

**DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE VISIÓN ESTEREOSCÓPICA PARA  
APLICACIONES ROBÓTICAS**

**DAVID ARMANDO REVELO LUNA  
FRANCISCO DANIEL USAMA OVIEDO**

**Trabajo de Grado presentado como requisito parcial para optar al título de  
Ingeniero Físico en la modalidad de trabajo de investigación**

**Director**

**Ing. JUAN FERNANDO FLOREZ MARULANDA**

**Universidad del Cauca  
Facultad de Ciencias Naturales, Exactas y de la Educación**

**Programa de Ingeniería Física**

**Popayán – Cauca**

**2010**

**DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE VISIÓN ESTEREOSCÓPICA PARA  
APLICACIONES ROBÓTICAS**

**DAVID ARMANDO REVELO LUNA  
FRANCISCO DANIEL USAMA OVIEDO**

**Universidad del Cauca  
Facultad de Ciencias Naturales, Exactas y de la Educación  
Programa de Ingeniería Física  
Popayán – Cauca  
2010**

**Nota de aceptación**

---

---

---

---

---

**Ing. JUAN FERNANDO FLOREZ MARULANDA**

Director del proyecto de investigación

---

**Ing. LEONAIRO PENCUE FIERRO**

Evaluador del Proyecto

---

**Mg. LUIS FERNANDO ECHEVERRY**

Evaluadora del Proyecto

Fecha de sustentación: 28 de junio 2010

## **AGRADECIMIENTOS**

Agradezco a mis padres Germán Armando Revelo y Lucy Luna por su apoyo incondicional, porque mi vida gira en torno a sus enseñanzas y por hacerme sentir orgulloso de ellos todos los días de mi vida. A mi hermano porque con él he vivido los momentos más trascendentales de mi vida, porque me ha visto llorar y reír, y sin importar mi estado de ánimo, siempre ha estado junto a mí. A mi sobrino David Alejandro Revelo Hoyos por alegrarme la vida.

Le agradezco a Lina por todo su amor, por estar conmigo incondicionalmente, por estar dispuesta a vivir conmigo momentos felices y tristes, por hacer que en los momentos tristes pueda sonreír... te adoro.

Gracias a mis amigos y compañeros que han hecho parte de mi formación, a los que están lejos persiguiendo sus sueños y a los que pueden compartir conmigo esta alegría. En especial agradezco a mi amigo y compañero Daniel Usama, con quien he compartido estos últimos meses en la carrera.

Le agradezco a nuestro director de trabajo de grado Juan Fernando Florez por su empeño en el proyecto, por querer que hagamos siempre las cosas bien, por todo su tiempo y dedicación en este trabajo. Quiero agradecer de forma especial a los jurados del trabajo de grado Luis Fernando Echeverry y Leonairo Pencue por sus sugerencias y correcciones que resultaron muy valiosas para el desarrollo de este trabajo de investigación. Finalmente agradezco al programa de Ingeniería Física de la Universidad del Cauca.

David Armando Revelo Luna  
Popayán, 28 de junio 2010

## **AGRADECIMIENTOS**

Agradezco a Dios por darme fortaleza y esperanza en todos los momentos de mi vida.

A mis padres, Rosa María Oviedo y Julio Cesar Usama por brindarme su apoyo, porque son los que alimentan mis sueños, me ayudan a superar mis obstáculos y me impulsan a cumplir mis metas.

Agradezco a mis hermanos Luis, Ivan y Martha por estar conmigo en los momentos felices y en los momentos tristes, por el amor que me brindan y por brindarme una sincera amistad.

Agradezco a Carolina por todo su amor y apoyo, por sus consejos y porque siempre está a mi lado en los buenos y malos momentos. A mis amigos, Diana, Jorge, Stephanie, Paula, David y Lina, por brindarme una amistad sincera, por alegrarme la vida, por sus buenos consejos y por aguantar mis locuras.

Agradezco a nuestro director Juan Fernando Florez, por guiarnos en este proceso, por compartir sus conocimientos y por ser una excelente persona.

Gracias a Luis Fernando Echeverry y a Leonairo Pencue por brindarme sus conocimientos.

