



UNIVERSIDAD DEL AZUAY
FACULTAD DE CIENCIA Y TECNOLOGÍA
ESCUELA DE INGENIERÍA ELECTRÓNICA

**“Implementación de un Sistema de Posicionamiento Óptico
para un Robot Móvil dentro de un Entorno de Trabajo.”**

Trabajo de graduación previo a la obtención del título de:

INGENIERO ELECTRÓNICO

Autor:

PABLO FERNANDO ARBITO CHICA

Director:

Mst. GABRIEL DELGADO OLEAS

CUENCA, ECUADOR

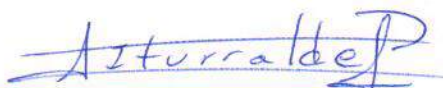
2018

Implementación de un Sistema de Posicionamiento Óptico para un Robot Móvil dentro de un Entorno de Trabajo.

RESUMEN

En el presente trabajo se describe el proceso de implementación de un prototipo de posicionamiento óptico, cuya misión es la de ubicar robots móviles y su orientación, con la ayuda de marcadores, en un área de trabajo. Así se hace énfasis en la etapa de diseño, tomándose en cuenta aspectos constructivos (limitaciones físicas, propiedades del entorno, adaptación de los periféricos, definición de campos de visión y demás), al igual que detalles de desarrollo (sistema operativo, kernels, protocolos de comunicación, anchos de banda entre otros), y caracterización del prototipo, terminando con el análisis de datos obtenidos.

Palabras clave: OpenCV, stitching, campo de visión, calibración de cámaras, posicionamiento, robot móvil, Linux.



Ing. Daniel Iturralde Piedra. Ph.D.
Coordinador de Carrera.



Mst. Gabriel Delgado Oleas.
Director de Trabajo de Titulación



Fernando Arbito Chica
Autor

Implementation of an Optical Positioning System within a Work Environment for a Mobile Robot.

ABSTRACT

This work describes the process to implement a prototype for optical positioning. The aim of the research was to locate mobile robots and their orientation within a work area with the help of markers. The design stage was emphasized and took into account construction aspects (physical limitations, environmental properties, adaptation of peripherals, definition of fields of vision and others), development details (operating system, kernels, communication protocols, bandwidths) and the characterization of the prototype. This work ended with an analysis of the data.

Keywords: OpenCV, stitching, field of vision, calibration of cameras, positioning, mobile robot, Linux.



Ing. Daniel Iturralde Piedra. Ph.D.
Faculty Coordinator



Mst. Gabriel Delgado Oleas
Thesis Director



Fernando Arbito Chica
Author

Implementación de un Sistema de Posicionamiento Óptico para un Robot Móvil dentro de un Entorno de Trabajo

Pablo Fernando Arbito Chica
Escuela de Ingeniería Electrónica
Universidad del Azuay
Cuenca, Ecuador
teslamillingec@outlook.com

Resumen—En el presente trabajo se describe el proceso de implementación de un prototipo de posicionamiento óptico, cuya misión es la de ubicar robots móviles y su orientación, con la ayuda de marcadores, en un área de trabajo. Así se hace énfasis en la etapa de diseño, tomándose en cuenta aspectos constructivos (limitaciones físicas, propiedades del entorno, adaptación de los periféricos, definición de campos de visión y demás), al igual que detalles de desarrollo (sistema operativo, kernels, protocolos de comunicación, anchos de banda entre otros) y caracterización del prototipo, terminando con el análisis de datos obtenidos. [1].

Index Terms—OpenCV, stitching, campo de visión, calibración de cámaras, posicionamiento, robot móvil, linux.

I. INTRODUCCIÓN Y ESTADO DEL ARTE

Dentro de varias áreas, como la robótica, control, animación 3d entre otras, existe la necesidad de capturar la ubicación, orientación, movimiento y trayectoria de diferentes partes que conformen sistemas en movimiento dentro de un espacio definido [2].

Para lograr este objetivo se ocupan diferentes tecnologías, siendo las principales encoders, servomotores, sistemas inerciales, ópticos, *SLAM* (*Simultaneous Localization And Mapping*) y *LIDAR* (*Laser Imaging Detection And Raging*), entre otras [3]. Sin embargo, la implementación de estas resultan costosas; además, la complejidad de ciertos sistemas limita las tecnologías que pueden aplicarse así como la alta capacidad de procesamiento por subsistema que se requiere para su interacción con el entorno, personas y otros subsistemas [3].

Para afrontar este problema se ha profundizado en el uso de tecnologías ópticas dada su gran adaptabilidad, donde podemos nombrar trabajos pertinentes como:

- En el trabajo [4] se utilizó un sistema de posicionamiento espacial de alta precisión para determinar el error de posicionamiento de un robot móvil.

- A su vez en [5] se presenta el concepto de un sistema flexible de asistencia para aplicaciones industriales usando localización mediante múltiples cámaras.
- En [6] se menciona una aproximación para hacer planeamiento adaptativo para reconocer el entorno de un robot móvil.
- Cabe señalar que [7] presenta un método para navegación y evitar obstáculos para robots móviles en un ambiente dinámico desconocido.
- En [8] se demuestra que la localización de objetos en movimiento es posible en un ambiente con múltiples cámaras.

En base a estos trabajos se propuso el presente proyecto con la finalidad de crear una herramienta prototipo capaz de ubicar objetos y su orientación en un área de trabajo cuya extensión requiera el uso de múltiples cámaras. Así también se contempló aminorar costos al optimizar el espacio designado así como los periféricos, sistema operativo y programas auxiliares.

II. IMPLEMENTACIÓN DEL PROTOTIPO

II-A. Consideraciones de diseño

El prototipo implementado en El Laboratorio de Telecomunicaciones de la Universidad del Azuay consta de un área de $25,37m^2$ y una altura de $2,43m$ definida por un pasillo de entrada y la sección del laboratorio propiamente dicha.

La iluminación del área está provista por 5 lámparas distribuidas en el lugar con tres barras led cada una, instaladas sobre un cielo raso removible, además el piso es de tipo laminado color madera.

Dentro del mismo se encuentra ubicado el computador donde se desarrollarán los elementos de software. Sus



Figura 1: Laboratorio.

características principales son:

- CPU Intel Core i7-6700K 4.00GHz.
- Memoria RAM de 16GB.
- Controlador USB 3.0.
- Tarjeta gráfica NVidia GeForce GTX 1060 3GB.



Figura 2: Primer plano de las cámaras utilizadas.

Las cámaras utilizadas son las *PS3 Eye Cam* de *Sony Computer Entertainment America Inc.* [9], cuyas especificaciones son:

- 56° a 75° de campo de visión diagonal de acuerdo al zoom.
- Distorsión menor al 1% con zoom fijo.
- Velocidad de captura de hasta 60fps a una resolución de 640×480 .
- Velocidad de captura de hasta 120fps a una resolución de 320×240 .
- Comunicación a través de protocolo USB2.0.
- Capacidad para capturar video sin compresión en formato *YUYV4:2:2* (compresión 2:1 en horizontal y 1:1 en vertical [10]), o con compresión *JPEG*.

II-B. Adecuación del entorno y periféricos

Para la adecuación del entorno se consideraron los siguientes puntos:

- Área disponible y altura.
- Campo de visión de las cámaras y área utilizada.
- Conexión de los periféricos.
- Otras adecuaciones.

II-B1. Área disponible y altura: El área útil es de $25,37\text{m}^2$ cuya disposición se indica en la figura 3, detallándose la distribución de puestos de trabajo, lámparas y el computador. La altura disponible es de $2,43\text{m}$.

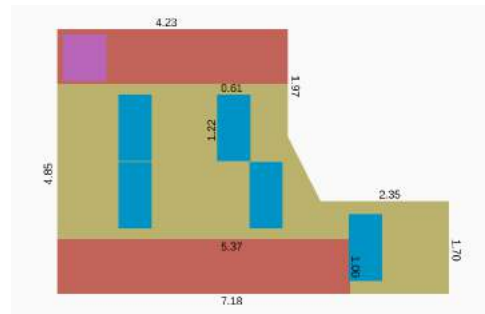


Figura 3: Disposición del laboratorio. En amarillo el laboratorio, en rojo los puestos de trabajo, en azul las lámparas led y en morado el computador.

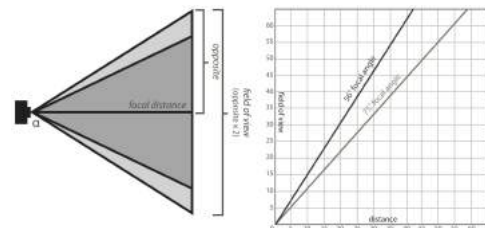


Figura 4: Campo de visión de las cámaras *PS3 Eye Cam* de acuerdo al fabricante.

II-B2. Campo de visión de las cámaras y área utilizada: El área utilizada y que delimita el entorno de trabajo está marcado por el campo de visión de las cámaras y la altura de instalación.

Así, se procede de manera experimental a obtener el campo de visión vertical, horizontal y a calcular el campo de visión diagonal como comprobación de los datos del fabricante (figura 4).

Para esto se coloca un objeto de longitud conocida (1m) paralelo al plano de la cámara, haciendo que el objeto apenas entre en el cuadro de la imagen y se procede a medir la distancia de la cámara a dicho objeto. Este proceso se repite

en el sentido horizontal y vertical en las 2 posiciones del *zoom* como se indica en la figura 5. En la tabla I se muestran los resultados efectuados en las cuatro cámaras.

