoscópica	9
Figura 5: Cámara estereoscópica ELP Dual Lens 4 MP	18
Figura 6: Soporte de cámara ELP Dual Lens 4MP	19
Figura 7: Pipeline para calibración estereoscópica	20
Figura 8: Pipeline para generar mapas de profanidad	21
Figura 9: Patrón con alta textura	26
Figura 10: Ejemplo de calibración estereoscópica	33
Figura 11: Mapa de profundidad con y sin calibración	34
Figura 12: Visita a empresa empaadora de mangos	34
Figura 13: Interfaz gráfica para la calibración de sistema de visión estereoscópico	35
Figura 14: Selector de parámetros para mapa de profundidad	37
Figura 15: FPS y factor de escala	40

Índice de tablas

Tabla 1: Especificaciones de la cámara ELP Dual Lens 4MP	18
Tabla 2: Valores para parámetros no seleccionados	24
Tabla 3: Rango de búsqueda seleccionado para P_1 y P_2	26
Tabla 4: Parámetros intrínsecos previo a calibración	30
Tabla 5: Coeficientes de distorsión	30
Tabla 6: Parámetros de cámara posterior a la calibración	32
Tabla 7: Valores para parámetros del entorno de trabajo	36
Tabla 8: Mejores y peores 10 combinaciones de parámetros	36
Tabla 9: Tiempo de cómputo en función del factor de escala	39

Capítulo 1

1. Introducción

1.1. Descripción del problema

La detección y seguimiento de objetos en una cinta transportadora es clave para implementar sistemas de control de calidad y empaquetado automático de alimentos, pero para su implementación se requieren datos 3D del entorno de trabajo. Para la obtención de datos 3D existen 2 tipos de tecnología: LIDAR (Ligth Detection and Ranging) y estereovisión.

Un sistema LIDAR es robusto ante cambios de iluminación y su precisión no depende de las condiciones de operación de la escena que se esté observando. Mientras que un sistema de estéreo visión requiere una escena correctamente iluminada y que los objetos a detectar tengan suficiente textura [1]. Un sistema LIDAR no captura información de color de la escena, por lo que se suele usar junto con un sistema de visión tradicional. [1]. La principal limitante de un sistema LIDAR es su alto costo comparado con un sistema de estéreo visión. La principal limitación de los sistemas de estéreo visión es su sensibilidad a la calibración y el costo computacional.

A pesar sus limitaciones un sistema de estéreo visión es la solución más costo-efectiva cuando se puede controlar la iluminación y las condiciones de operación del entorno de trabajo y se usan valores adecuados en la calibración. Además, para distancias cortas como las usadas en este proyecto los datos de profundidad producidos por estéreo visión ofrecen mayor resolución espacial (cantidad de datos) que el LIDAR [2].

1.2. Justificación del problema

La implementación de plantas de empaquetado de alimentos autónomas requiere la obtención de datos de profundidad de los objetos ubicados sobre cintas transportadoras. El relativo bajo costo de estéreo visión permite el acceso a tecnologías de automatización a pequeñas y medianas empresas fortaleciendo su competitividad y crecimiento. Sin embargo, estos sistemas son sensibles a la calibración como a los parámetros de los algoritmos de procesamiento de imágenes.

En este proyecto se desarrolló un sistema semiautomático de calibración de cámaras

estereoscópicas y generación de mapas de profundidad, que a futuro pueda ser utilizado para la implementación de líneas de producción automáticas. El sistema de calibración propuesto fue evaluado en una línea de empaquetado de mangos de una empresa local.

1.3. Objetivos

1.3.1. Objetivos generales

Implementar un sistema de calibración para la obtención de datos de profundidad de una cinta transportadora mediante estéreo-visión.

1.3.2. Objetivos específicos

- Implementar un protocolo de calibración estereoscópica funcional usando patrón de tablero de ajedrez.
- Desarrollar un sistema de generación de mapas de profundidad usando algoritmos existentes en la literatura.
- Desarrollar una interfaz de usuario para seleccionar los parámetros de la generación de mapas de profundidad.
- Determinar y caracterizar el error en los mapas de profundidad generados.

1.4. Marco teórico

1.4.1. Parámetros ópticos de cámara

Los parámetros ópticos de una cámara definen como los objetos de una escena se proyectan sobre la imagen, estos parámetros pueden ser divididos en 2 grupos: parámetros intrínsecos y extrínsecos. Los parámetros extrínsecos definen la posición y orientación de la cámara en la escena, mientras que los parámetros intrínsecos definen como un punto del espacio 3D se proyecta sobre el plano 2D de la imagen (sensor)[3].

Los parámetros intrínsecos se suelen agrupar en forma de una matriz, a esta matriz se

le llama matriz intrínseca y se detalla a continuación.

$$A = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \quad (1)$$

Donde A es la matriz de intrínseca, f_x y f_y son las distancias focales para las direcciones x y y , c_x y c_y son las coordenadas x y y del centro del lente. Si bien la óptica de una cámara define una sola distancia focal f , la matriz de cámara define 2 valores f_x y f_y para compensar que los píxeles del sensor podrían no ser cuadrados perfectos, parte del proceso de calibración de una cámara implica aplicar las correcciones necesarias para garantizar que estos valores sean iguales [3].

Los parámetros extrínsecos también se agrupan en forma de matriz a esta matriz se la llama la matriz extrínseca que se detalla a continuación:

$$K = [R|\mathbf{t}] = \begin{bmatrix} R_{3 \times 3} & \mathbf{t}_{3 \times 1} \\ 0 & 1 \end{bmatrix} \quad (2)$$

Donde K es la matriz extrínseca que puede ser expresada como la concatenación entre la matriz de rotación R (ecuación 3) y el vector de traslación \mathbf{t} (ecuación 4) [3].

$$R = \begin{bmatrix} \cos(\psi) \cos(\theta) & -\sin(\psi) \cos(\phi) + \cos(\psi) \sin(\theta) \sin(\phi) & \sin(\psi) \sin(\phi) + \cos(\psi) \sin(\theta) \cos(\phi) \\ \sin(\psi) \cos(\theta) & \cos(\psi) \cos(\phi) + \sin(\psi) \sin(\theta) \sin(\phi) & -\cos(\psi) \sin(\phi) + \sin(\psi) \sin(\theta) \cos(\phi) \\ -\sin(\theta) & \cos(\theta) \sin(\phi) & \cos(\theta) \cos(\phi) \end{bmatrix} \quad (3)$$

La ecuación 3 muestra la matriz de rotación expresada en función de los ángulos de orientación de la cámara Yaw(ψ), Pitch(θ), Roll(Φ) (ver figura 1). Los valores de t_x , t_y y t_z son las coordenada x , y , z del la cámara en el entorno de trabajo [3].

$$\mathbf{t} = \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix} \quad (4)$$

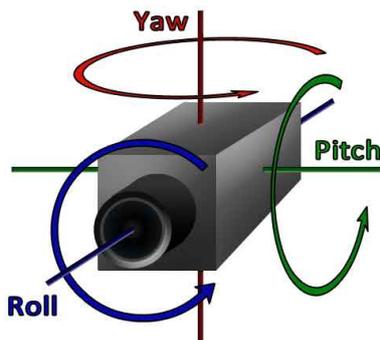
Figura 1:*Ángulos de orientación de cámara*

Imagen obtenida de [4]

La matriz de proyección P (ecuación 5) es la combinación de las matrices intrínseca y extrínseca, esta matriz indica como un punto de la escena \mathbf{p}_w se proyecta sobre el plano de la imagen en un píxel \mathbf{p} (ecuación 6).

$$P = [A|0] \times [K] \quad (5)$$

$$\mathbf{p} = P \times \mathbf{p}_w \quad (6)$$

1.4.2. Estéreo visión

La estéreo visión es el cálculo de la profundidad de la escena relativo a una de las cámaras del estéreo par, basándose en la disparidad binocular entre 2 cámaras, es decir, la captura de una misma escena desde 2 perspectivas. Se suele referir a 2 imágenes (izquierda y derecha) como estero-par o imagen estereoscópica. A partir de un estéreo-par se calcula la disparidad, la cual está definida como la distancia horizontal entre un píxel y su homólogo en la otra imagen (figura 2). Se considera un par de píxeles como homólogos si ambos provienen

del mismo punto de la escena. La relación entre disparidad y profundidad está dada por la ecuación 7.

Figura 2:

Diagrama estereo-visión

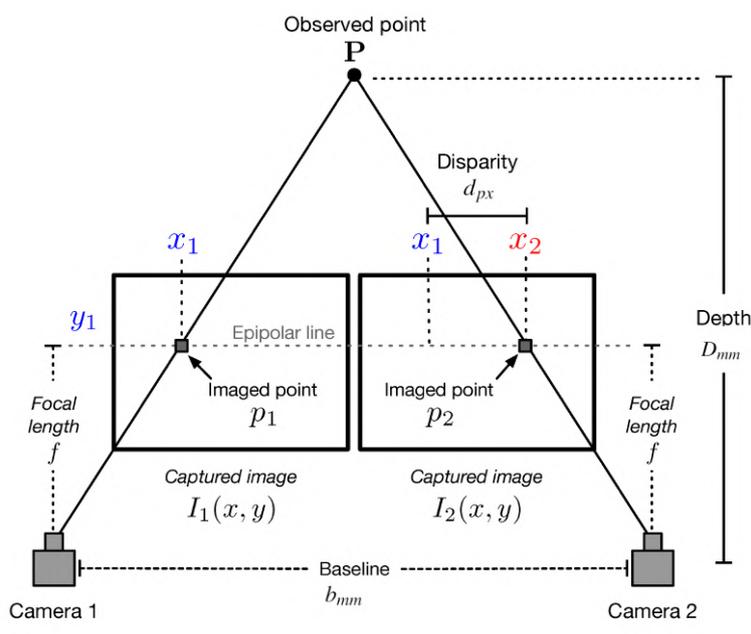


Imagen modificada de [5]

$$D_{mm} = \frac{f_{px} * b_{mm}}{d_{px}} \quad (7)$$

Donde D_{mm} es la profundidad en mm, f_{px} es la distancia focal de las cámaras en píxeles, b_{mm} es la distancia entre las cámaras en mm, d_{px} es la disparidad de un punto de la imagen en píxeles [3].