Francisco Daniel Usama Oviedo

Popayán, 28 de junio 2010

## RESUMEN

En la actualidad, técnicas de visión artificial son ampliamente estudiadas por ser de importante aplicación en la robótica. No obstante, la mayoría de trabajos se han implementado bajo imágenes bidimensionales, suponiendo una fuerte restricción en el amplio campo de posibilidades de la visión de máquina. Para dar solución a esta limitación se han desarrollado, durante la última década, estructuras que permiten obtener la información tridimensional de una escena, siendo una de las más utilizadas la constituida por un par estereoscópico de cámaras a semejanza de la visión humana.

En este trabajo se implementó un sistema de calibración y rectificación estereoscópica. Se utilizó el método de *Brown* para determinar los parámetros de distorsión, y el método de *Hartley* para la rectificación. El sistema estereoscópico consta de un módulo *hardware*, cuya función es la de alinear los ejes ópticos del par estereoscópico y un módulo *software* de rectificación de imágenes. Los resultados muestran que se logra obtener un mapa de disparidades del entorno, donde se obtiene la mayor resolución en la medida de profundidad a mayor distancia de *baseline* (separación entre ejes ópticos), y el rango se amplía al aumentar la separación entre las cámaras. Con el sistema de visión estereoscópico se confirmó la relación inversa que existe entre la disparidad y la profundidad, se verificó que el proceso de correspondencia es mucho mejor en escenarios con abundante textura. Se aplicaron técnicas de *flujo óptico* y *suma absoluta de diferencias (SAD)* para realizar la correspondencia estéreo. Finalmente se logró reconstruir escenas en 3D partiendo de imágenes estáticas y de video.

Tanto el dispositivo de visión estéreo como los algoritmos fueron probados en la plataforma robótica Player bajo el entorno Linux, siendo fieles a la idea de un sistema

robótico integrado que permita articular varios sensores de diferentes tipos, que a su vez se puedan comunicar bajo un mismo estándar de mensajes.

En robótica un aspecto importante, en la adquisición y manipulación de los datos enviados por los sensores, es el tiempo de cómputo que emplean los algoritmos en procesar la información. En este trabajo el procesamiento de las imágenes estéreo se realiza por regiones y extracción de características, con bajo coste computacional lo que permite ser implementado en secuencia de video usando un computador de gama media.

# Tabla de contenido

AGRADECIMIENTOS.....	IV
AGRADECIMIENTOS.....	V
RESUMEN .....	VI
LISTA DE TABLAS .....	XI
LISTA DE FIGURAS.....	XII
INTRODUCCIÓN .....	1
CAPÍTULO 1 .....	4
MODELO GEOMETRICO Y CALIBRACIÓN DE CÁMARA .....	4
1.1 INTRODUCCIÓN .....	4
1.2 MODELOS DE CÁMARAS .....	5
1.2.1. Modelo libre de distorsiones (cámara <i>pin-hole</i> ).....	5
1.2.2. Modelos de cámara con distorsiones geométricas.....	8
1.2.3 Simulación de los modelos de distorsiones geométricas .....	12
1.3 CÁMARA CCD y CMOS.....	13
1.3.1 Responsividad.....	15
1.3.2 Rango Dinámico (RD) .....	15
1.3.3 Ruido .....	15
1.3.4 Respuesta Uniforme.....	16
1.3.5 Velocidad .....	16
1.3.6 <i>Blooming</i> .....	16
1.4 CAMARA UTILIZADA EN EL PROYECTO .....	17
1.5 PROCESO DE CALIBRACIÓN GEOMÉTRICA.....	18
1.5.1 Métodos de calibración.....	18