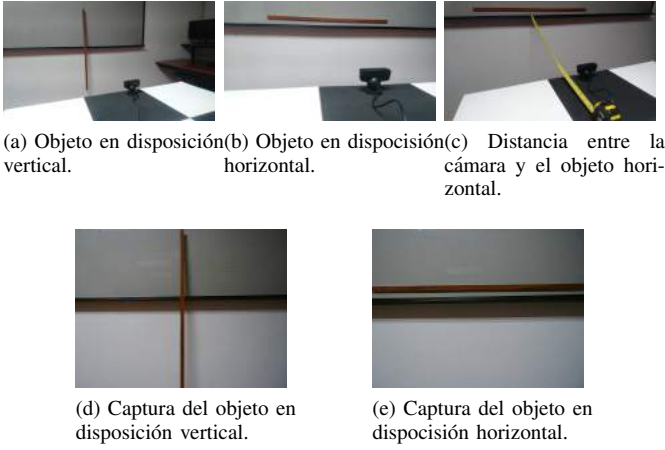


Figura 5: Toma experimental de datos para el cálculo del campo de visión.

Tabla I: Distancia al objeto

	cam1	cam2	cam3	cam4
zoom mínimo				
dist hor	0.81	0.81	0.82	0.82
dist ver	1.08	1.08	1.12	1.12
zoom máximo				
dist hor	1.15	1.15	1.19	1.19
dist ver	1.55	1.55	1.61	1.61

La igualdad entre pares de valores de la tabla I es debido a que se utilizaron dos versiones del mismo modelo de cámaras.

Por trigonometría, se utilizaron las fórmulas 1 y 2 para calcular el campo de visión de las cámaras.

$$\alpha = 2 \arctan \left(\frac{loC/2}{dH} \right) \quad (1)$$

$$\beta = 2 \arctan \left(\frac{loC/2}{dV} \right) \quad (2)$$

Siendo α y β los ángulos del campo de visión horizontal y vertical, loC la longitud del objeto conocido, y dH , dV las distancias a dicho objeto en disposición horizontal y vertical.

El campo de visión diagonal se calcula con las fórmulas 3 y 4.

$$\gamma = 2 \arctan \frac{\sqrt{(dT \times \tan(\alpha/2))^2 + (dT \times \tan(\beta/2))^2}}{dT} \Big|_{dT=1,00} \quad (3)$$

$$\gamma = 2 \arctan \sqrt{\tan(\alpha/2)^2 + \tan(\beta/2)^2} \quad (4)$$

Siendo la ecuación 4 la simplificación de la ecuación 3, donde γ es el campo de visión diagonal, α y β los resultados de las ecuaciones 1 y 2, y dT una constante arbitraria, que representa la distancia de la cámara a una superficie de incidencia. Los resultados se muestran en la tabla II.

Tabla II: Campos de visión promedio

zoom mínimo					
α	63.06°	β	48.90°	γ	74.73°
zoom máximo					
α	46.29°	β	35.13°	γ	56.02°

Con estos resultados se comprueba las especificaciones del fabricante y se obtiene los campos de visión vertical y horizontal, datos con los que se calcula el área que cubre cada cámara una vez instalada usando las ecuaciones 5 y 6.

$$hor = 2h \tan(\alpha/2) \quad (5)$$

$$ver = 2h \tan(\beta/2) \quad (6)$$

Siendo hor la longitud horizontal del campo de visión, ver la longitud vertical del campo de visión y h la altura de instalación.

Para el correcto funcionamiento de los algoritmos que se detallan en la sección II-C5, es necesaria la superposición parcial de los campos de visión de las cámaras entre sí, habiéndose utilizado una superposición de aproximadamente 0,70m en vertical y horizontal.

Para obtener la máxima área utilizada se disponen las cámaras en el zoom máximo y se las posicionaron como se indica en la figura 6, siendo esto beneficioso ya que obtenemos un mayor detalle de las imágenes capturadas por las cámaras cumpliendo con lo anterior estipulado.

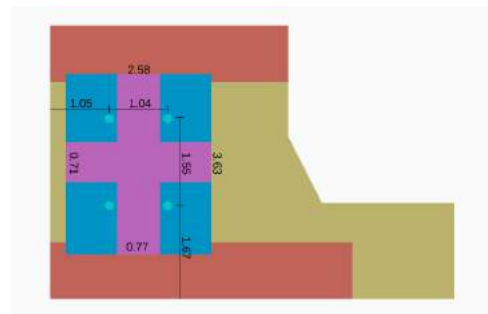


Figura 6: Campo de visión teórico. En azul el campo de visión total y en morado las áreas superpuestas.

Así el área utilizada es de $9,35m^2$ distribuidos en una arista vertical de $3,63m$ y una horizontal de $2,58m$. Esto supone un 34% y 50% de superposición horizontal y vertical de acuerdo con el margen estipulado anteriormente.

Se debe tener en cuenta que los valores anteriores son teóricos y pueden variar de acuerdo a factores como:

- Falta de paralelismo entre el plano de la cámara con la superficie de trabajo.
- Incorrecta alineación entre los planos de coordenadas de las cámaras con el plano del laboratorio.
- Imprecisiones al momento de medir la posición y separación de las cámaras.
- Variaciones en la altura real de las cámaras debido al método de instalación utilizado.

El montaje de las cámaras se realiza en la parte posterior del cielo raso, aprovechándose toda la altura estipulada y eliminándose la necesidad de crear un soporte para las mismas. Para este fin se abre un agujero en las posiciones descritas en la figura 6 y en la figura 7 se muestra la instalación de una de las cuatro cámaras.



(a) Cámara instalada, vista frontal. (b) Cámara instalada, vista posterior

Figura 7: Vista frontal y posterior de una cámara instalada.

II-B3. Conexión de los periféricos: El protocolo utilizado para interconectar el sistema es el estándar USB, teniendo las cámaras conectividad con la versión 2.0 y el computador la 3.1.

Tabla III: Resumen de características estándar USB [11].

Versión	Ancho de banda	Longitud máxima	Retrocompatibilidad
1.0	12Mbps	2m - 5m	
2.0	480Mbps		1.0
3.0	3Gbps		1.0 - 2.0
3.1	8Gbps		1.0 - 2.0 - 3.0
3.2	16Gbps		1.0 - 2.0 - 3.0 - 3.1

Como se puede inferir de los datos de la figura 6, el computador se encuentra a una distancia de $6,27m$ de la

cámara más distante, lo que no permite una conexión directa debido a las especificaciones del estándar (tabla III) así como al largo del cable de las cámaras ($2m$).



(a) Extensión USB 3.0.

(b) HUB USB 3.0.



(c) Conexión de los HUBs y las cámaras

Figura 8: Elementos utilizados en la conexión de los elementos.

Por esta razón se añade una extensión USB 3.0 (2 HUBs USB en serie) de $10m$, así como un HUB USB 3.0 de cuatro puertos, los cuales se muestran en la figura 8.

Estos elementos fueron conectados en una configuración tipo estrella con las cámaras en sus extremos tal como lo manifiesta la arquitectura del estándar USB [11].

Esta configuración permite a los HUBs conectados consecutivamente tener el ancho de banda completo. La arquitectura del estándar se muestra en la figura 9.

II-B4. Otras adecuaciones: Posterior a las adecuaciones descritas se realizan pruebas de comunicación de las cámaras para comprobar su correcto funcionamiento, teniendo resultados positivos de las mismas.

Sin embargo se nota que debido al material del piso sumado a la posición de las lámparas respecto de las cámaras, estas captan reflejos, dando la impresión de una iluminación no homogénea como se observa en la figura 10.

Ante la necesidad de eliminar dicho factor se adecua una superficie con las siguientes características:

- Grado de reflexión mínimo o nulo.

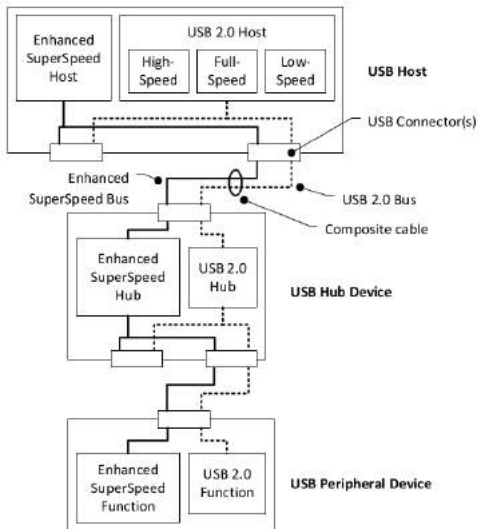


Figura 9: Arquitectura del estándar USB 3.x [11].

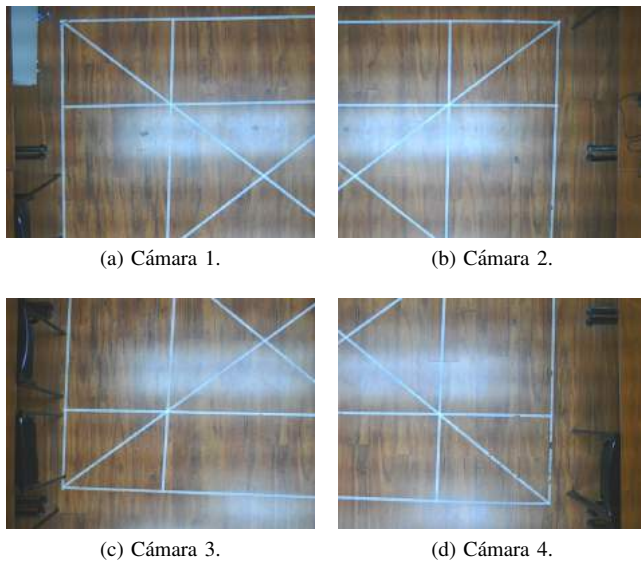


Figura 10: Imágenes capturadas antes de la adecuación del entorno.

- Desmontable.
- Sirva de elemento de soporte.
- Fácil limpieza.

El material elegido fue el foamy, debido a que se trata de un polímero termoplástico de fácil manipulación (cortar y pegar), lavable, no tóxico, liviano, no reflectivo y la presentación comercial incluye una gran gama de colores [12].

Los colores utilizados en la superficie antireflectante de foamy son blanco y negro cortados en recuadros de $0,30m \times 0,30m$ pegados alternadamente sobre láminas de cartón corrugado de $0,90m \times 1,20m$ formando paneles de fácil



Figura 11: Paneles de foamy.

montaje y manejo como se muestra en la figura 11. Los resultados se pueden apreciar en las figuras 12 y 13.