Es recomendable que ambas cámaras tengan la misma distancia focal, de no ser el caso una de las imágenes debe ser escalada para igualar las distancias focales mediante correcciones de software, esto no es recomendable que el escalamiento puede introducir artefactos no deseado en la imagen que afecten la calidad de los datos de profundidad.

Para el buen funcionamiento de las técnicas de estereo visión se requiere una iluminación uniforme y adecuada porque las regiones sobreexpuestas y subexpuestas generan valores de profundidad poco precisos [3]. Para el cálculo de la disparidad se necesita emparejar (matching) cada píxel de la imagen izquierda con su homólogo en la imagen

derecha.

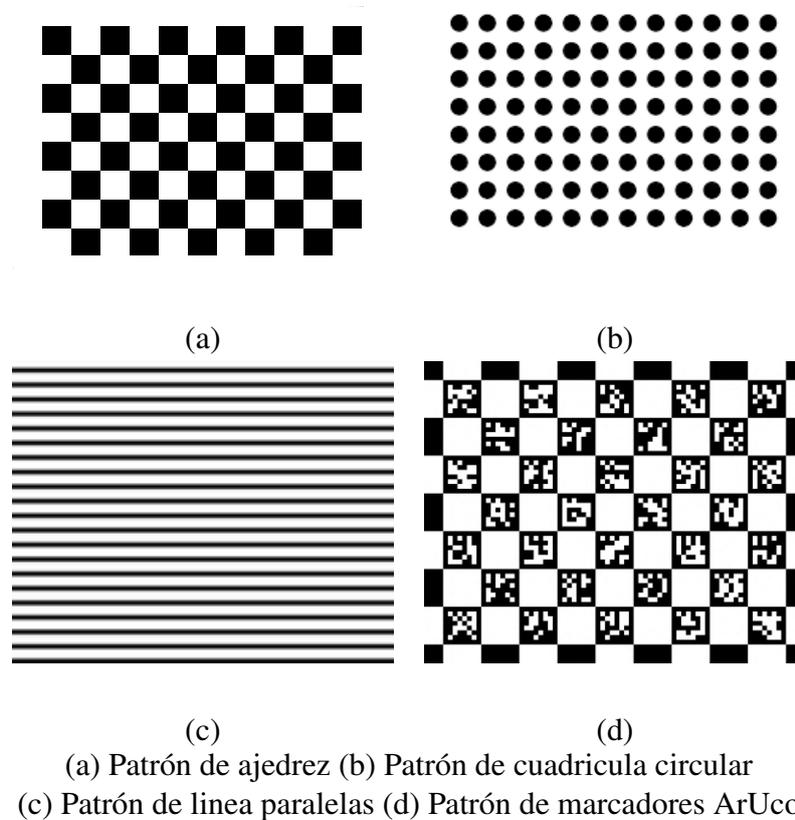
Para identificar puntos homólogos, la imagen debe tener niveles altos de textura (regiones distintivas en una escena). En regiones con textura los píxeles homólogos pueden ser localizados consistentemente y por ende el cálculo de la disparidad es más preciso, mientras que en regiones sin textura la disparidad resultante suele tener valores inconsistentes [3].

1.4.3. Calibración de un sistema de estéreo visión

Patrón de calibración

El patrón de calibración sirve como una imagen de referencia con geometría conocida y puntos fáciles de localizar. Al capturar una foto del patrón de calibración se puede observar como la imagen ha sido distorsionada por la óptica de la cámara y determinar que correcciones deben realizarse para eliminarlas.

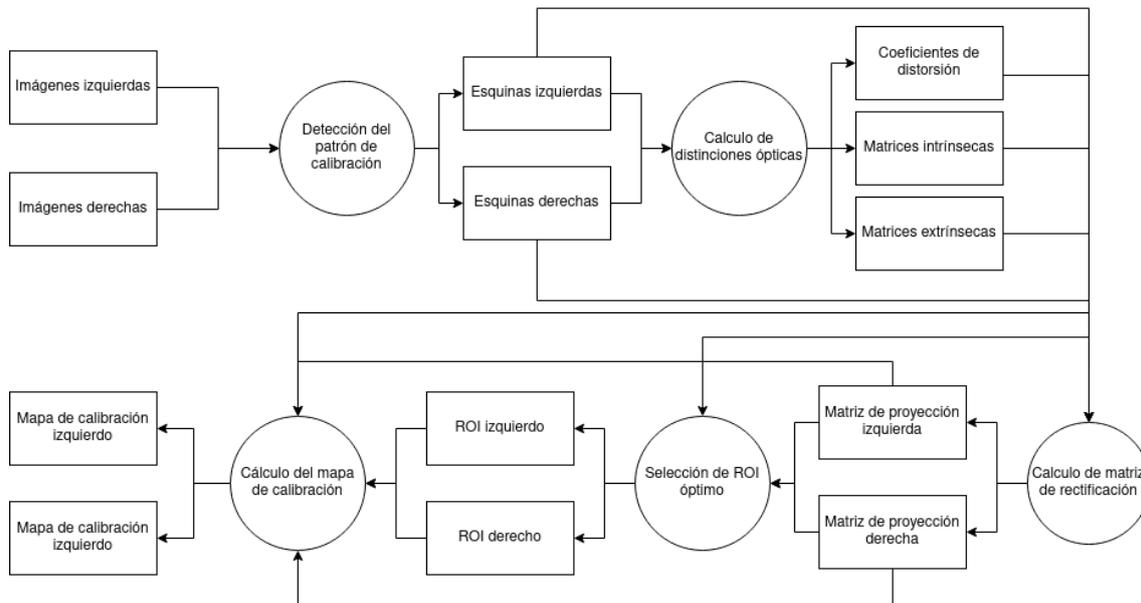
Existen varios tipos de patrones de calibración entre los más usados están: el tablero de ajedrez (figura 3 a), cuadrícula circular (figura 3 b), líneas paralelas (figura 3 c) y marcadores ArUco (figura 3 d). El tablero de ajedrez presenta esquinas bien definidas y fáciles de detectar, permite la estimación de distorsión radial y tangencial. El patrón de cuadrícula permite únicamente la estimación de distorsión radial y es computacionalmente intenso de procesar. Las líneas paralelas son fáciles de detectar, pero solo permiten la estimación de distorsión tangencial. El patrón de marcadores de ArUco permite la estimación de la orientación y puede trabajar con imágenes parciales del patrón sin ningún problema, sin embargo, su detección es computacionalmente intensa y no ofrece mejores resultados que el patrón de calibración de ajedrez [3].

Figura 3:*Tipos de patrón de calibración*

Si bien el patrón de ajedrez no ofrece datos de orientación, ni permite trabajar con tomas parciales como si lo permite el patrón ArUco, estas limitaciones no son relevantes para los sistemas de calibración de estereo visión, por lo que el uso del patrón de ajedrez es la opción más conveniente para este proyecto.

La calibración estereoscópica implica 5 pasos principales, la figura 4 muestra un diagrama de cada uno de los pasos:

1. Detección del patrón de calibración.
2. Cálculo de distorsiones ópticas.
3. Cálculo de matriz de rectificación.
4. Selección de ROI(Región de interés) óptimo.
5. Cálculo del mapa de calibración.

Figura 4:*Proceso de calibración estereoscópica*

A continuación, se explican cada uno de estos pasos:

Detección de patrón de calibración

Para detectar el patrón de calibración se ejecuta un detector de esquinas sobre la imagen del patrón, en escala de grises. Luego se valida que estas esquinas formen una malla con las dimensiones específicas del patrón.

En la librería OpenCV [6] el método que implementa este procedimiento es **cv::findChessboardCorners** el cual se basa en el algoritmo de detección de esquinas de Harris [7], las esquinas resultantes de la ejecución de este método se refinan usando **cv::cornerSubPix** [6] para estimar la posición subpíxel de cada esquina. El resultado de este paso son 2 listas (coordenadas izquierdas y derechas), cada elemento de una de estas listas es el conjunto de coordinas de las esquinas detectadas en un estéreo-par.

Cálculo de distorsiones ópticas

La distorsión de barril provoca que las líneas rectas se curven hacia afuera, alejándose del centro de la imagen. Este es un tipo común de distorsión en lentes de ángulo amplio como

los usados en este proyecto. La fórmula matemática para corregir la distorsión en barril se puede expresar en coordenadas polares como:

$$r_{\text{corrected}} = r_{\text{original}} \cdot \left(1 - k_1 \cdot r_{\text{original}}^2 - k_2 \cdot r_{\text{original}}^4 - k_3 \cdot r_{\text{original}}^6 - \dots \right) \quad (8)$$

Donde $r_{\text{corrected}}$ es la distancia corregida desde el centro de distorsión, r_{original} es la distancia original desde el centro de distorsión (intersección entre el plano de la imagen y el eje óptico), $k_1, k_2, k_3, \dots, k_n$ son los coeficientes de distorsión [6].

El método `cv::stereoCalibrate` de la librería OpenCV [6] es una implementación de este algoritmo que estima la matriz de intrínseca, los coeficientes de distorsión para ambas cámaras de estéreo-par, también calcula la matriz de rotación y el vector de traslación entre las cámaras del estéreo-par a partir de las coordenadas de las esquinas del patrón obtenidas en el paso anterior.

Cálculo de matriz de rectificación

La rectificación es una transformación de proyección 2D descrita por la siguiente matriz de 3x3, (ecuación 9).

$$B = \begin{bmatrix} s_x \cdot \cos(\theta) & -s_y \cdot \sin(\theta) & t_x \\ s_x \cdot \sin(\theta) & s_y \cdot \cos(\theta) & t_y \\ 0 & 0 & 1 \end{bmatrix} \quad (9)$$

Donde B es la matriz de rectificación, s_x, s_y son factores de escala para x y y , θ es el ángulo de rotación y t_x, t_y son los valores de traslación horizontal y vertical [8]. Esta transformación se aplica sobre las posiciones de los píxeles de la imagen (ecuación 10) con el fin de alinear los píxeles homólogos en las mismas filas y colocar el centro de la cámara en la misma posición (x,y) para ambas imágenes. Esta transformación facilita la correspondencia estéreo y el cálculo de disparidades, la ecuación que define esta transformación se detalla a continuación:

$$\begin{bmatrix} \mathbf{pf}_x \\ \mathbf{pf}_y \\ 1 \end{bmatrix} = B \times \begin{bmatrix} \mathbf{p}_x \\ \mathbf{p}_y \\ 1 \end{bmatrix} \quad (10)$$

Donde \mathbf{p} es un píxel de la imagen en forma de vector columna, y \mathbf{pf} es la posición final del píxel luego de aplicar la transformación, coordenadas homogéneas [8].