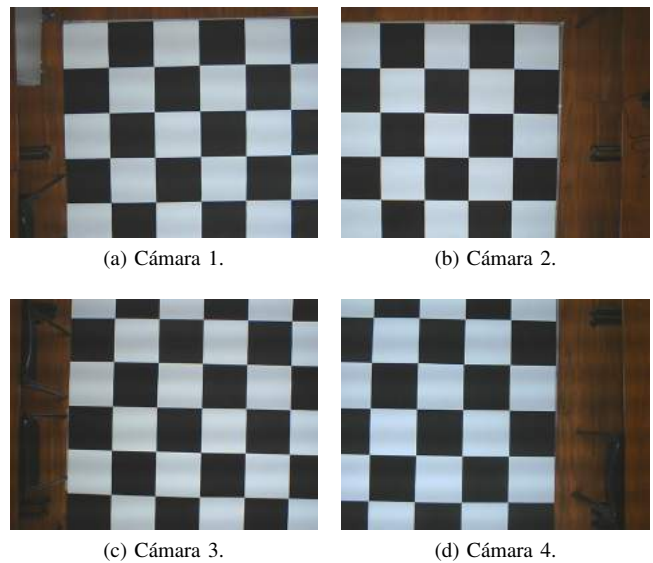


Figura 12: Imágenes capturadas después de la adecuación del entorno.

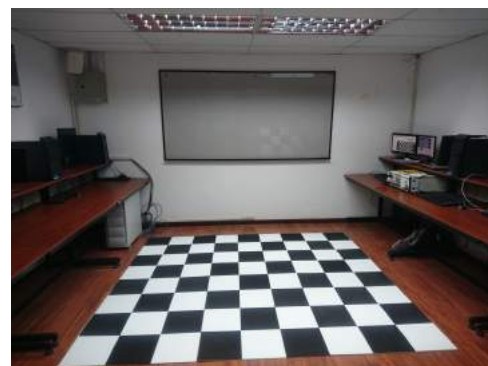


Figura 13: Ensamblado de paneles de foamy en el laboratorio.

II-C. Adecuación y desarrollo de software

Dentro del desarrollo del sistema se tuvo en cuenta factores externos (distorsiones, posicionamiento de las cámaras) como internos (sistema operativo, detección de periféricos entre otros), buscando la optimización de recursos y el correcto funcionamiento del prototipo. Para esto se trabaja con los siguientes puntos:

- Sistema operativo y kernel.
- Kernel: detección y funcionamiento de las cámaras.
- Scripts y modificaciones del kernel.
- Calibración intrínseca de las cámaras.
- Calibración extrínseca de las cámaras.
- Desarrollo del sistema de posicionamiento.

A continuación se describe cada una de estas operaciones.

II-C1. Sistema operativo y kernel: Las opciones de sistema operativo fueron *Windows* y *Linux*, cuyo análisis detallamos a continuación.

Windows® es un sistema operativo con licencia comercial, con lo que las modificaciones y el acceso a funciones del kernel estén restringidas de manera nativa, y los programas que lo permiten son muy limitados y de pago. Cabe mencionar que los entornos de desarrollo también son de pago [13].

Linux es un sistema operativo con licencia *GPL-2.0 (General Public License)* que permite el acceso y modificación de todas las partes del kernel, realizándose mediante comandos ejecutados por el usuario o guardados en archivos de texto para que el sistema operativo los ejecute pertinentemente, así como permitir su libre distribución. A su vez dispone de multitud de entornos de desarrollo libres de pago [14].

Por estas razones el sistema operativo elegido fue *Ubuntu* [15], el cual está basado en *Debian/GNULinux* [16], que dispone de programas y kernels distribuidos en forma de paquetes, los que son revisados continuamente por el equipo de *Canonical* [17], asegurando su compatibilidad y ejecución, permitiendo aumentar la funcionalidad del sistema operativo sin mayor esfuerzo.

El kernel de baja latencia utilizado es el 4.15.0-36-*lowlatency* que da prioridad de ejecución a los procesos de acuerdo a su procedencia, teniendo mayor prioridad los *Dispositivos de Interfaz Humana (HID)* como son las cámaras en nuestro caso [18].

II-C2. Kernel: detección y funcionamiento de las cámaras: Las herramientas y funciones que se utilizaron son:

- *lshw*: herramienta para la extracción de información detallada de la configuración de todo hardware conectado

a la máquina [19].

- *lsusb*: utilidad que muestra información referente a los buses USB en el sistema y los dispositivos conectados a estos [20].
- *usbmon*: herramienta del kernel que sirve para recolectar rastros de las operaciones de entrada/salida de los puertos USB [21].
- *v4l2-ctl*: aplicación que permite el control de todos los aspectos de los dispositivos que hagan uso de los drivers *video4linux* [22].

II-C2a. Buses USB: El resumen de resultados de la herramienta *lshw* se muestra en la tabla IV, donde se indican los *buses de direccionamiento* utilizados desde la computadora hasta las cámaras, coincidiendo con los resultados de la herramienta *lsusb*. El detalle de salida de estas herramientas se muestra en los apéndices A y B.

Tabla IV: Buses USB.

Nivel	Id	Bus	Descripción
0	usb controller	———	controlador usb
1	usbhost:0	3	hub usb
2	usbhub	3:1	puertos de la computadora
3	usbhub	3:1:1	extensión usb
4	usbhub	3:1:1:1	hub de 4 puertos
5	usb0	3:1:1:1:1	cámara 1
5	usb1	3:1:1:1:2	cámara 2
5	usb2	3:1:1:1:3	cámara 3
5	usb3	3:1:1:1:4	cámara 4

II-C2b. Tipo de transferencia y ancho de banda:

Con la información de los *buses* se procede a usar la herramienta *v4l2-ctl* que lista y modifica los parámetros de funcionamiento que disponen las cámaras. El resumen se muestra en la tabla V y se detalla en el apéndice C.

El *formato de píxel* hace referencia a la forma en que se codifica el color, así como a ciertas características del mismo. El formato *YUYV 4:2:2* codifica el color en un canal *luma* y en dos canales *chroma*, teniendo una tasa de submuestreo 2:1 en horizontal y 1:1 en vertical; su *profundidad de píxel* es de 16 *bits* [10], y se utilizó una resolución de 640 x 480.

Con estos datos se calcula los anchos de banda haciendo uso de la ecuación 7.

$$Bw = resV \times resH \times pD \times fps \quad (7)$$

Donde *Bw* es el ancho de banda, *resV* y *resH* son las dimensiones vertical y horizontal de la imagen en píxeles, *pD* la profundidad de píxel y *fps* la cantidad de cuadros

Tabla V: Parámetros de funcionamiento de las cámaras.

Parámetro	Valor
Formato de pixel	YUYV 4:2:2
Resolución	FPS
320 x 240	187
	150
	137
	125
	100
	75
	60
	50
	37
	30
640 x 480	60
	50
	40
	30
	15

por segundo. Los resultados teóricos se detallan en la tabla VI.

Tabla VI: Ancho de banda teórico.

Resolución	fps	Ancho de banda (Mbps)
640 x 480	60	295
	50	246
	40	197
	30	147
	15	74

Posteriormente se procede a comprobar estos datos con la herramienta *usbmon*, la cual adicionalmente identifica el tipo de transferencia utilizado como *Bulk*, el cual asegura al dispositivo el máximo ancho de banda sacrificando la periodicidad en el envío de datos [11]. Los resultados se muestran en la tabla VII y su detalle en el apéndice D.

Tabla VII: Ancho de banda medido.

Resolución	FPS	Ancho de banda (Mbps)
640 x 480	60	298
	50	248
	40	199
	30	149
	15	74

Ha de tenerse en cuenta que la arquitectura del estándar USB, por razones de retrocompatibilidad, consta de circuitos independientes para las versiones 3.0 y 2.0 (figura 9), por lo que el sistema tiene un ancho de banda disponible de *480Mbps* como se muestra en la tabla III [11].

Teniendo en cuenta los datos anteriores se infiere que el ancho de banda necesario para el funcionamiento de las cuatro

cámaras del sistema es aproximadamente de cuatro veces el indicado en la tabla VII, lo cual nos limita el funcionamiento a *15fps* utilizando un ancho de banda aproximado de *300Mbps*.

II-C3. Scripts y modificaciones del kernel:

II-C3a. Identificación de las cámaras: Tal y como se indica en [23], el kernel asigna los nombres de acceso a los dispositivos de acuerdo al orden de detección. Esto ocasiona que las cámaras no siempre tengan el mismo nombre de acceso generando incertidumbre al acceder a una cámara en específico por parte del prototipo.

Por esta razón se utiliza la herramienta *udev* que proporciona información al software de sistema sobre eventos, maneja los permisos de nodos de dispositivos, crea accesos directos en el directorio */dev/* entre otras funciones [23].

Con esta herramienta se configura el kernel para que, independientemente del orden de detección de las cámaras, se asigne un nombre definido de acuerdo a los parámetros de éstas. Esto se logra agregando una *regla* basada en parámetros únicos de la cámara, utilizándose en el prototipo el *bus de direccionamiento* obtenido en la sección II-C2a. El detalle se muestra en el apéndice E.

II-C3b. Configuración de los parámetros de las cámaras: Las cámaras *PS3 Eye Cam* tienen varios parámetros de configuración dedicados a mejorar la calidad de las imágenes capturadas, siendo ajustados automáticamente por defecto [9].

Debido a que las cámaras ajustan sus parámetros individualmente se genera discordancia entre el color, iluminación y otros parámetros de las imágenes capturadas.

Por esto se utiliza la aplicación *v4l2-ctl*, la cual obtiene y configura los parámetros que el *driver v4l* detecte de la cámara en cuestión [22].

En la tabla VIII se muestra los parámetros detectados, su rango y el valor utilizado, habiéndose grabado en un archivo para que el kernel lo cargue y ejecute con el inicio de sesión del sistema. El detalle se muestra en el apéndice F.