El método `cv::stereoRectify` de la librería OpenCV [6] implementa el cálculo de la matriz B para las cámaras (izquierda y derecha) y toma como argumentos: las posiciones de las esquinas del patrón obtenidas en el primer paso, las matrices de intrínsecas, los coeficientes de distorsión para ambas cámaras, la matriz de rotación y vector de traslación obtenidas en el paso anterior. Este método retorna las matrices de proyección para ambas cámaras.

Selección de ROI óptimo

La aplicación de correcciones ópticas y rectificación implica mover los píxeles de su posición incorrecta en la imagen original a su posición correcta en la imagen calibrada, este movimiento de píxeles puede causar que aparezcan regiones negras en los bordes de la imagen calibrada, estas regiones se deben a que no hay píxeles válidos con los que llenar dichos espacios [3]. La presencia de estas regiones puede causar problemas en el procesamiento posterior de las imágenes. Para solucionar este problema se selecciona el rectángulo (ROI) más grande que contenga únicamente píxeles válidos, además para facilitar el proceso de correspondencia estéreo es necesario que los rectángulos de ambas cámaras tengan el mismo centro y tamaño. Luego de la selección del ROI este rectángulo se escala para tener la misma resolución que la imagen original, debido al escalamiento es necesario volver a estimar los parámetros intrínsecos y extrínsecos para la imagen calibrada.

Cálculo de mapa de calibración

Por motivos de eficiencia computacional en lugar de aplicar las correcciones ópticas, rectificación y selección de ROI de manera independiente, se combinan estos pasos en un solo procedimiento. Este procedimiento consiste en usar las ecuaciones (8 y 10) y el ROI óptimo

para calcular la relación entre las posiciones de los píxeles en la imagen calibrada y la imagen original. Para cada píxel \mathbf{p} en la imagen calibrada se calcula su ubicación \mathbf{p}_2 en la imagen original. Los resultados de este cálculo se usan para la construcción de un Lookup Table (LUT) llamado mapa de calibración M donde $M(\mathbf{p}) = \mathbf{p}_2$, la ecuación 11 muestra cómo se usa M para obtener la imagen calibrada I_c a partir de la imagen original I_r :

$$I_c(\mathbf{p}) = I_r(M(\mathbf{p})) \quad (11)$$

El método `cv::initUndistorRectifyMap` se usa para calcular M para ambas cámaras independientemente. Esta función recibe la matriz de cámara, coeficiente de distorsión, matriz de rectificación, ROI óptimo de una cámara y retorna M para dicha cámara. Así se obtienen 2 mapas de calibración, uno para la cámara izquierda y otro para la cámara derecha. Estos mapas de calibración pueden usarse para aplicar la calibración a cualquier par estéreo capturado por el mismo sistema de calibración de manera eficiente.

1.4.4. Mapas de disparidad

En un par estéreo, la disparidad es la distancia horizontal medida en píxeles entre los puntos homólogos en ambas imágenes. Los objetos más cercanos a las cámaras tienen una mayor disparidad, mientras que los objetos más lejanos tienen una menor disparidad. Un mapa de disparidad es una imagen donde el valor de cada píxel equivale a la distancia entre un píxel en la imagen base (izquierda) y su homólogo en la imagen de correspondencia (derecha). El mapa de disparidad se usa para el cálculo del mapa de profundidad usando la ecuación 7 que convierte cada valor de disparidad en una profundidad.

Estimación del mapa de disparidad

`cv::StereoSGBM` es la implementación de OpenCV [6] del algoritmo *SemiGlobal Matching* descrito en [9]. StereoSGBM divide las imágenes en bloques, busca las correspondencias entre bloques de la imagen izquierda y derecha para lo cual emplea un enfoque de optimización semi-global (usando cambios locales, como globales). Este algoritmo cuenta con 3 pasos principales:

1. Cálculo del cubo de costos
2. Agregación de costos
3. Selección de valores de disparidad

Cálculo del cubo de costos

El cubo de costos se construye comparando cada píxel en la imagen izquierda con una serie de píxeles en la imagen derecha para posibles valores de disparidad. La medida de costo para un píxel p y una disparidad d se puede expresar mediante una función de costo (ecuación 12). Para la construcción de este cubo se calcula el costo para cada posible disparidad en un rango definido por los parámetros: disparidad mínima y disparidad máxima a intervalos de 1 px. El cubo tiene dimensiones $width \times height \times d_{range}$, donde $width$ y $height$ son las dimensiones de la imagen y d_{range} es el rango de disparidades posibles.

$$C(\mathbf{p}_x, \mathbf{p}_y, d) = \sum_{\mathbf{p}2 \in N(\mathbf{p}, w) \cup \mathbf{p}} (I_L(\mathbf{p}2_x, \mathbf{p}2_y) - I_R(\mathbf{p}2_x - d, \mathbf{p}2_y))^2 \quad (12)$$

Donde \mathbf{p}_x y \mathbf{p}_y son las coordenadas (x, y) de un píxel \mathbf{p} , d es el valor de disparidad, w es el tamaño de la ventana de búsqueda, $N(\mathbf{p}, w) \cup \mathbf{p}$ son todos los píxeles en un vecindario de tamaño w alrededor de p incluyendo \mathbf{p} , I_L y I_R son las imágenes izquierda y derecha respectivamente [10].

Agregación de costos

Después de construir el cubo de costos, se realiza un proceso de agregación para encontrar la disparidad global óptima para cada píxel y mejorar la calidad de los resultados. La agregación se realiza mediante programación dinámica a lo largo de 16 direcciones, que incluyen izquierda-derecha, arriba-abajo y diagonales. Se busca minimizar una función de costo global $S(\mathbf{p}, d)$ (ecuación 13) que considera la acumulación de costos a lo largo de las direcciones seleccionadas.

$$S(\mathbf{p}, d) = \sum_{\mathbf{r}} L'_{\mathbf{r}}(\mathbf{p}, d) \quad (13)$$

Donde \mathbf{r} es un vector que corresponde a cada una de las direcciones a evaluar [9].

La función de acumulación de costos para una dirección \mathbf{r} (ecuación 14) se define a continuación

$$\begin{aligned}
 L'_r(\mathbf{p}, d) = & C(\mathbf{p}, d) + \min(L'_r(\mathbf{p} - \mathbf{r}, d), \\
 & L'_r(\mathbf{p} - \mathbf{r}, d - 1) + P_1, \\
 & L'_r(\mathbf{p} - \mathbf{r}, d + 1) + P_1, \\
 & \min_i L'_r(\mathbf{p} - \mathbf{r}, i) + P_2) - \min_k L'_r(\mathbf{p} - \mathbf{r}, k)
 \end{aligned} \tag{14}$$

Donde P_1 y P_2 son parámetros de penalización para disparidades pequeñas y grandes respectivamente, i son todos los valores de disparidad excepto $\{d - 1, d, d + 1\}$, k son todos los valores de disparidad en el rango de búsqueda D_{range} [9].

Selección de disparidad

Luego de obtener el cubo de costos agregado $S(\mathbf{p}, d)$, para cada \mathbf{p} se selecciona la disparidad d que contenga el costo agregado mínimo (ecuación 15). El resultado de este paso es un mapa de disparidad $D(\mathbf{p})$ con las mismas dimensiones de la imagen calibrada.

$$D(\mathbf{p}) = \underset{d}{\operatorname{arg\,min}} S(\mathbf{p}, d) \tag{15}$$

Filtrado Weighted Least Squares (WLS)

Finalmente se aplica el filtro WLS que rellena los valores de disparidad con baja confianza con valores de píxeles vecinos con alta confianza, su objetivo principal es suavizar el mapa de disparidad, en base a los siguientes parámetros:

1. Sigma Color (σ_{color}) Controla la sensibilidad a las diferencias de color en la imagen. Un valor mayor de σ_{color} permite una mayor variación en el color entre píxeles vecinos antes de considerarlos como diferentes.
2. Lambda (λ): Controla el nivel de suavizado del filtro. Un valor mayor de λ resulta en un suavizado más fuerte.

La confianza $F(\mathbf{p})$ del valor de disparidad de un píxel \mathbf{p} se calcula así:

$$V(\mathbf{p}) = \frac{\sum_{\mathbf{p}_2 \in N(\mathbf{p}, w) \cup \mathbf{p}} D(\mathbf{p}_2)^2}{\|N(\mathbf{p}, w) \cup \mathbf{p}\|} - \left(\frac{\sum_{\mathbf{p}_2 \in N(\mathbf{p}, w) \cup \mathbf{p}} D(\mathbf{p}_2)}{\|N(\mathbf{p}, w) \cup \mathbf{p}\|} \right)^2 \quad (16)$$

$$F(\mathbf{p}) = \min\left(1 - \frac{V(\mathbf{p})}{1000}, 0\right) \quad (17)$$

El rango de valores confianza va de 0 (sin confianza) a 1 (confianza total). $N(\mathbf{p}, w) \cup \mathbf{p}$ son todos los píxeles vecinos de \mathbf{p} incluyendo \mathbf{p} en una ventana de w píxeles.

Capítulo 2

2. Metodología

2.1. Análisis

En esta sección se listan los requerimientos funcionales y no funcionales identificados para el sistema de calibración estereoscópica y generación de mapas de profundidad según las necesidades del cliente.

2.1.1. *Requerimientos funcionales*

- Generar archivos de calibración usando el protocolo de calibración estereoscópica propuesto.
- Seleccionar los mejores parámetros para la generación de mapas de profundidad.
- Generar mapas de disparidad y profundidad para un entorno de trabajo real.
- Medir el tiempo de ejecución y FPS del algoritmo de generación de mapas de profundidad desarrollado.