II-C4. Calibración intrínseca de las cámaras: Ante la masificación tecnológica y la creciente demanda de mayor funcionalidad en los dispositivos electrónicos, la industria ha respondido produciendo componentes de menor costo, aún a expensas de su exactitud y precisión [24].

Por esta razón los dos tipos principales de distorsión que afectan a toda cámara se ven amplificadas de acuerdo a la marca, modelo y precio. Estas distorsiones son:

Tabla VIII: Parámetros de funcionamiento de las cámaras.

Parámetro	Rango	Valor utilizado
Brillo	0 : 255	0
Contraste	0 : 255	40
Saturación	0 : 255	70
Hue	-90 : 90	0
Balance automático de blancos	0 : 1	0
Exposición	0 : 255	30
Ganancia automática	0 : 1	0
Ganancia	0 : 63	0
Espejo vertical	0 : 1	0
Espejo horizontal	0 : 1	0
Frecuencia de línea eléctrica	0 : 1	0
Nitidez	0 : 63	10
Exposición automática	0 : 1	1

- *Distorsión radial*: provoca que los objetos se vean curvados a medida que se alejan del centro óptico [25].
- *Distorsión tangencial*: debida a los errores de alineación entre el lente de captura y el plano de la imagen [25].

A estas se las conoce como *distorsiones intrínsecas* de la cámara, siendo obligatoria su corrección en cualquier aplicación de visión por computadora. [25]. Su modelo matemático se ve representado en las ecuaciones 8, 9, 10 y 11 [25].

$$x_{dist\ rad} = x(1 + k_1r^2 + k_2r^4 + k_3r^6) \quad (8)$$

$$y_{dist\ rad} = y(1 + k_1r^2 + k_2r^4 + k_3r^6) \quad (9)$$

$$x_{dist\ tan} = x + [2p_1xy + p_2(r^2 + 2x^2)] \quad (10)$$

$$y_{dist\ tan} = y + [2p_1(r^2 + 2y^2) + 2p_2xy] \quad (11)$$

Con estas se procede a calcular la *distancia focal* y el *centro óptico* de la cámara, siendo representados de manera matricial como se indica en la ecuación 12. [25]

$$matriz\ intrínseca = \begin{Bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{Bmatrix} \quad (12)$$

La aplicación de esta matriz sobre las imágenes capturadas por la cámara corrige sus distorsiones, conociéndose a este proceso como *calibración intrínseca* de la cámara.

Para el cálculo de la *matriz intrínseca* de cada cámara se utilizan funciones de la librería *OpenCV*, la que hace uso de múltiples capturas de una cuadrícula de dimensiones

conocidas para este fin [25]. El algoritmo utilizado se lo muestra en la figura 14.

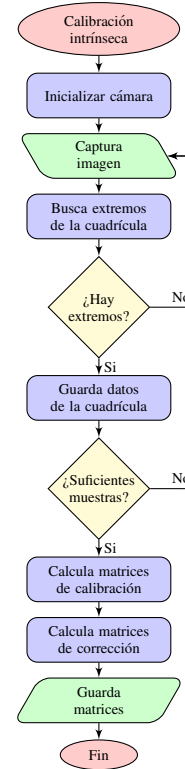
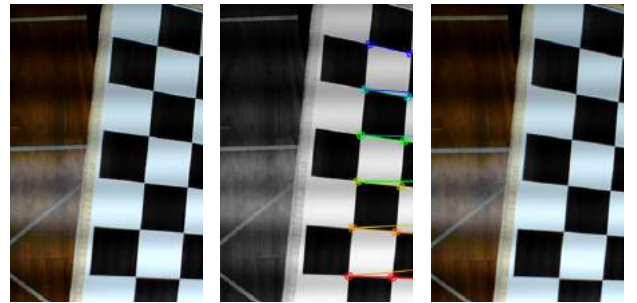


Figura 14: Diagrama de flujo: calibración intrínseca.

Para que el algoritmo funcione adecuadamente es necesario un mínimo de 10 capturas del tablero en diferentes posiciones, habiéndose utilizado 30 en el presente caso. En la figura 15 se muestra parte del proceso de calibración.



(a) Imagen cámara 1. (b) Imagen cámara 2. (c) Imagen cámara 3.

Figura 15: Imágenes capturadas por las cámaras.

II-C5. Calibración extrínseca de las cámaras: La *calibración extrínseca* hace referencia a los vectores de *traslación* y *rotación* de las cámaras respecto a las demás, haciendo que el sistema encuentre y una las imágenes capturadas para generar una imagen compuesta [26].

El cálculo de estos vectores se lo realiza de la siguiente manera:

- Se realiza un reconocimiento de *puntos clave* (también denominados como *puntos sobresalientes* o *puntos únicos*) y sus *descriptores* (o características) en las imágenes. Esto se lo puede realizar haciendo uso de las funciones: *SURF*, *SIFT* y *ORB* [26].
- A continuación, el algoritmo, calcula las mejores opciones para la coincidencia entre los puntos anteriores. A este proceso se lo conoce como *stitching* [26].
- Por último se calcula la matriz de *deformación* y *traslación* requerida por cada imagen para que los puntos coincidentes se superpongan. A esta matriz se la conoce como *matriz de homografía* [26].

Cabe puntualizar que las funciones del primer punto brindan mejores resultados si las imágenes son ricas en contrastes por los objetos presentes. [26]

En el presente proyecto se utiliza la función *SIFT* entre otras de la librería *OpenCV* para calcular las *matrices de homografía* entre cámaras. El algoritmo implementado se muestra en la figura 16 y el procesamiento en las figuras 21, 22, 23 y 24 del apéndice G.

II-C6. Desarrollo del sistema de posicionamiento:

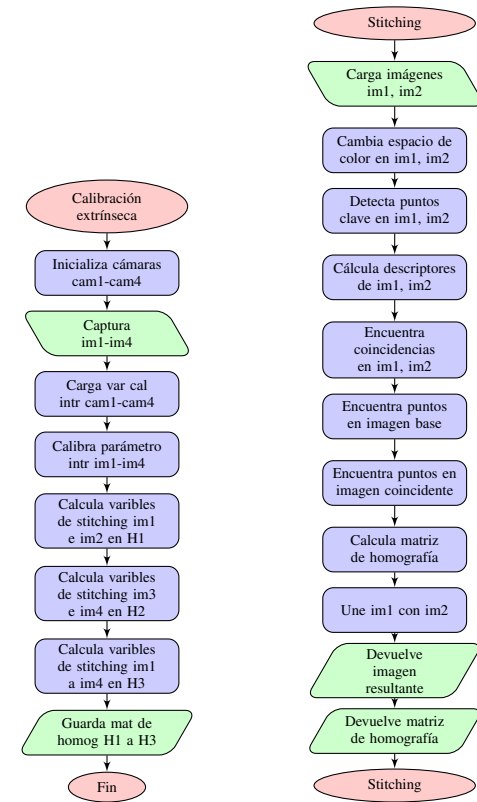
Se define el *área de trabajo* como el área cubierta por los *campos de visión* de las cámaras y la *superficie analizada* a la superficie ensamblada con las *matrices de homografía* habiéndose corregido previamente las *distorsiones intrínsecas* provenientes de cada cámara.

Por consiguiente el *sistema de posicionamiento* es el programa que calcula la *posición y orientación* de objetos dentro del *área de trabajo* con la ayuda de *marcadores*, utilizando para ello etapas de filtrado sobre la *superficie analizada*.

Los *marcadores* utilizados tienen una geometría de *triángulo isósceles* (se muestran en la figura 17), definiendo su *centroide o baricentro* como su posición y el ángulo comprendido entre la *mediana de mayor longitud* y el eje *y* como su orientación. Estos elementos se detallan en la figura 18.

Para que el sistema cumpla con su objetivo eficientemente, optimizando el tiempo de *procesamiento por iteración*, el programa cuenta con dos características clave:

- Los parámetros de calibración *intrínseca* y *extrínseca* son calculados previamente, eliminando la complejidad que implica para el usuario así como el tiempo de



(a) Diagrama de flujo de stitching.

(b) Diagrama de flujo de calibración extrínseca.

Figura 16: Diagrama de flujo: calibración extrínseca.

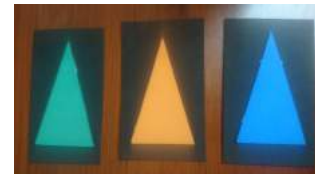


Figura 17: Marcadores utilizados.

procesamiento requerido.

- La captura, decodificación y corrección de los *streamings* provenientes de las cámaras resultan ser tareas intensivas en el tiempo y recursos de procesamiento. Así para evitar el *bloqueo* del programa se ha asignado *hilos de proceso independientes* a cada cámara.

El algoritmo del programa desarrollado se muestra en la figura 19.