2.1.2. *Requerimientos no funcionales*

- El sistema desarrollado debe ejecutarse en un ordenador con sistema operativo Linux.
- El sistema debe de funcionar con cualquier cámara que soporte el protocolo video4linux2.
- Medir el tiempo de procesamiento del sistema en función de los parámetros seleccionados.

2.1.3. *Alcance de la solución*

Para este proyecto se desarrollaron 2 programas: uno para la calibración estereoscópica que tiene interfaz de línea de comandos e interfaz gráfica y permite seleccionar las imágenes del patrón de calibración y las dimensiones del mismo; el segundo programa permite la selección de parámetros para la generación de mapas de profundidad y ver en

tiempo real los resultados. Todas las funcionalidades necesarias para la calibración estereoscópica y la generación de mapas de profundidad fueron colocadas dentro de la librería StereoCamLib que se desarrolló durante este proyecto.

2.2. Sistema de estéreo visión

Para probar el protocolo de calibración implementado se utilizó la cámara *ELP Dual Lens 4MP* (figura 5), la cual fue provista por el Centro de Transformación Digital (CTD). A diferencia de un sistema de estéreo visión convencional donde se trabaja con 2 cámaras independientes la cámara *ELP Dual Lens 4MP* integra 2 sensores en una solo PCB es decir la sincronización de cámaras viene implementada en hardware, al estar colocadas en un mismo PCB la posición relativa entre las cámaras no cambia, lo cual es necesario para un sistema de estéreo visión.

Figura 5:

Cámara estereoscópica ELP Dual Lens 4 MP



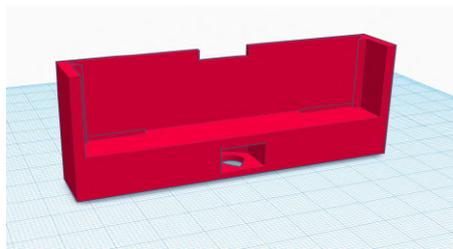
Cámara ELP Dual Lens 4MP

Las especificaciones de la cámara *ELP Dual Lens 4MP* se detallan a continuación.

Tabla 1:*Especificaciones de la cámara ELP Dual Lens 4MP*

Parámetro	Valor
Resolución (px)	1920x1080
FPS ($\frac{1}{s}$)	70
Formato	MJPEG, YUY2
Sensor	1/2.7 " JX-F35
Tipo de sensor	CMOS
Ángulo de visión	120° H
Separación entre lentes	58 mm
Distancia focal (px) ¹	594.6

Un problema que se detectó con la cámara *ELP Dual Lens 4MP* es que el PCB es muy flexible y esto ocasionaba que los ejes ópticos de ambas cámaras no estén paralelos cuando se fijaba a un soporte flexible o se ejercía presión sobre el PCB. Para resolver este problema se diseñó e imprimió en plástico un soporte para sostener las cámaras sin ejercer presión sobre el PCB (figura 6).

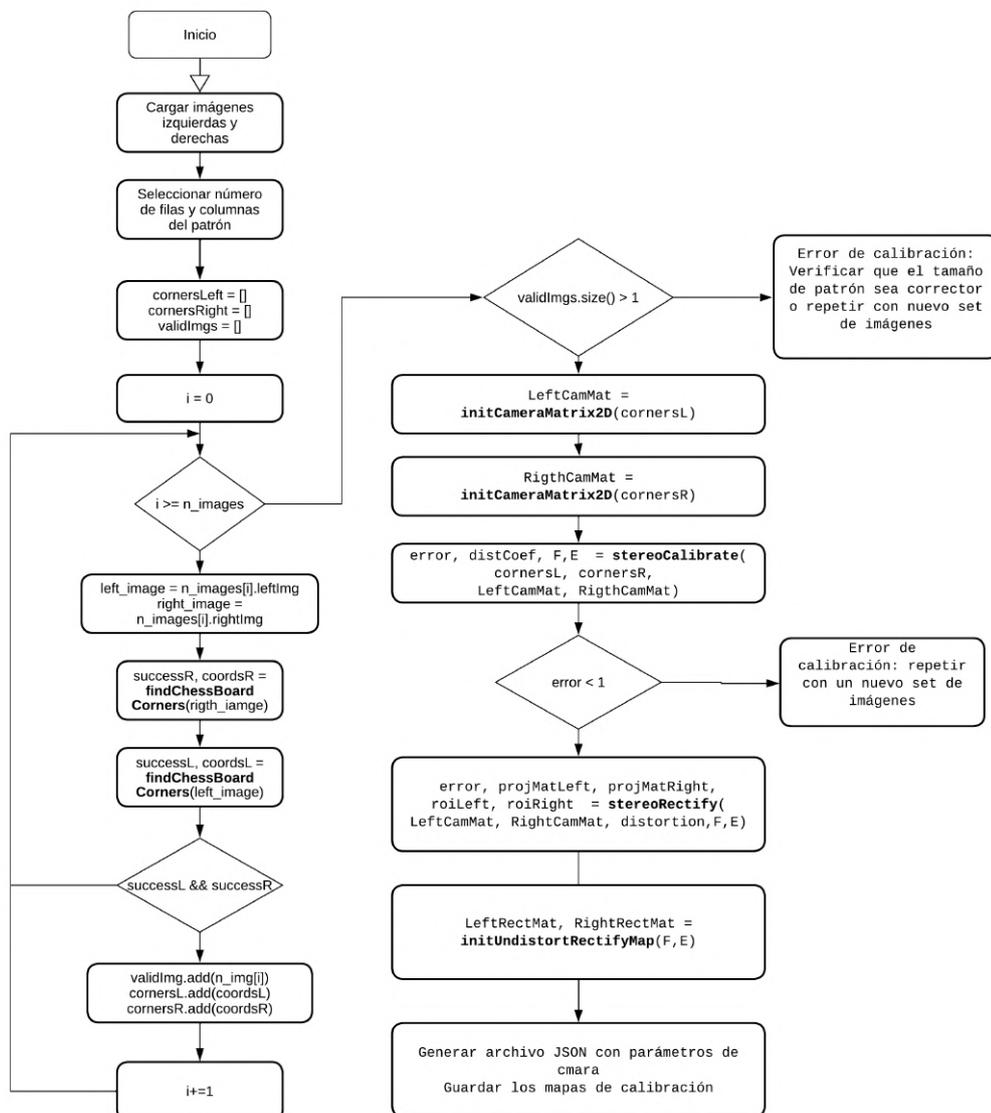
Figura 6:*Soporte de cámara ELP Dual Lens 4MP*

2.3. Diseño de la solución

El protocolo de calibración estereoscópica es un procedimiento estándar, este proceso inicia con la captura de fotos del patrón de calibración desde varias perspectivas, el patrón debe ser completamente visible en ambas cámaras (izquierda y derecha), para cada estéreo-par se obtienen las coordenadas de las esquinas del patrón de calibración, con estas coordenadas se realiza: corrección de distorsión, rectificación y selección de ROI óptimo según lo descrito en la sección 1.4.3. La figura 7 muestra el algoritmo de calibración estereoscópica utilizado.

Figura 7:

Pipeline para calibración estereoscópica

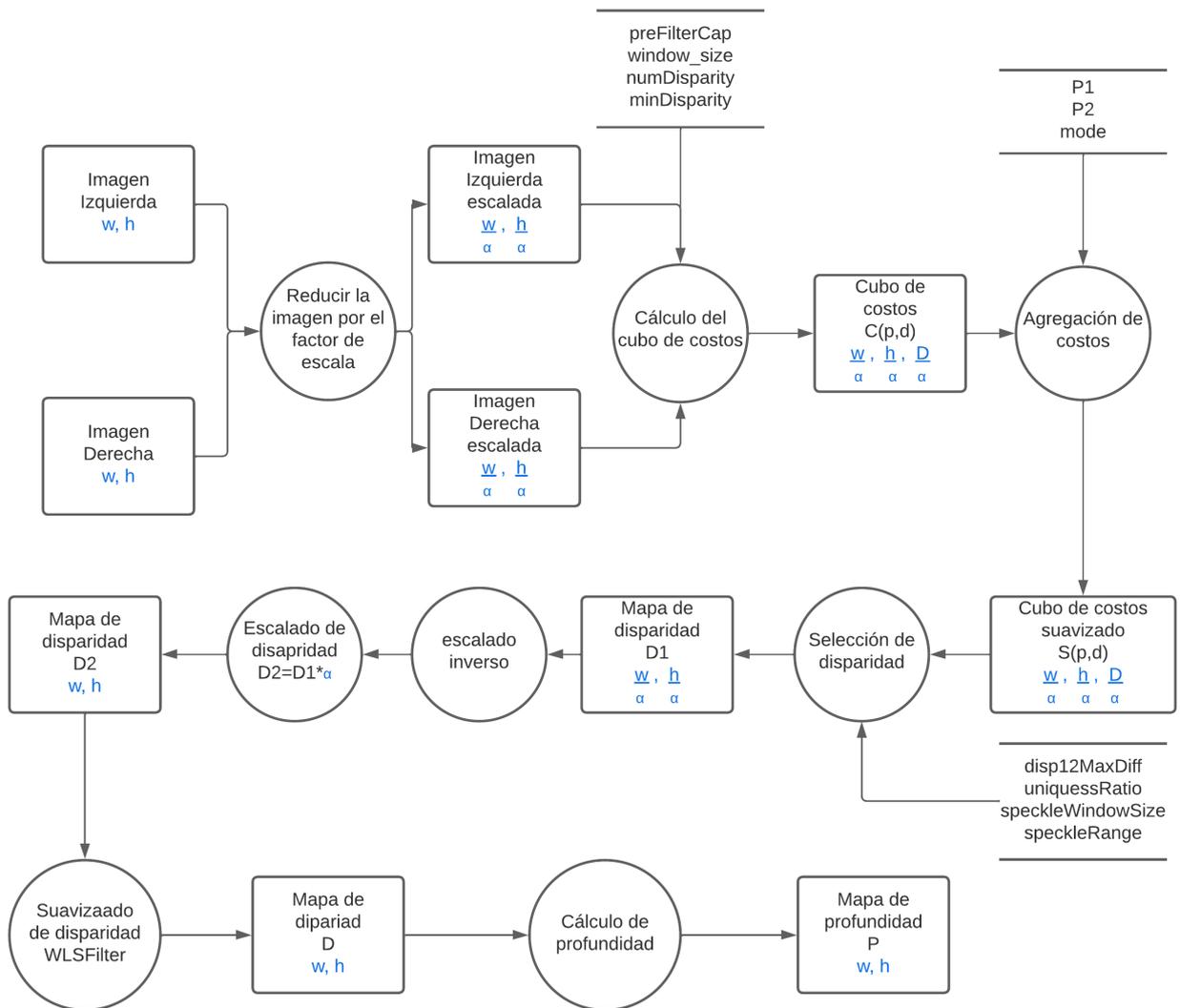


En la imagen se muestra paso a paso el pipeline de calibración estereoscópica

El resultado de la calibración son 2 archivos: JSON de configuración que contiene los parámetros intrínsecos y extrínsecos de las cámaras, archivo YML que contiene los mapas de calibración para ambas cámaras.