III. CARACTERIZACIÓN DEL SISTEMA

Una vez completadas las fases de desarrollo e implementación de un dispositivo, este debe ser caracterizado

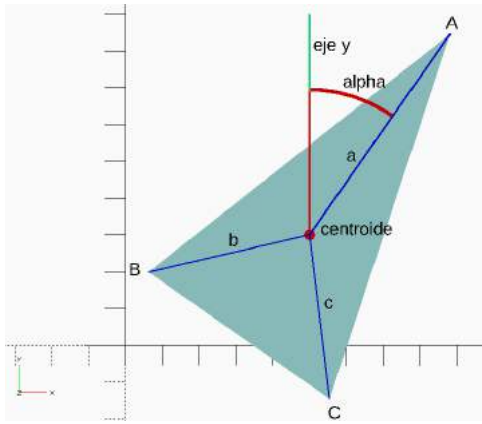
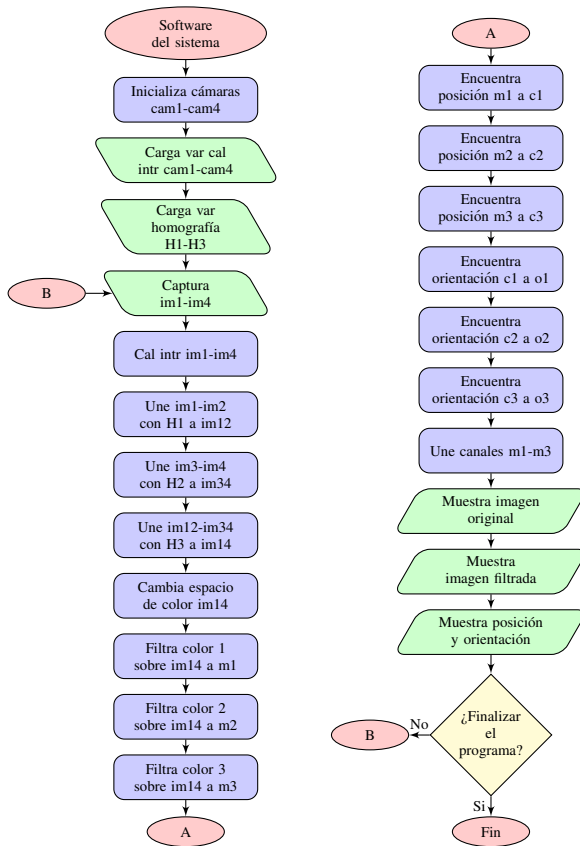


Figura 18: Elementos calculados de los marcadores. En mayúsculas los *vértices*, azul las *medianas*, verde el eje *y*, el *centroide* en la unión de las medianas y *alpha* la orientación respecto al eje *y*.



(a) Diagrama de flujo: software del sistema parte 1. (b) Diagrama de flujo: software del sistema parte 2.

Figura 19: Diagrama de flujo: sistema de posicionamiento.

mediante un conjunto de pruebas destinadas a medir y calcular los parámetros de entrada y salida del mismo.

El presente proyecto, al tratarse de un sistema de posicionamiento óptico, se somete a pruebas dedicadas a

encontrar la uniformidad, precisión, exactitud y resolución al momento de ubicar y orientar objetos que se encuentren dentro del *área de trabajo*. Las pruebas realizadas se detallan a continuación:

- Toma de 1000 muestras de dos marcadores colocados a una distancia conocida paralelo al eje horizontal, habiéndose repetido el proceso en cada cuadrante del *área de trabajo*.
- Toma de 1000 muestras de dos marcadores colocados a una distancia conocida paralelo al eje vertical, habiéndose repetido el proceso en cada cuadrante del *área de trabajo*.
- Toma de 1000 muestras de un marcador colocado a una orientación conocida respecto del eje vertical del sistema, repitiéndose el proceso cuatro veces por cuadrante en cada los cuadrante del *área de trabajo*.
- Toma del tiempo por muestra de los datos obtenidos anteriormente.

Estos datos se analizaron de forma estadística, utilizándose para ello las siguientes funciones: [27]

- *Media aritmética*: calcula el promedio del valor medido en el conjunto de muestras.
- *Mediana*: calcula el valor medio entre el valor máximo y mínimo medido en el conjunto de muestras.
- *Moda*: indica el valor que más se repite dentro del conjunto de muestras.
- *Desviación estándar*: calcula el grado de dispersión de las muestras.

Los cálculos obtenidos de las pruebas nombradas se muestran en la tabla IX y los gráficos de dispersión se encuentran en el apéndice H.

De la tabla nombrada se infiere que:

- El sistema tiene un alto grado de *precisión* debido a la proximidad de los valores de la *media*, *mediana* y *moda*, así como al bajo nivel de dispersión dado por la *desviación estándar*.
- Consta de un alto índice de *uniformidad* dada la poca diferencia entre las magnitudes de la *moda* en los diferentes *cuadrantes* de la *superficie analizada*.
- La *exactitud* de las medidas angulares es elevada debido a la cercanía de los valores calculados en la

Tabla IX: Resumen de mediciones

		Cuadrantes				Prom
		1	2	3	4	
Estadísticas longitudinales (<i>píxels</i>)						
Hor	Media	327.14	318.70	321.74	317.54	321.28
	Mediana	327.14	318.67	321.74	317.50	321.26
	Moda	327.00	318.74	321.71	315.80	320.813
	Desv est	0.10	0.24	0.10	2.11	0.64
Vert	Media	315.08	319.04	316.18	324.01	318.58
	Mediana	313.57	319.03	316.18	323.93	318.18
	Moda	313.16	318.74	316.06	323.46	317.86
	Desv est	5.59	0.13	0.17	1.33	1.805
Estadísticas angulares (<i>grados</i>)						
$\theta 0$	Media	14.03	7.52	3.09	4.21	7.21
	Mediana	15.32	8.14	0	5.87	7.33
	Moda	0	0	0	5.60	1.4
	Desv est	13.75	5.10	8.71	8.70	9.07
$\theta 90$	Media	85.82	91.81	96.81	103.25	94.42
	Mediana	90	90	100.00	106.76	386.76
	Moda	90	90	90	90	90
	Desv est	11.73	3.57	8.46	6.75	7.63
$\theta 180$	Media	162.56	161.55	168.81	166.58	164.88
	Mediana	159.87	161.82	180	167.09	167.20
	Moda	180	162.59	180	180	175.65
	Desv est	11.35	2.41	11.76	5.52	7.76
$\theta -90$	Media	-89.94	-82.84	-99.41	-93.75	91.49
	Mediana	-90	-80.31	-102.00	-90	90.58
	Moda	-90	-90	-90	-90	-90
	Desv est	12.15	4.29	8.10	5.18	7.43
Estadísticas temporales por iteración (<i>ms</i>)						
δt	Media	19.485				
	Mediana	18.00				
	Moda	17				
	Desv est	2.937				

superficie analizada respecto de la orientación en el *área de trabajo*.

- Dado el intervalo de tiempo por muestra del prototipo, se calcula mediante la ecuación 13 que el sistema puede procesar un total de 58,82 muestras por segundo. Esto implica que el sistema utiliza de manera efectiva todas las capturas obtenidas por parte de los periféricos según el teorema de Nyquist. [28]
- La resolución del sistema, calculada con la ecuación 14, es de aproximadamente 319 *píxels/metro*, lo que da una sensibilidad de aproximadamente 3 *mm/píxel*.

$$freq_{muestras} = 1/t_{muestra} \quad (13)$$

$$resolución = píxels/distancia \quad (14)$$

IV. ANÁLISIS DE DATOS OBTENIDOS

Dentro del análisis de resultados se busca la comparación de las características de nuestro sistema con las de sistemas similares con el objetivo de validar el prototipo.

Así, el proyecto [29] que realiza un análisis sistemático de la precisión y exactitud de un sistema de captura de movimiento por vídeo, ejecutando pruebas sobre un accionador electromecánico para el labrado de piezas donde obtuvo una desviación estándar de la exactitud inferior a 35 y una desviación estándar de la precisión de 15 en medidas longitudinales.

Además el proyecto [30] mide y analiza del movimiento angular del tobillo en el análisis de la marcha usando Kinovea obteniendo una desviación estándar entre 6,56 y 1,28 dependiendo de la fase de la marcha medida.

Cabe mencionar el proyecto [31] analiza el movimiento biomecánico utilizando Kinovea aplicado a actividades deportivas donde se comprobaron los ángulos dentro de pruebas de salto a nivel inferior utilizando dos tecnologías (Kinovea y Cortex) obteniendo una desviación media entre 6,56 y 1,28 siendo el porcentaje de diferencia entre las dos entre 0,21 % y el 37,97 %.

Los resultados obtenidos con nuestro prototipo en contraste tienen una desviación media entre 5,59 y 0,13 en medidas longitudinales, y entre 13,75 y 2,41 en medidas angulares, teniendo adicionalmente una moda muy próxima a la media.

La comparación de estos resultados valida el prototipo desarrollado en el presente proyecto al presentar una dispersión de datos dentro de los márgenes de otras herramientas así como presentar una repetitividad en el rango de estas tecnologías.

Como resultado en la imagen 20 se puede apreciar los *marcadores* dentro de la *superficie analizada*, producto de los algoritmos expuestos en este documento, aplicados sobre las capturas del *área de trabajo*.

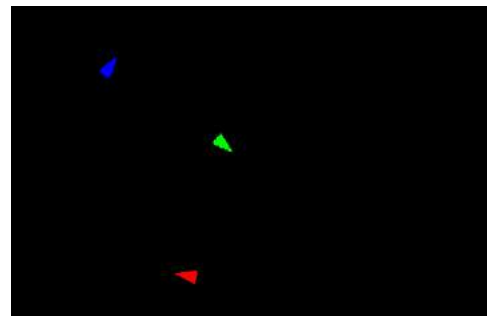


Figura 20: Resultado final (*Superficie analizada*).

V. CONCLUSIONES

En resumen, el presente documento fue orientado al desarrollo e implementación de un prototipo capaz de ubicar objetos y su orientación en un *área de trabajo*, cuyos resultados dentro del análisis de uniformidad, resolución, precisión y exactitud se encuentran dentro de márgenes de funcionamiento generales de otros dispositivos similares.

La minuciosidad de las pruebas realizadas sobre los periféricos así como a la unidad de procesamiento central, donde se obtuvieron varios parámetros de funcionamiento así como características y limitaciones del sistema, resulta de gran importancia al definir los alcances de la configuración del sistema así como las pautas para el incremento de características del mismo.

El presente prototipo fue concebido para ser una herramienta que pueda ser utilizada en multitud de aplicaciones, por lo que se recomienda se complementen las pruebas aquí realizadas con pruebas afines a la aplicación con el fin de asegurar la confiabilidad de las mediciones efectuadas por el sistema.

APÉNDICE A RESULTADOS COMANDO *lshw*

Log

```
fxr-uda
...
*-core
...
*-pci
...
*-pci:0
...
*-usb
...
*-communication
...
*-storage
...
*-pci:1
...
*-pci:2
  descripción: PCI bridge
  producto: Sunrise Point-H PCI Express Root Port #1
  fabricante: Intel Corporation
  id físico: 1c
  información del bus: pci@0000:00:1c.0
  versión: f1
  anchura: 32 bits
  reloj: 33MHz
  capacidades: pci pciexpress msi pm normal_decode bus_
master cap_list
  configuración: driver=pcieport
  recursos: irq:16 memoria:f7100000-f71fffff
*-usb
  descripción: USB controller
  producto: ASM1142 USB 3.1 Host Controller
  fabricante: ASMedia Technology Inc.
  id físico: 0
  información del bus: pci@0000:03:00.0
  versión: 00
  anchura: 64 bits
  reloj: 33MHz
  capacidades: msi msix pm pciexpress xhci bus_master
cap_list
  configuración: driver=xhci_hcd latency=0
  recursos: irq:16 memoria:f7100000-f7107fff
*-usbhost:0
  producto: xHCI Host Controller
  fabricante: Linux 4.15.0-36-lowlatency xhci-hcd
  id físico: 0
  información del bus: usb@3
  nombre lógico: usb3
  versión: 4.15
  capacidades: usb-2.00
  configuración: driver=hub slots=2 speed=480Mbit/s
*-usb
  descripción: USB hub
  producto: USB2.0 Hub
  fabricante: VIA Labs, Inc.
  id físico: 1
  información del bus: usb@3:1
  versión: 90.01
  capacidades: usb-2.10
  configuración: driver=hub slots=4 speed=480Mbit/
s
*-usb
  descripción: USB hub
  producto: USB2.0 Hub
  fabricante: VIA Labs, Inc.
  id físico: 1
  información del bus: usb@3:1.1
  versión: 90.01
  capacidades: usb-2.10
  configuración: driver=hub slots=4 speed=480Mbi
t/s
*-usb
  descripción: USB hub
  producto: 4-Port USB 2.0 Hub
  fabricante: Generic
  id físico: 1
  información del bus: usb@3:1.1.1
```

```

versión: 1.21
capacidades: usb-2.10
configuración: driver=hub slots=4 speed=480Mb
it/s
*-usb:0
  descripción: Dispositivo de sonido
  producto: USB Camera-B4.09.24.1
  fabricante: OmniVision Technologies, Inc.
  id físico: 1
  información del bus: usb@3:1.1.1.1
  versión: 2.00
  capacidades: usb-2.00 audio-control
  configuración: driver=snd-usb-audio maxpowe
r=500mA speed=480Mbit/s
*-usb:1
  descripción: Dispositivo de sonido
  producto: USB Camera-B4.09.24.1
  fabricante: OmniVision Technologies, Inc.
  id físico: 2
  información del bus: usb@3:1.1.1.2
  versión: 2.00
  capacidades: usb-2.00 audio-control
  configuración: driver=snd-usb-audio maxpowe
r=500mA speed=480Mbit/s
*-usb:2
  descripción: Dispositivo de sonido
  producto: USB Camera-B4.09.24.1
  fabricante: OmniVision Technologies, Inc.
  id físico: 3
  información del bus: usb@3:1.1.1.3
  versión: 2.00
  capacidades: usb-2.00 audio-control
  configuración: driver=snd-usb-audio maxpowe
r=500mA speed=480Mbit/s
*-usb:3
  descripción: Dispositivo de sonido
  producto: USB Camera-B4.09.24.1
  fabricante: OmniVision Technologies, Inc.
  id físico: 4
  información del bus: usb@3:1.1.1.4
  versión: 2.00
  capacidades: usb-2.00 audio-control
  configuración: driver=snd-usb-audio maxpowe
r=500mA speed=480Mbit/s
*-usbhost:1
...
*-pci:3
...
*-isa
...
*-memory NO RECLAMADO
...
*-multimedia
...
*-serial NO RECLAMADO
...
*-network
...

```

APÉNDICE B RESULTADOS COMANDO *lsusb*

Log

```

Bus 004 Device 004: ID 0bda:0411 Realtek Semiconductor Corp.
Bus 004 Device 003: ID 2109:0813 VIA Labs, Inc.
Bus 004 Device 002: ID 2109:0813 VIA Labs, Inc.
Bus 004 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root h
ub
Bus 003 Device 008: ID 1415:2000 Nam Tai E&E Products Ltd. o
r OmniVision Technologies, Inc. Sony Playstation Eye
Bus 003 Device 007: ID 1415:2000 Nam Tai E&E Products Ltd. o
r OmniVision Technologies, Inc. Sony Playstation Eye
Bus 003 Device 006: ID 1415:2000 Nam Tai E&E Products Ltd. o
r OmniVision Technologies, Inc. Sony Playstation Eye
Bus 003 Device 005: ID 1415:2000 Nam Tai E&E Products Ltd. o
r OmniVision Technologies, Inc. Sony Playstation Eye
Bus 003 Device 004: ID 0bda:5411 Realtek Semiconductor Corp.
Bus 003 Device 003: ID 2109:2813 VIA Labs, Inc.
Bus 003 Device 002: ID 2109:2813 VIA Labs, Inc.
Bus 003 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root h
ub
Bus 002 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root h
ub
Bus 001 Device 003: ID 258a:0001
Bus 001 Device 002: ID 0458:0186 KYE Systems Corp. Mouse Systems
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root h
ub

```

APÉNDICE C RESULTADOS COMANDO *v4l2 - ctl*

Log

```

ioctl: VIDIOC_ENUM_FMT
Index      : 0
Type       : Video Capture
Pixel Format: 'YUYV'
Name       : YUYV 4:2:2
Size: Discrete 320x240
Interval: Discrete 0.005s 187.000 fps
Interval: Discrete 0.007s 150.000 fps
Interval: Discrete 0.007s 137.000 fps
Interval: Discrete 0.008s 125.000 fps
Interval: Discrete 0.010s 100.000 fps
Interval: Discrete 0.013s 75.000 fps
Interval: Discrete 0.017s 60.000 fps
Interval: Discrete 0.020s 50.000 fps
Interval: Discrete 0.027s 37.000 fps
Interval: Discrete 0.033s 30.000 fps
Size: Discrete 640x480
Interval: Discrete 0.017s 60.000 fps
Interval: Discrete 0.020s 50.000 fps
Interval: Discrete 0.025s 40.000 fps
Interval: Discrete 0.033s 30.000 fps
Interval: Discrete 0.067s 15.000 fps

```


APÉNDICE D
RESULTADOS HERRAMIENTA *usbmon*

Para el uso correcto de esta herramienta se siguen los siguientes pasos:

1. Se activa su uso en el sistema.
2. Se verifica que los *bus sockets* estén presentes.
3. Se leen los archivos correspondientes del socket, donde la información de tiempo y tamaño de trama se procede a calcular el ancho de banda utilizado. Adicionalmente se puede saber que tipo de transferencia se está utilizando.

El formato de salida se debe leer como se indica en la tabla X.

Tabla X: Campos de salida del socket

Campo	Descripción
URB Tag	Se utiliza para identificar las direcciones otorgadas por el kernel en hexadecimal.
Timestamp	El momento en que inició el evento en microsegundos.
Event type	Hace referencia al formato del evento. <i>S</i> sumisión, <i>C</i> llamado, <i>E</i> error de sumisión.
Address	Consiste de 4 parámetros que son: tipo y dirección del <i>URB</i> , número de <i>Bus</i> , dirección del dispositivo, número de <i>endpoint</i> .
URB status	Puede ser una letra o varios números separados por puntos. Indican varios parámetros de la comunicación.
Data length	Para sumisiones, este parámetro es el tamaño de la trama solicitado, para llamados es el tamaño de trama otorgado.
Data tag	Si existe el símbolo = significa que <i>usbmon</i> logró capturar la trama.
Data words	Es la información que se transmitió en esa trama.

En los *logs* se muestran los primeros 10 resultados adquiridos.

D-A. Bus sockets

0s~0u~1s~1t~1u~2s~2t~2u~3s~3t~3u~4s~4t~4u

D-B. Log 15FPS

Log

```
ffff8b79a513c840 2037201013 C Bi:3:007:1 0 16384 = 0c0cfb34
4e0e44f8 5d0e0501 6351708a 60544f8a 5a52688c 6e4b748d 774971
8f
ffff8b79a513c840 2037201023 S Bi:3:007:1 -115 16384 <
ffff8b79a513ca80 2037202678 C Bi:3:007:1 0 16384 = 0c0cfb34
4e0edbbc 5e0e0601 7b457f92 7f457f92 7f4a7592 77517b8e 7c579e
84
ffff8b79a513ca80 2037202684 S Bi:3:007:1 -115 16384 <
ffff8b79a513c840 2037204389 C Bi:3:007:1 0 16384 = 0c0cfb34
4e0e677e 5f0e0801 4b58608e 664f618f 6d46748f 73446c90 6d496e
```

D-C. Log 30FPS

Log

```
ffff8b79a53fdb40 2002904880 C Bi:3:007:1 0 16384 = 0c0d77f4
641aa9ec 721a1003 824a8a90 844c8b8d 874d7d8e 7f4a818f 854882
8f
ffff8b79a53fdb40 2002904889 S Bi:3:007:1 -115 16384 <
```

```
ffff8b79a53fd240 2002905737 C Bi:3:007:1 0 16384 = 0c0d77f4
641ac84d 731a1003 3f5d4b8d 1c5d258d 575d408d 4a55528d 405255
8c
ffff8b79a53fd240 2002905742 S Bi:3:007:1 -115 16384 <
ffff8b79a53fdb40 2002906300 C Bi:3:007:1 0 11816 = 0c0d77f4
641a40b1 731a1103 8c467791 5b4c6492 724d7291 73497592 704867
```