Para la obtención de mapas de disparidad se hace uso del algoritmo StereoSGBM seguido de un paso de refinamiento usando filtrado WLS y finalmente se calcula el mapa de profundidad usando la ecuación 7. El flujo de datos empezando desde las imágenes calibradas hasta el mapa de profundidad se presenta en la imagen 8, en la imagen también se muestra en color azul las dimensiones de cada objeto donde w, h son el ancho y alto de la imagen calibrada, α es el factor de escala que se explicara más adelante, \mathbf{D} es el número de disparidades, los parámetros del algoritmo StereoSGBM se muestran en nodos de datos (cuadrados sin líneas verticales) contados al proceso (círculo) que los utiliza.

Figura 8:
Pipeline para generar mapas de profanidad



2.3.1. Configuración de parámetros

El pipeline propuesto para la calibración estereoscópica solo requiere 2 parámetros: el número de filas y columnas del patrón de calibración, mientras que el pipeline para generar mapas de profundidad requiere 12 parámetros: 11 pertenecen al algoritmo `cv::StereoSGBM` y se explicarán más adelante; 1 parámetro: *scale* fue introducido como un paso extra antes del cómputo del mapa de disparidad para controlar el tiempo de procesamiento, el parámetro *scale* controla la reducción de resolución del estéreo-par, esto resulta en un mapa de disparidad con menor resolución que el estéreo-par por lo tanto es necesario realizar un

escalado inverso a este mapa (ver figura 8).

Para el escalado de las imágenes estereoscópicas previo al cómputo del mapa de disparidad se utiliza el método **cv::resize** con el factor de escala *scale* y el método de interpolación **cv::INTER_AREA** ya que este método conserva mejor la textura de la imagen, mientras que en el escalado inverso del mapa de disparidad se utiliza el método de interpolación **cv::INTER_NEAREST** para conservar los bordes del mapa de disparidad.

Existen varias técnicas para la selección de valores óptimos para un algoritmo, sin embargo dada la limitación de no contar con un algoritmo matemáticamente diferenciable (función matemática con derivadas para todo su dominio) el problema se limita únicamente a algoritmos de fuerza bruta [11]; el cual que consiste en calcular todas las posibles combinaciones de parámetros y seleccionar la combinación que tenga menor error según una métrica definida. El problema con este método es el crecimiento exponencial de la cantidad de combinaciones en función del número de parámetros, lo que implica grandes cantidades de: poder de cómputo, almacenamiento y tiempo[11].

Por lo tanto, es necesario reducir la cantidad de parámetros para los cuales se necesita encontrar valores óptimos. Para reducir la cantidad de parámetros se utilizó las restricciones conocidas del sistema, también se reescribió varios parámetros en función de unidades métricas que facilitan la compresión y selección de estos parámetros. A continuación, explican los parámetros para la generación de mapas de profundidad y las restricciones que limitan los posibles valores de cada parámetro.

El algoritmo **cv::stereoSGBM** tiene 11 parámetros:

minDisparity y **numDisparity** controlan el rango de valores búsqueda de disparidad para la construcción del cubo de costos (ecuación 12). Estos valores pueden ser calculados en función del rango de distancia que se desea medir usando la ecuación 18.

$$d_{px}(D_{mm}, scale) = \frac{f_{px} * b_{mm}}{D_{mm} * scale} \quad (18)$$

$$minDisparity = d_{px}(distancia_minima, scale) \quad (19)$$

$$numDisparity = d_{px}(distancia_maxima, scale) - minDisparity \quad (20)$$

Idealmente *distancia_maxima* debe ser la distancia al objeto más lejano visible en la imagen y *distancia_minima* debe ser la distancia al objeto más cercano visible en la imagen. La presencia en objetos fuera de este rango de distancias genera errores en el mapa de disparidad.

window_size es el tamaño de la ventana de búsqueda, su valor puede ser calculado en base a la resolución espacial de datos que se desee en el mapa de disparidad

$$window_size = \lfloor \frac{resolucion_{mm} * f_{px}}{distancia_maxima * scale} \rfloor$$

P₁ y **P₂** son penalidades para cambios pequeños y grandes de disparidad de píxeles vecinos. Los valores recomendados por OpenCV [6] para estas penalidades son

$$P_1 = window_size^2 * 8$$

$$P_2 = window_size^2 * 32$$

. además $P_1 \geq P_2$ [9]

disp12MaxDiff, **preFilterCap**, **uniquenessRatio**, **speckleWindowSize** y **speckleRange** fueron descartados porque no se evidenció que sus valores modifiquen significativamente el mapa de disparidad resultante. A continuación, se muestran los rangos de valores probados para cada parámetro descartado, así como el valor que se le dará a estos parámetros en el resto del proyecto.

Tabla 2:*Valores para parámetros no seleccionados*

Parámetro	valores probado	valor seleccionado
disp12MaxDiff	[-1, 100]	-1
preFilterCap	[-1, 255]	0
uniquessRatio	[5, 15]	0
speckleWindowSize	[50, 200]	0
speckleRange	[0, 2]	0

mode es un valor que indica el número de direcciones para el proceso de agregación de costos (ecuación 14). Los valores posibles son:

- **cv::StereoSGBM::MODE_SGBM** 4 direcciones
- **cv::StereoSGBM::MODE_SGBM_3WAY** 6 direcciones
- **cv::StereoSGBM::MODE_HH** 4 direcciones
- **cv::StereoSGBM::MODE_HH4** 8 direcciones

cv::StereoSGBM::MODE_HH y **cv::StereoSGBM::MODE_HH4** utilizan una variante jerárquica del algoritmo donde se inicia con las imágenes en muy baja resolución y esta se va aumentando progresivamente con el objetivo de eliminar falsos positivos y mejorar la precisión en regiones de baja textura. Se decidió trabajar con el modo **cv::StereoSGBM::MODE_HH4** ya que ofrece la mayor precisión, su limitante es el costo computacional, pero el mismo se puede controlar usando el factor de escala *scale*.

De esta forma se logro seleccionar 2 parámetros cuyos valores óptimos deben ser encontrados: **P1**² y **P2**.

²Los parámetros recomendados por OpenCV para P_1 y P_2 podrían no ser óptimos para la aplicación en cuestión.

Para la búsqueda de los valores óptimos se utilizó Grid Search[12], es un algoritmo de fuerza bruta que consiste en recorrer el espacio de parámetros para un número predeterminado de valores, calcular el error para cada combinación y seleccionar aquella con menor error.

Tabla 3:

Rango de búsqueda seleccionado para P_1 y P_2

Parámetro	Rango de valores
P1	$P1_range = \{0, 2, 4, \dots, 16\}$
P2	$P2_range = \{0, 8, 16, \dots, 64\}$

El conjunto de todas la combinaciones validas entre valores de P_1 y P_2 para los rangos de valores en tabla 3 descrito por:

$$CP_1P_2 = \{(P_1, P_2) \mid P_1 \in P1_range, P_2 \in P2_range, P_2 \geq P_1\} \quad (21)$$

Donde $P1_range$ y $P2_range$ son los rangos de valores para P_1 y P_2 , (P_1, P_2) es cada uno de los elementos de CP_1P_2 que es el conjunto de las 60 combinaciones validas para los rangos establecidos en la tabla 3.

Con la lista de parámetros y sus rangos de valores seleccionados se realizó GridSearch con todas las posibles combinaciones de parámetros (CP_1P_2), para cada combinación se determina el error MSE (Error cuadrático medio) según la ecuación 22. Luego de obtener el error para cada combinación de parámetros se selecciona la combinación de parámetros con menor error.

Para determinar el error de un mapa de disparidad se diseñó un experimento usando un patrón rectangular con mucha textura [13] Figura 9.

Figura 9:*Patrón con alta textura*

El patrón se imprimió a blanco y negro con un tamaño de 53 x 36 cm en escala de grises StereoSGBM funciona sobre imágenes en escala de grises de esta forma no afecta a los resultados imprimir el patrón a escala de grises. La impresión del patrón se pegó a una tabla de madera de 60 x 40 cm para asegurarse que este se mantenga plano. Se colocó el sistema de estéreo visión a 3 alturas: 30, 40 y 50 cm medidos desde la tabla de madera que contiene el patrón hasta el punto focal de la cámara izquierda, es necesario garantizar que el sistema de estéreo visión se encuentre paralelo al patrón. Se capturaron 10 estéreo-par para cada altura, luego de capturar cada imagen se desplazó el patrón horizontalmente para cubrir todo el plano de imagen entre las 10 fotos de cada altura.

Una vez obtenidas las imágenes estereoscópicas se procede a generar los mapas de disparidad con cada una de las 60 combinaciones de parámetros dentro CP_1P_2 y se usa la ecuación 22 para el cálculo del error. Para cada combinación de parámetros calcularon 30 mapas de profundidad (*3 alturas * 10 imágenes*).

$$MSE = \sum_h \sum_{D \in D_s(h)} \sum_{\mathbf{p} \in D} (D(\mathbf{p}) - h)^2 \quad (22)$$

Donde h es cada una de las alturas (30, 40 y 50 cm), $D_s(h)$ es el conjunto de todos los mapas de profundidad generados a una altura h . D es cada uno de los mapas de profundidad dentro de D_s , \mathbf{p} es cada uno de los píxeles del mapa de disparidad solo en región donde se encuentra

el patrón, $D(\mathbf{p})$ es la profundidad asignada al píxel \mathbf{p} en el mapa de profundidad D .

Capítulo 3

3. Resultados y Análisis

3.1. Resultados de calibración estereoscópica

Para iniciar el proceso de calibración de la cámara *ELP Dual Lens 4MP* primero se capturaron 30 fotos del patrón de calibración (figura 3(a)) desde diferentes perspectivas, el primer paso de la calibración es encontrar las coordenadas en píxeles de las esquinas del patrón de calibración en de cada estéreo-par. Tras obtener estas coordenadas se continúa con el cálculo de las distorsiones ópticas, para lo cual hay que estimar los parámetros intrínsecos y coeficientes de distorsión de cada cámara, estos resultados se muestran a continuación.

Tabla 4:

Parámetros intrínsecos previo a calibración

Parámetro	Cámara izquierda	Cámara derecha
Distancia focal X f_x (px)	593.17	593.17
Distancia focal Y f_y (px)	594.29	594.29
Centro X (px)	957.16	963.74
Centro Y (px)	591.17	571.17

En la tabla 4 se muestran los parámetros intrínsecos de las cámaras izquierda y derecha, las distancias focales f_x y f_y son iguales entre ambas cámaras esto garantiza que se cumpla la restricción mencionada en la sección 1.4.2 que indica que ambas cámaras deben tener una misma distancia focal. Además, f_x no es igual a f_y , esto indica que los píxeles del sensor no son cuadrados perfectos, la relación de aspecto $\frac{f_y}{f_x} = 1.001888$ indica que los píxeles del sensor son ligeramente más altos que anchos, además, las cámaras tienen centros diferentes esto se debe usualmente a defectos de fabricación, estos defectos ópticos complican la búsqueda de píxeles homólogos, por lo tanto en el siguiente paso de calibración se corregirán estos defectos.

Tabla 5:*Coefficientes de distorsión*

Coefficiente	Cámara izquierda	Cámara derecha
k1	1.4439e-02	-1.4838e-02
k2	6.9285e-03	1.7702e-03
k3	-9.1024e-06	4.1649e-05
k4	6.5267e-05	2.9038e-04
k5	5.7572e-03	2.0627e-02
k6	3.0576e-03	-1.9401e-02
k7	3.2922e-02	1.9151e-02
k8	3.2790e-03	2.1207e-02
k9, ..., k14	0.0	0.0

En la tabla 5 se muestran los coeficientes de distorsión calculados para ambas cámaras, es importante mencionar que a pesar de que ambas cámaras usan los mismos lentes y sensores los coeficientes de distorsión son diferentes entre sí esto implica que la calibración de la cámara no es un proceso transferible entre dispositivos (incluso entre dispositivos similares). Además los coeficientes de distorsión pueden cambiar a lo largo del tiempo debido a vibraciones que pueden desplazar los componentes de la óptica de la cámara, esto implica que es necesario recalibrar el sistema con frecuencia.

Una vez obtenidos los coeficientes de distorsión se procede con el cálculo de las matrices de rectificación, que consiste en una transformación lineal que se aplica a cada imagen para alinear sus centros y colocar puntos homólogos en las mismas filas. A continuación se muestran las matrices de rectificación obtenidas.

Matrices de rectificación

$$\begin{array}{ccc} \left[\begin{array}{ccc} 1 & 0 & -54.47 \\ 0 & 1.001888 & -3.06 \\ 0 & 0 & 1 \end{array} \right] & & \left[\begin{array}{ccc} 1 & 0 & -61.05 \\ 0 & 1.001888 & -16.94 \\ 0 & 0 & 1 \end{array} \right] \\ \text{Izquierda} & & \text{Derecha} \end{array}$$

Usando la ecuación 9 se puede calcular que el ángulo de rotación $\theta = 0$ para ambas cámaras, el factor de escala $s_x = 1$ y $s_y = 1.00188$ para ambas cámaras. $\theta = 0$ debido a que en el hardware las cámaras y planos de imagen están alineados, $s_y = 1.00188 = \frac{f_x}{f_y}$ garantiza que f_x sea igual a f_y estirando la imagen mínimamente en la dirección vertical. $s_x = 1$ implica que no se realiza ningún estiramiento horizontal.

El siguiente paso de la calibración estereoscópica es la selección del ROI óptimo (Región de la imagen donde no hay píxeles inválidos). Para esta calibración se obtuvo que el ROI seleccionado es equivalente a toda la imagen, esto implica que durante las correcciones de distorsiones ópticas y rectificación ningún píxel inválido fue introducido dentro del área de la imagen. Por lo tanto, este paso no tiene ningún efecto para la calibración del hardware, este paso podría ser relevante para la calibración de otro sistema de estéreo-visión.

Finalmente se calculan los mapas de calibración para cada cámara y se guardaron en un archivo YML. También se guardaron dentro de un JSON los parámetros de intrínsecos y extrínsecos para las imágenes calibradas detalladas a continuación.

Tabla 6:*Parámetros de cámara posterior a la calibración*

Parámetro	Valor	Tipo
Distancia focal izquierda	594.29 px	Intrínseco
Centro izquierdo	(588.11, 902.69) px	Intrínseco
Distancia focal derecha	594.29px	Intrínseco
Centro derecho	(588.11, 902.69) px	Intrínseco
Separación entre cámaras	58 mm	Extrínseco

Luego de ejecutar el proceso de calibración se obtuvo un error cuadrático medio (rms) de 0.81, este valor es inferior a 1 píxel entonces se puede concluir que la calibración es adecuada para el propósito del proyecto. A continuación, se muestra un ejemplo de los resultados de calibración

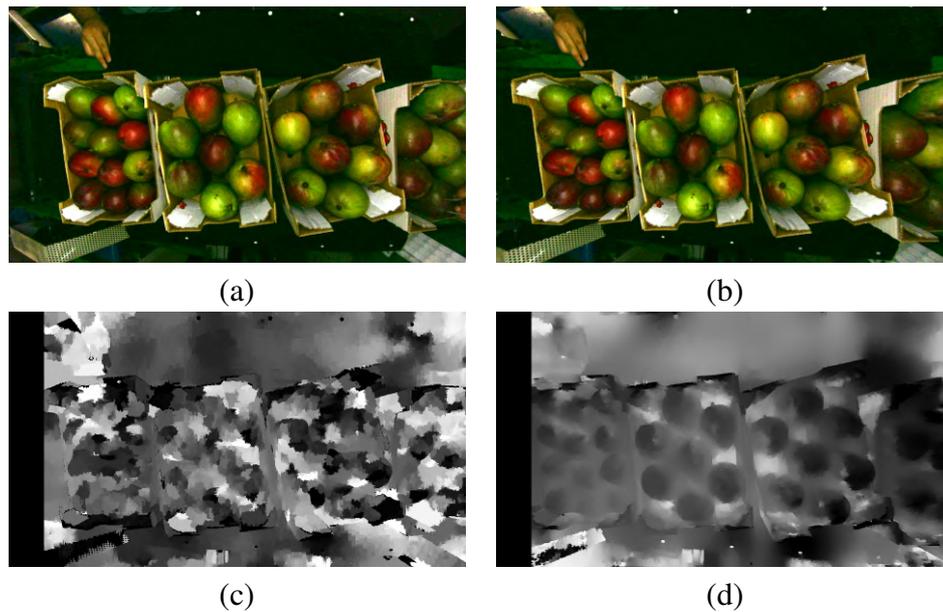
Figura 10:*Ejemplo de calibración estereoscópica*

En (figura 10) se muestra un ejemplo del resultado de calibración, se puede apreciar como las líneas horizontales y verticales del patrón aparecen curvas en las imágenes sin

calibrar debido a la distorsión del lente, en las imágenes calibradas se muestra como este problema se resuelve luego de la calibración. El efecto de la rectificación es más difícil de visualizar en las imágenes, pero sus efectos son evidentes durante la construcción del mapa de profundidad como se muestra en la figura 11.

Figura 11:

Mapa de profundidad con y sin calibración



(a) Imagen izquierda sin calibración (b) Imagen izquierda calibrada
 (c) Mapa de profundidad usando imágenes sin calibración
 (d) Mapa de profundidad usando imágenes calibradas

Para probar el sistema de calibración y obtención de mapas de profundidad se realizó una visita a una empresa local empacadora de mangos (figura 12) y se colocó el sistema de estereo visión a 41 cm de altura sobre una de sus líneas de producción, la figura 11 a muestra un ejemplo de las imágenes capturadas durante la visita. Para la generación de (c) y (d) se usó los parámetros seleccionados en la sección 3.2

Es evidente que el mapa de profundidad generado con imágenes no calibradas (figura 11c) presenta más ruido y no se puede distinguir las formas de los objetos en la escena, mientras que el mapa de profundidad generado con imágenes calibradas (figura 11d) se puede distinguir claramente los bordes de las cajas y cada uno de los mangos, así como su curvatura (cambio de profundidad entre píxeles vecinos).