D-D. Log 40FPS

Log

```
ffff8b7b58b27c00 1973668971 C Bi:3:007:1 0 16384 = 0c0d15d8
c30b6534 c50bde00 b95ec17a ca5ec878 c35dbc79 bf59b97b aa5a94
80
ffff8b7b58b27c00 1973668988 S Bi:3:007:1 -115 16384 <
ffff8b7b58b273c0 1973669622 C Bi:3:007:1 0 16384 = 0c0d15d8
c30ba97d c50bdf00 7a3d7d93 773a7594 753f7992 75496c91 614b60
91
ffff8b7b58b273c0 1973669633 S Bi:3:007:1 -115 16384 <
ffff8b7b58b27c00 1973670246 C Bi:3:007:1 0 16384 = 0c0d15d8
c30b0bc5 c50bdf00 934e968c 9b4e908c 904e938c 8f4d8a8c 7d4d84
```

D-E. Log 50FPS

Log

```
ffff8b7af5b3c240 1927515874 C Bi:3:007:1 0 16384 = 0c0d9afe
f90b3c80 fe0b9804 37633186 2c622988 3866268a 1566288a 396243
8a
ffff8b7af5b3c240 1927515896 S Bi:3:007:1 -115 16384 <
ffff8b7af5b3c480 1927516387 C Bi:3:007:1 0 16384 = 0c0d9afe
f90bd5ba fe0b9904 655f648c 5861688a 555d4b8c 735b7e8d 7e5779
8d
ffff8b7af5b3c480 1927516401 S Bi:3:007:1 -115 16384 <
ffff8b7af5b3c240 1927516869 C Bi:3:007:1 0 16384 = 0c0d9afe
f90b6ef5 fe0b9904 68487994 7f468092 924d8290 7c57848c 8f5fb2
```

D-F. Log 60FPS

Log

```
ffff8b79a51506c0 1882151268 C Bi:3:007:1 0 16384 = 0c0d463a
d928da8e d9286703 866f9b7e af68b77c b865b97b ba65bb7a bb65bc
7a
ffff8b79a51506c0 1882151277 S Bi:3:007:1 -115 16384 <
ffff8b79a5150900 1882151673 C Bi:3:007:1 0 16384 = 0c0d463a
d92858bf d9286703 89598788 8d54928a 8d52858c 80547f8d 855b84
8a
ffff8b79a5150900 1882151680 S Bi:3:007:1 -115 16384 <
ffff8b79a51506c0 1882152091 C Bi:3:007:1 0 16384 = 0c0d463a
d92840ef d9286803 c46bc37d c569c67d c668c67d c46ac57d c76dc4
```

APÉNDICE E
REGLAS *udev* IMPLEMENTADAS

Log

```
SUBSYSTEM=="video4linux", KERNELS=="1-6:1.0", SYMLINK+="vidc
am"
SUBSYSTEM=="video4linux", KERNELS=="3-1.1.1.3:1.0", SYMLINK+
="vidFondIzq"
SUBSYSTEM=="video4linux", KERNELS=="3-1.1.1.4:1.0", SYMLINK+
="vidFondDer"
SUBSYSTEM=="video4linux", KERNELS=="3-1.1.1.1:1.0", SYMLINK+
="vidFrenIzq"
SUBSYSTEM=="video4linux", KERNELS=="3-1.1.1.2:1.0", SYMLINK+
="vidFrenDer"
```

APÉNDICE F
INICIALIZACIÓN DE LAS CÁMARAS

F-A. Carga de parámetros

Log

```
#!/bin/bash

if [[ -e /dev/vidFondIzq && -e /dev/vidFondDer && -e /dev/vidFrenIzq && -e /dev/vidFrenDer ]]; then
    v4l2-ctl -d /dev/vidFondIzq --set-ctrl=white_balance_automatic=0,gain_automatic=0,auto_exposure=1
    v4l2-ctl -d /dev/vidFondDer --set-ctrl=white_balance_automatic=0,gain_automatic=0,auto_exposure=1
    v4l2-ctl -d /dev/vidFrenIzq --set-ctrl=white_balance_automatic=0,gain_automatic=0,auto_exposure=1
    v4l2-ctl -d /dev/vidFrenDer --set-ctrl=white_balance_automatic=0,gain_automatic=0,auto_exposure=1

    v4l2-ctl -d /dev/vidFondIzq --set-ctrl=brightness=0,contrast=40,saturation=70,hue=0,exposure=30,gain=0,sharpness=10
    v4l2-ctl -d /dev/vidFondDer --set-ctrl=brightness=0,contrast=40,saturation=70,hue=0,exposure=30,gain=0,sharpness=10
    v4l2-ctl -d /dev/vidFrenIzq --set-ctrl=brightness=0,contrast=40,saturation=70,hue=0,exposure=30,gain=0,sharpness=10
    v4l2-ctl -d /dev/vidFrenDer --set-ctrl=brightness=0,contrast=40,saturation=70,hue=0,exposure=30,gain=0,sharpness=10

    echo "$date se cargaron los parámetros en todas las cámaras"
else
    echo "$date una o mas cámaras no están conectadas"
fi
```

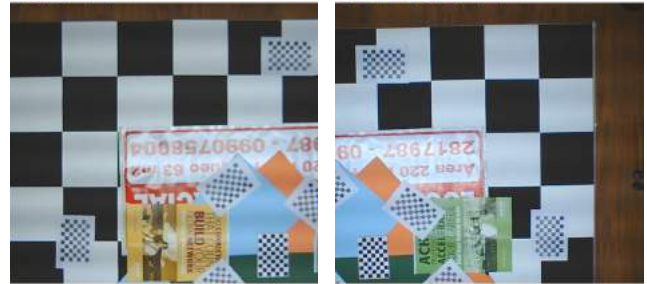
F-B. Ejecución al encendido del sistema

Log

```
@reboot sleep 10 && /home/fxr/.camsParams >> .camsParamsLog
```

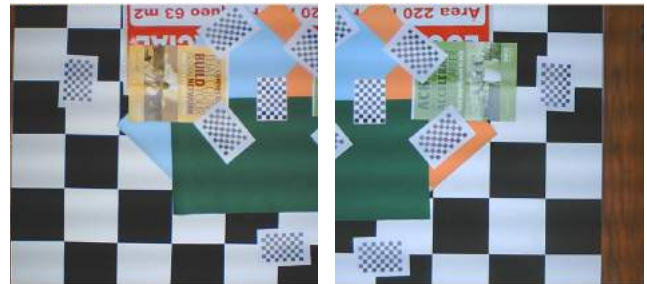
APÉNDICE G

PROCESO DE ENSAMBLE DE IMÁGENES CALCULANDO LAS MATRICES DE HOMOGRAFÍAS



(a) Imagen cámara 1.

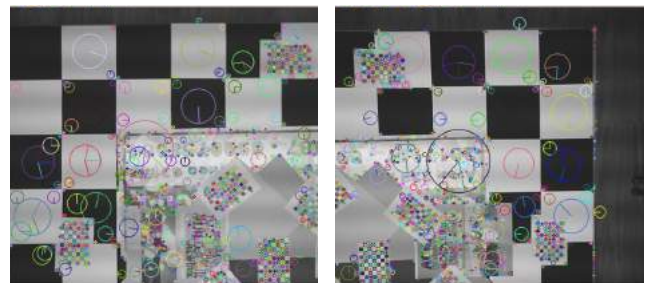
(b) Imagen cámara 2.



(c) Imagen cámara 3.

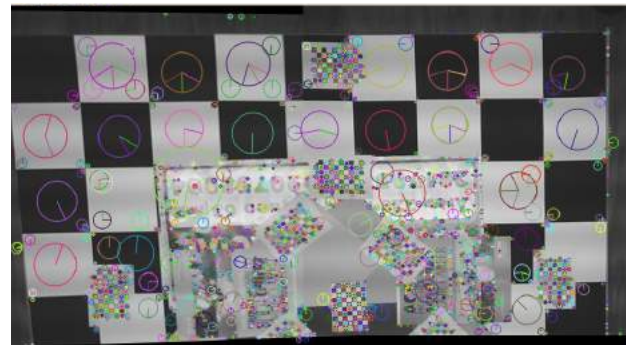
(d) Imagen cámara 4.

Figura 21: Imágenes capturadas por las cámaras.



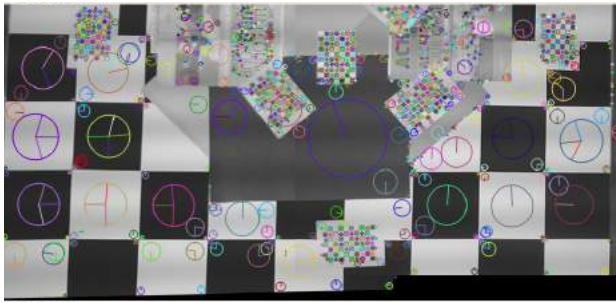
(a) Resultado SIFT cámara 1.

(b) Resultado SIFT cámara 2.

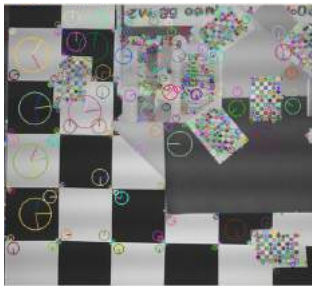


(c) Resultado SIFT cámaras 1 y 2.

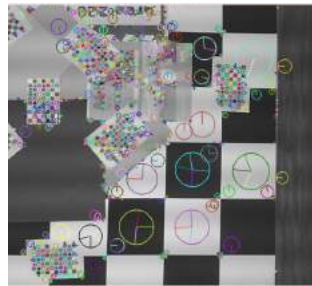
Figura 22: Resultados de la función *SIFT* y proceso de ensamblado de la imagen superior.



(a) Resultado SIFT cámaras 3 y 4.