Figura 12:

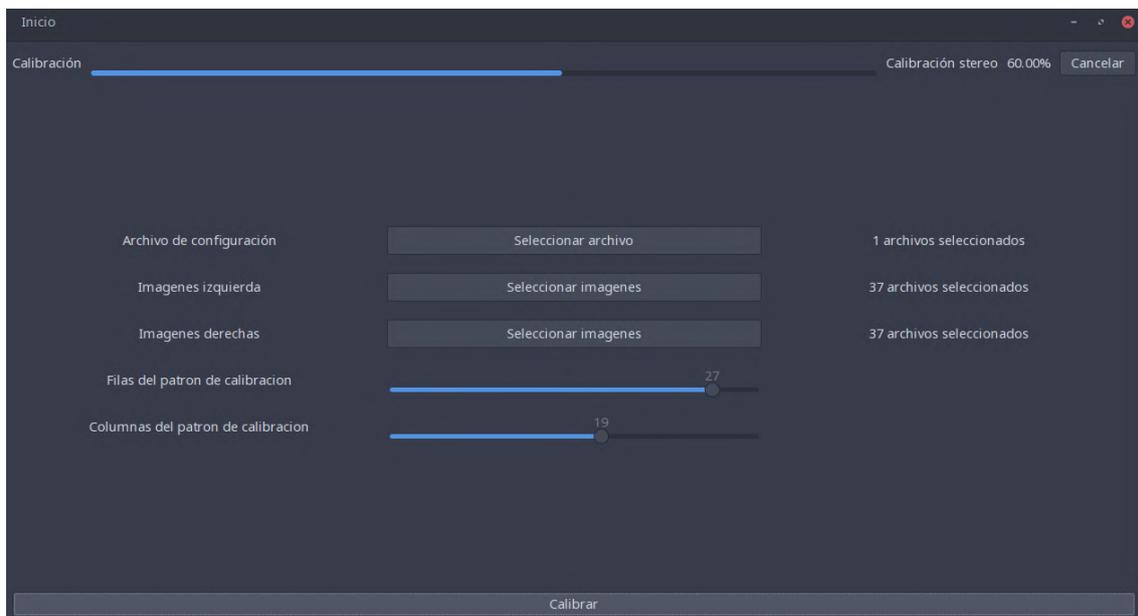
Visita a empresa empaadora de mangos



Para facilitar el uso del protocolo de calibración se desarrolló una interfaz gráfica que permite la selección de parámetros de calibración: imágenes izquierdas y derechas que contienen el patrón de calibración, el archivo de configuraciones JSON, la cantidad de filas y columnas de patrón de calibración de tipo tablero de ajedrez figura 13. Para el desarrollo de esta interfaz se utilizó el framework GTK 3.0 en el lenguaje de programación C++.

Figura 13:

Interfaz gráfica para la calibración de sistema de visión estereoscópico



3.2. Selección de parámetros óptimos para la generación de mapas de profundidad

El algoritmo `cv::StereoSGBM` usado para la generación de mapas de disparidad tiene 11 parámetros, en la sección 2.3.1 se definió los valores constantes que se asignaron a *disp12MaxDiff*, *perFilterCap*, *uniquenessRatio*, *speckleWiondowSize*, *speckleRange*, *mode*. También se explicó que el valor de *minDispariy*, *numDisparity* y *window_size* pueden ser calculados en función de las condiciones de operación. A continuación, se muestra los valores asignados a estos parámetros.

Tabla 7:

Valores para parámetros del entorno de trabajo

Parámetro	Valor
minDisparity	80
numDisparity	66
windowSize	3

Los 2 parámetros: P_1 y P_2 , cuyos valores óptimos son seleccionados usando GridSearch (algoritmo de fuerza bruta) que consiste en probar todas las combinaciones posibles para un rango definido de valores (tabla 3) y seleccionar la mejor combinación. A continuación se muestran las 10 mejores y peores combinaciones de parámetros para P_1 y P_2

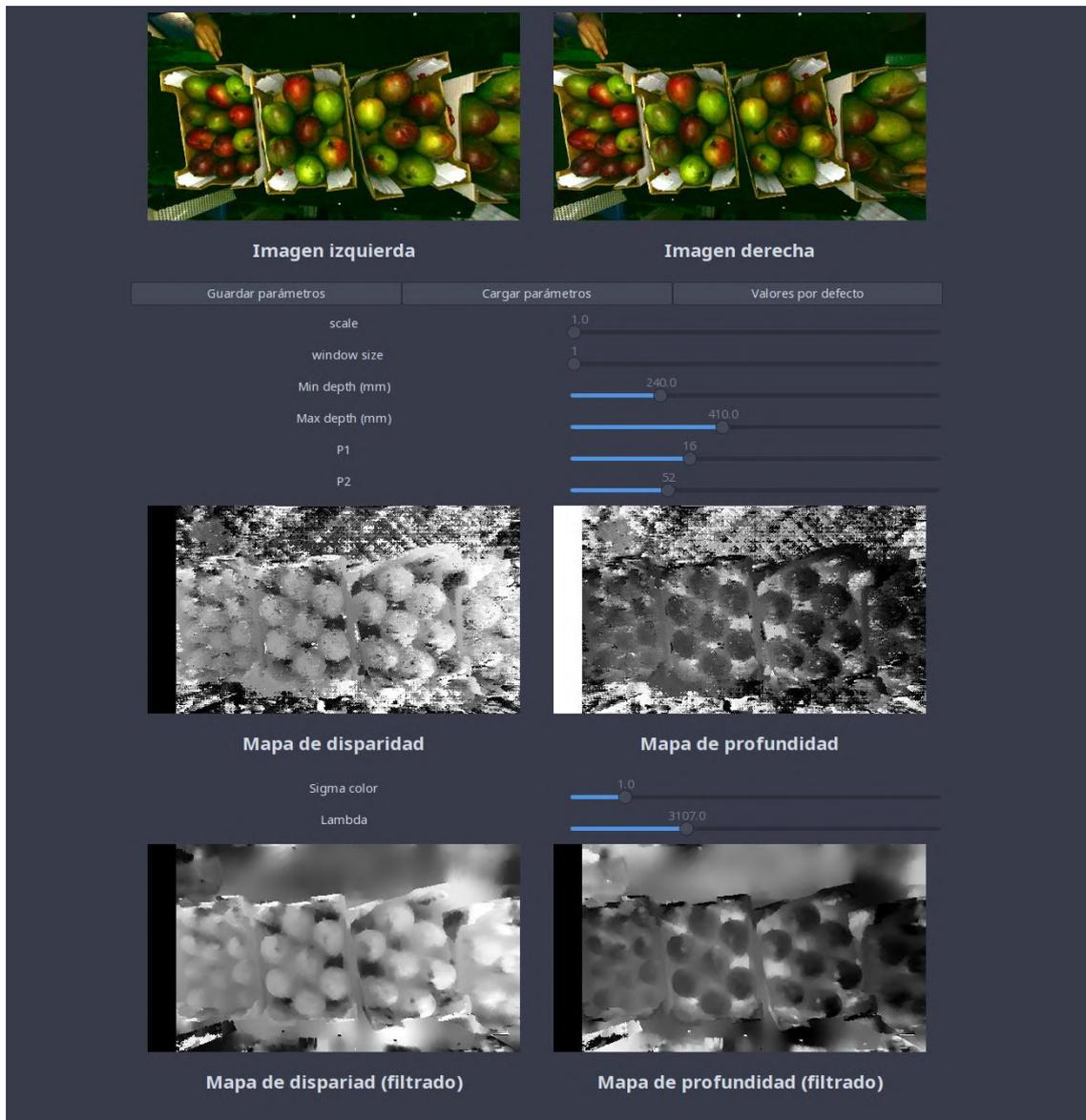
Tabla 8:*Mejores y peores 10 combinaciones de parámetros*

Mejores			Peores		
P_1	P_2	RMS(mm)	P_1	P_2	RMS(mm)
16	64	2.7211	16	16	3.2161
16	56	2.7256	14	16	3.2603
16	48	2.7325	2	16	3.2928
14	64	2.7345	0	16	3.2978
14	56	2.7402	2	8	4.4321
14	48	2.7445	4	8	4.6055
16	40	2.7464	0	8	4.6609
12	64	2.7534	8	8	5.0975
14	40	2.7567	6	8	5.2173
12	56	2.7567	0	0	6.5723

La diferencia entre los errores de la mejor y peor combinación es 3.8467 mm esto implica que los valores de P_1 y P_2 no tienen mayor efecto sobre la precisión del mapa de profundidad.

Para poder visualizar el resultado de las diferentes combinaciones de parámetros se desarrolló una interfaz gráfica que permite la selección de los siguientes parámetros: *scale*, *resolucion_mm*, *min_depth*, *max_depth*, P_1 , P_2 y los parámetros de post filtrado σ_{Color} y *lambda*. Esta interfaz fue desarrollada usando el framework Gtk 3.0 con el lenguaje de programación C++, dentro de la interfaz se pueden visualizar las imágenes izquierda y derecha, los mapas de disparidad (con y sin filtrado WLS) y los mapas de profundidad (con y sin filtrado WLS). La interfaz permite visualizar en tiempo real como cada parámetro afecta los mapas. Esta interfaz también permite exportar un archivo de parámetros que puede usarse para la generación de mapas de profundidad con los parámetros seleccionados en la interfaz.

Figura 14:
Selector de parámetros para mapa de profundidad



3.3. Tiempo de cómputo para la generación de mapa de profundidad

Para aplicar la generación de mapas de profundidad en plantas de empaquetados reales, es necesario que este proceso se pueda realizar en tiempo real, el parámetro que tiene más influencia sobre el tiempo de cómputo de todo el pipeline (figura 8) es el factor de escala, mientras más grande es el factor de escala menor es el tiempo de cómputo, esto se debe a que el factor de escala reduce las dimensiones de la imagen y por lo tanto la cantidad de píxeles

para los cuales se debe calcular la disparidad, el factor de escala también reduce el rango de búsqueda de disparidades ya que al reducir las dimensiones de la imagen los píxeles homólogos se encuentran más cerca.

Para medir el tiempo de cómputo del algoritmo se usó una computadora con sistema operativo Linux, procesador intel core i5 de 7ma generación, 16 GB de RAM DDR4 4200MHz. Para evitar interferencias con demás componentes de software del computador las mediciones de tiempo de ejecución se realizaron en un entorno TTY y se le asignó al proceso prioridad máxima usando el comando *nice -20*, el tiempo de ejecución medido no toma en cuenta el tiempo que se tarda en leer las imágenes ni en guardar los mapas de profundidad en la memoria física (disco duro).

Para el cálculo de mapas de profundidad se utilizaron los parámetros mencionados previamente en las tablas 2 y 7, para los parámetros que no están en las tablas se usó $P_1 = 16$ y $P_2 = 64$. Se definió un área de trabajo desde 240 a 450 mm, y se trabajó con la resolución máxima del sistema de estéreo-visión (1920x1080) píxeles.