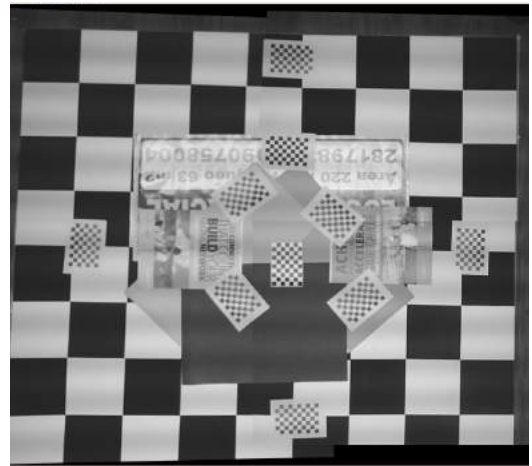


(b) Resultado SIFT cámara 3.



(c) Resultado SIFT cámara 4.

Figura 23: Resultados de la función *SIFT* y proceso de ensamblado de la imagen inferior.



(a) Imagen ensamblada.



(b) Imagen final.

Figura 24: Resultados finales.

APÉNDICE H
GRÁFICOS DE DISPERSIÓN

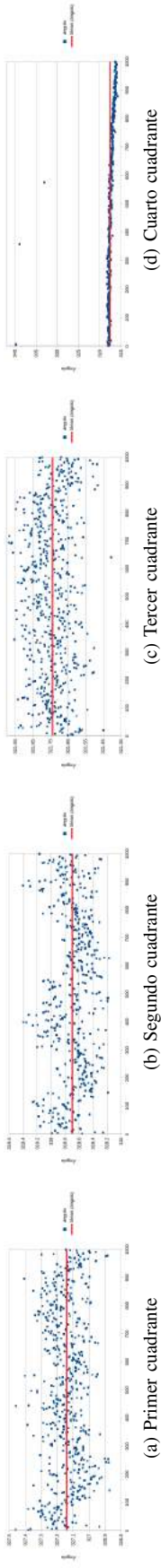


Figura 25: Muestras longitudinales horizontales

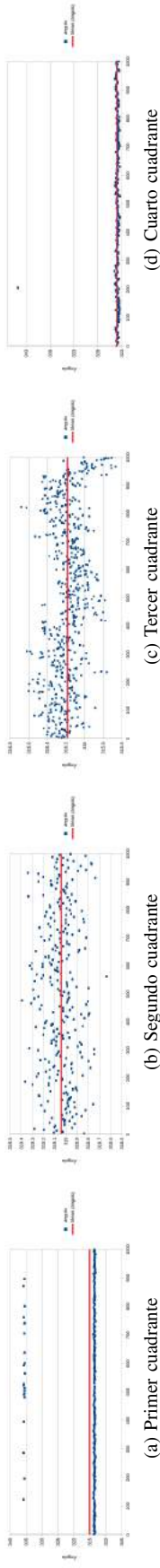


Figura 26: Muestras longitudinales verticales

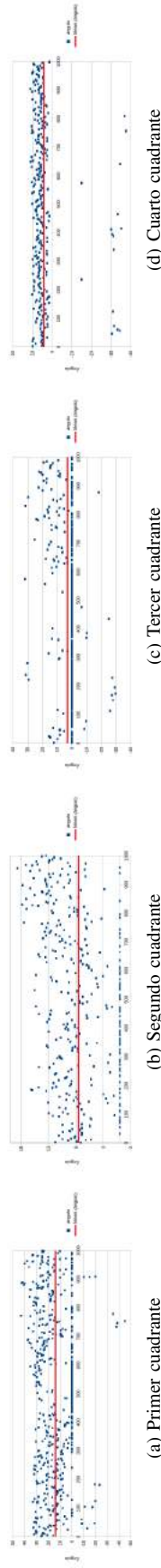


Figura 27: Muestras angulares 0°

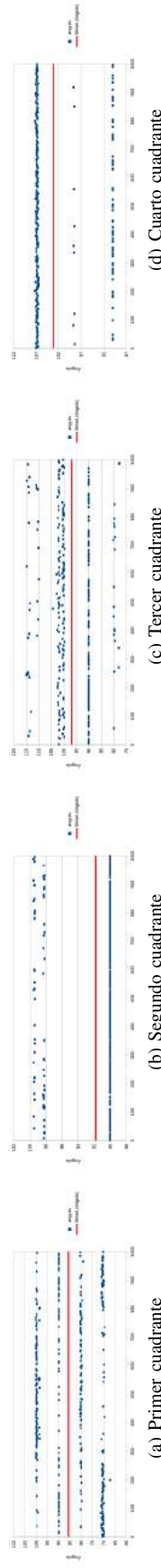


Figura 28: Muestras angulares 90°

REFERENCIAS

- [1] H. Kopka and P. W. Daly, *A Guide to LATEX*, 3rd ed., A. Wesley, Ed., 1999.
- [2] S. Goswami, *Indoor Location Technologies*. Springer, 2013.
- [3] S. Riisgaard and M. R. Blas, *SLAM for Dummies A Tutorial Approach to Simultaneous Localization and Mapping*.
- [4] J. Rowekamper, C. Sprunk, G. D. Tipaldi, C. Stachniss, P. Pfaff, and W. Burgard, "On the position accuracy of mobile robot localization based on particle filters combined with scan matching," *International Conference on Intelligent Robots and Systems*, Agosto 2012.
- [5] J. Imtiaz, N. Koch, H. Flatt, J. Jasperneite, M. Voit, and F. van de Camp, "A flexible context-aware assistance system for industrial application using camera based localization," *IEEE Emerging Technology and Factory Automation (ETFA)*, 2014.
- [6] DavideBaccio, C. Gallicchio, A. Micheli, M. D. Rocco, and A. Saffiotti, "Learning context-aware mobile robot navigation in home environments," *IISA 2014*, 2014.
- [7] M. Faisal, R. Hedjar, M. A. Sulaiman, and K. Al-Mutib, "Fuzzy logic navigation and obstacle avoidance by a mobile robot in an unknown dynamic environment," *International Journal of Advanced Robotic Systems*, vol. 10, no. 37, pp. 1–7, 2013.
- [8] L. Havasi and Z. Szlavik, "A statistical method for object localization in multi-camera tracking," *Proceedings of 2010*, 2010.
- [9] J. Clark and B. Keltner. (2007) Playstation® eye brings next-generation communication to playstation®3. [Online]. Available: https://www.sony.com/en_us/SCA/company-news/press-releases/sony-computer-entertainment-america-inc/2007/playstation-eye-brings-nextgeneration-communication-to-playstation3.html
- [10] G. Sullivan and S. Estrop. (2008) Recommended 8:bit yuv formats for video rendering. [Online]. Available: <https://docs.microsoft.com/en-us/windows/desktop/medfound/recommended-8-bit-yuv-formats-for-video-rendering>
- [11] U. . P. Group, *Architectural Overview*, 2017.
- [12] J. A. Reyes-Labarta and A. Marcilla, "Thermal treatment and degradation of cross-linked ethylene vinyl acetate–polyethylene–azodicarbonamide–zno foams. complete kinetic modeling and analysis," *Industrial & Engineering Chemistry Research*, vol. 51, no. 28, pp. 9515–9530, 2012. [Online]. Available: <https://doi.org/10.1021/ie3006935>
- [13] Windows. (2018) Windows kernel. [Online]. Available: https://docs.microsoft.com/en-us/windows-hardware/drivers/ddi/content/_kernel/
- [14] Ckimes. (2017) Kernel. [Online]. Available: <https://help.ubuntu.com/community/Kernel>
- [15] Ubuntu. (2018) Ubuntu desktop. [Online]. Available: <https://www.ubuntu.com/download/desktop>
- [16] Debian. ¿qué es gnu/linux? [Online]. Available: <https://www.debian.org/releases/stable/arm64/ch01s02.html.es>
- [17] Canonical. (2018) Canonical. [Online]. Available: <https://www.canonical.com>
- [18] Maechler. (2015) Ubuntustudio/realtimkernel. [Online]. Available: <https://help.ubuntu.com/community/UbuntuStudio/RealTimeKernel>
- [19] L. Vincent. (2010) lshw - list hardware. [Online]. Available: <http://manpages.ubuntu.com/manpages/bionic/man1/lshw.1.html>
- [20] T. Sailer. (2016) lsusb - list usb devices. [Online]. Available: <http://manpages.ubuntu.com/manpages/bionic/man8/lsusb.8.html>
- [21] I. P. Gonzalez. (2017) Usbmon documentation. [Online]. Available: <https://www.kernel.org/doc/Documentation/usb/usbmon.txt>
- [22] H. Verkuil. (2018) v4l2-ctl - utils. [Online]. Available: <https://git.linuxtv.org/v4l-utils.git/about/>
- [23] Canonical. (2016) udev - dynamic device management. [Online]. Available: <http://manpages.ubuntu.com/manpages/xenial/man7/udev.7.html>
- [24] G. Bradski and A. Kaehler, *Learning OpenCV*, 1st ed., M. Koukides, Ed. O'Reilly Media, Inc, Septiembre 2008.
- [25] OpenCV. (2018) Camera calibration. Open Source Computer Vision. [Online]. Available: https://docs.opencv.org/3.4.3/dc/d43/tutorial_camera_calibration_square_chess.html
- [26] S. Ghosh. (2016, Septiembre) Camera calibration using c++ and opencv. [Online]. Available: <http://sourishghosh.com/2016/camera-calibration-cpp-opencv/>
- [27] G. C. Canavos, *Conceptos de probabilidad*, 1998.
- [28] M. Semeria, "Los tres teoremas. fourier - nyquist - shannon," *Serie Documentos de Trabajo*, 2015.
- [29] M. Windolf, N. Gotzen, and M. Morlock, "Systematic accuracy and precision analysis of video motion capturing systems - exemplified on the vicon-460 system," *Journal of Biomechanics*, 2008.
- [30] N. A. H. Hishman, A. F. A. Nazri, J. Madete, and L. Herawati, "Measuring ankle angle and analysis of walking gait using kinovea," in *International Medical Device and Technology Conference*, 2017.
- [31] N. M. N. Adnan, M. N. A. A. Patar, H. Lee, S.-I. Yamamoto, L. Jong-Young, and J. Mahmud, "Biomechanical analysis using kinovea for sports application," in *Materials Science and Engineering*, 2018.