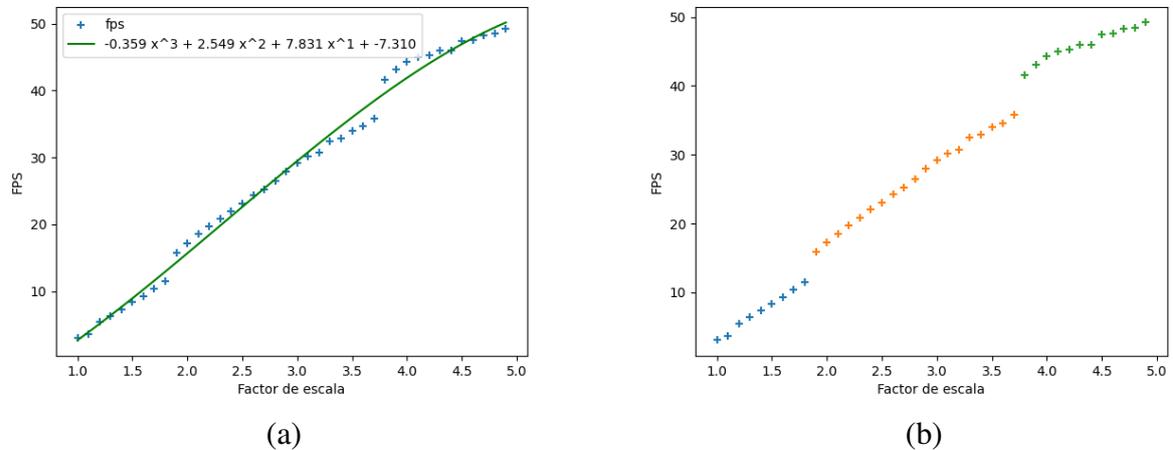
Se midió el tiempo de ejecución del algoritmo para los valores de escala $\{1, 1, 1, 1, 2, \dots, 4.8, 4.9\}$, para cada valor de escala se generaron 20 mapas de profundidad y se mide el tiempo de ejecución para cada 1 de las 20 iteraciones, finalmente se calcula el tiempo de ejecución promedio para cada factor de escala, los fotogramas por segundo equivalen a la cantidad de mapas de profundidad que se pueden generar en un segundo y se define como el recíproco del tiempo de ejecución.

Tabla 9:*Tiempo de cómputo en función del factor de escala*

Scale	Tiempo (s)	FPS	Scale	Tiempo (s)	FPS	Scale	Tiempo (s)	FPS
1.0	0.329	3.043	2.3	0.048	20.879	3.6	0.029	34.631
1.1	0.273	3.664	2.4	0.045	22.030	3.7	0.028	35.790
1.2	0.183	5.470	2.5	0.043	23.094	3.9	0.023	43.105
1.3	0.158	6.345	2.6	0.041	24.332	4.0	0.023	44.345
1.4	0.137	7.317	2.7	0.040	25.262	4.1	0.022	45.022
1.5	0.120	8.355	2.8	0.038	26.502	4.2	0.022	45.263
1.6	0.108	9.245	2.9	0.036	27.971	4.3	0.022	45.978
1.7	0.097	10.329	3.0	0.034	29.180	4.4	0.022	45.932
1.8	0.087	11.469	3.1	0.033	30.198	4.5	0.021	47.436
1.9	0.063	15.816	3.2	0.033	30.745	4.6	0.021	47.577
2.0	0.058	17.211	3.3	0.031	32.492	4.7	0.021	48.258
2.1	0.054	18.551	3.4	0.030	32.855	4.8	0.021	48.504
2.2	0.051	19.694	3.5	0.029	33.981	4.9	0.020	49.222

El algoritmo para la generación de mapas de profundidad tiene una complejidad temporal $O(\frac{W}{s} \cdot \frac{H}{s} \cdot \frac{D}{s})$ donde W, H son las dimensiones de la imagen, D es el rango de búsqueda de disparidades, la complejidad temporal se puede reescribir como $O(\frac{W \cdot H \cdot D}{s^3})$ de esta forma es evidente que el tiempo de cómputo disminuye cúbicamente con el factor de escala, por lo tanto los $fps = \frac{1}{tiempo_computo}$ pueden ser expresados como un polinomio de tercer grado en función de la escala. El polinomio de tercer grado que mejor describe la relación entre fps y factor de escala es $-0.3579x^3 + 2.5495x^2 + 7.8311x^1 - 7.3100$ el cual se muestra graficado en la figura 15a.

Figura 15:
FPS y factor de escala



La figura 15b muestra como los datos de FPS se separan en 3 grupos, el incremento de fps que se da al final de cada grupo es mayor que el incremento entre puntos del mismo grupo, esto se debe a que el rango de búsqueda de disparidades debe ser un número divisible para 16 esto ocasiona que el número de disparidades se reduzca de manera escalonada, mientras que la resolución de la imagen se reduce de manera continua en función del factor de escala, los 3 grupos visibles en el gráfico corresponden a 3 diferentes número de disparidades 46, 32, 16px. Esto implica que dentro de cada grupo el aumento de FPS se debe únicamente a la reducción de resolución, mientras en los cambios de grupo el aumento en fps se debe a la reducción del número de disparidades y la reducción de la resolución.

Capítulo 4

4. Conclusiones y Recomendaciones

4.1. Conclusiones

- Se logró implementar un protocolo de calibración estereoscópica funcional, el cual permite obtener calibraciones con un error menor a 1px para el sistema de estéreo visión descrito en la sección 2.2.
- Se logró determinar que para el sistema de adquisición descrito en la sección 2.2 es posible generar mapas de profundidad por precisión de 2.72mm para un área de trabajo de 240 a 450 mm de profundidad.
- Se logro demostrar para un área de trabajo de 240 a 450 mm de profundidad es posible generar mapas de profundidad en tiempo real si se reduce lo suficiente el factor de escala.
- Se logró implementar interfaces gráficas, utilidades de línea de comando y una librería para la captura, calibración y generación de mapas de profundidad.

4.2. Recomendaciones

- Para implementaciones futuras se recomienda incrementar la textura de la escena, si no es posible aumentar la textura de los objetos en la escena modificando sus colores, se recomienda proyectar sobre la escena un patrón de alta textura.
- Para la generación de mapas de profundidad se recomienda reducir el factor de escala tanto como sea posible tomando en cuenta las pérdidas de resolución, además se puede reducir el área de búsqueda eliminando regiones donde no interesa conseguir datos de profundidad.
- La precisión también puede ser mejorada acercando la cámara al objeto que se desee medir tanto como sea posible para la aplicación deseada, reducir el campo de visión de las cámaras también es recomendable para maximizar la precisión.

Referencias

- [1] A. Broggi, P. Grisleri, and P. Zani, “Sensors technologies for intelligent vehicles perception systems: A comparison between vision and 3d-lidar,” in *16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)*, IEEE, Oct. 2013. DOI: 10.1109/itsc.2013.6728344. [Online]. Available: <http://dx.doi.org/10.1109/ITSC.2013.6728344>.
- [2] V. Centing, “Approximating material area, volume, and velocity for belt conveyor system applications using 3d depth sensor technology,” Ph.D. dissertation, Your University, 2023.
- [3] R. Szeliski, *Computer Vision: Algorithms and Applications*. Springer, 2011, ISBN: 978-1848829343.
- [4] U. of Minnesota. “Localization: Vanishing points.” Lecture slides from the University of Minnesota. (Year of the lecture), [Online]. Available: [https://www-users.cse.umn.edu/~hspark/CSci5980/Lec4_Localization\(VanishingPoint\).pdf](https://www-users.cse.umn.edu/~hspark/CSci5980/Lec4_Localization(VanishingPoint).pdf).
- [5] V. H. Diaz-Ramirez, M. Gonzalez-Ruiz, V. Kober, and R. Juarez-Salazar, “Stereo image matching using adaptive morphological correlation,” *Sensors*, vol. 22, no. 23, 2022, ISSN: 1424-8220. DOI: 10.3390/s22239050. [Online]. Available: <https://www.mdpi.com/1424-8220/22/23/9050>.
- [6] G. Bradski, “The OpenCV Library,” *Dr. Dobb’s Journal of Software Tools*, vol. 25, no. 11, pp. 120–126, 2000. [Online]. Available: <https://opencv.org/>.
- [7] C. Harris and M. Stephens, “A combined corner and edge detector,” *Proceedings of the Alvey Vision Conference*, pp. 147–151, 1988. DOI: 10.5244/C.2.23.
- [8] D. A. Forsyth and J. Ponce, *Computer Vision: A Modern Approach*. Prentice Hall, 2002, ISBN: 978-0130851987.

- [9] H. Hirschmuller, “Stereo processing by semiglobal matching and mutual information,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 2, pp. 328–341, Feb. 2008, ISSN: 0162-8828. DOI: 10.1109/tpami.2007.1166. [Online]. Available: <http://dx.doi.org/10.1109/TPAMI.2007.1166>.
- [10] S. Birchfield and C. Tomasi, “Depth discontinuities by pixel-to-pixel stereo,” in *Sixth International Conference on Computer Vision (IEEE Cat. No.98CH36271)*, ser. ICCV-98, Narosa Publishing House. DOI: 10.1109/iccv.1998.710850. [Online]. Available: <http://dx.doi.org/10.1109/ICCV.1998.710850>.
- [11] P. Domingos, “A few useful things to know about machine learning,” *Communications of the ACM*, vol. 55, no. 10, pp. 78–87, Oct. 2012, ISSN: 1557-7317. DOI: 10.1145/2347736.2347755. [Online]. Available: <http://dx.doi.org/10.1145/2347736.2347755>.
- [12] P. Liashchynskyi and P. Liashchynskyi, *Grid search, random search, genetic algorithm: A big comparison for nas*, Dec. 2019.
- [13] BestHQwallpapers. “Descargar fondos de pantalla ornements floraux colorés, 4k, vagues abstraites, créatifs, fonds arc-en-ciel, art floral abstrait, arrière-plans abstraits, motifs ondulés libre.” *Imágenes Fondos de Descarga Gratuita*, besthqwallpapers.com. (2021), [Online]. Available: <https://besthqwallpapers.com/es/abstracto/ornements-floraux-color%C3%A9s-4k-vagues-abstraites-cr%C3%A9atifs-fonds-arc-en-ciel-167836> (visited on 01/17/2024).