<b>CAPÍTULO 2 .....</b>	<b>24</b>
<b>TÉCNICA DE VISIÓN ESTEREOSCÓPICA .....</b>	<b>24</b>
<b>2.1 INTRODUCCION .....</b>	<b>24</b>
<b>2.2 GEOMETRIA ESTEREOSCÓPICA.....</b>	<b>25</b>
2.2.1 Cámaras no paralelas .....	27
2.2.2 Geometría Epipolar .....	28
2.2.3 La matriz esencial E y la matriz fundamental F .....	30
2.2.4 Alineación <i>off-line</i> .....	34
2.2.5 Proceso de Rectificación .....	35
<b>2.3 PREPROCESAMIENTO DE IMÁGENES.....</b>	<b>37</b>
2.3.1 Filtros pasa bajas .....	38
2.3.2 Filtros pasa altas.....	39
<b>2.4 CORRESPONDENCIA ESTÉREO .....</b>	<b>40</b>
2.4.1 Suma de diferencias absolutas SAD.....	41
2.4.2 Flujo óptico .....	43
<b>2.5 DETERMINACIÓN DE PROFUNDIDAD Y RECONSTRUCCIÓN 3D .....</b>	<b>46</b>
<b>CAPÍTULO 3 .....</b>	<b>51</b>
<b>IMPLEMENTACIÓN DEL SISTEMA DE VISIÓN ESTÉREO .....</b>	<b>51</b>
<b>3.1 INTRODUCCIÓN .....</b>	<b>51</b>
<b>3.2 SISTEMA ESTÉREO HARDWARE .....</b>	<b>52</b>
3.2.1 Requisito de Alineación de ejes ópticos.....	52
3.2.2 Vista 3D del hardware.....	55
3.2.3 Implementación del hardware .....	56
<b>3.3 SISTEMA ESTÉREO SOFTWARE .....</b>	<b>57</b>
3.3.1 Captura de video .....	57
3.3.2 Corrección de distorsiones geométricas .....	58
3.3.3 Rectificación software.....	62
3.3.4 Disparidad densa .....	65
3.3.5 Disparidad por extracción de características .....	69

3.3.6 Restricción epipolar.....	72
3.3.7 Restricción de máxima disparidad.....	73
3.3.8 Representación 3D .....	75
<b>3.4 CARACTERIZACIÓN DEL SISTEMA ESTÉREO .....</b>	<b>81</b>
3.4.1 Profundidad vs. disparidad .....	82
3.4.2 Rango .....	88
3.4.3 Sensibilidad.....	88
3.4.4 Coste Computacional .....	89
<b>CAPÍTULO 4 .....</b>	<b>94</b>
<b>APLICACIÓN DE LA TECNICA DE VISIÓN ESTÉREO A LA CONSTRUCCION DE MAPAS .....</b>	<b>94</b>
4.1 INTRODUCCIÓN .....	94
4.2 Sistema de visión estéreo y Player/Stage.....	95
4.2 Construcción de mapas en interiores usando visión estereoscópica .....	97
4.2 Respuesta del sistema de visión estéreo en exteriores .....	101
<b>CONCLUSIONES .....</b>	<b>104</b>
<b>TRABAJOS FUTUROS.....</b>	<b>107</b>
<b>REFERENCIAS.....</b>	<b>108</b>

## **LISTA DE TABLAS**

Tabla 1.1 Características de la cámara utilizada	17
Tabla 3.1. Parámetros intrínsecos	61
Tabla 3.2 Valores relacionados con la lineación de las curvas en la Fig. 3.28	86
Tabla 3.3 Características del computador y tiempos de ejecución	93

## LISTA DE FIGURAS

Fig. 1.1	Modelo pin-hole	5
Fig. 1.2	Proyección en el plano $(Y_c, Z_c)$	7
Fig. 1.3	Simulación de distorsiones	13
Fig. 1.4	Cámara webcam ST-HC-328	17
Fig. 2.1	Modelo básico de visión estéreo	25
Fig. 2.2.	Configuración estéreo real y alineación matemática necesaria para alineación	26
Fig. 2.3.	Geometría epipolar bifocal	29
Fig. 2.4.	Transformación geométrica de un punto en $O_l$ a su correspondiente en $O_r$ .	31
Fig. 2.5	Filtros pasa bajas, mecanismo de aplicación	38
Fig. 2.6	Ejemplos de <i>kernel</i> pasa altas	40
Fig. 2.7	Búsqueda de correspondencia	42
Fig. 2.8	Recorrido sobre la línea epipolar	43
Fig. 2.9	Flujo óptico aplicado al problema de búsqueda de correspondencia	45
Fig. 2.10.	Superior imagen en simulación, inferior mapa de disparidad	47
Fig.2.11.	marco de referencia utilizado en la geometría de triangulación	48
Fig. 2.12	Dos posibles representaciones de los datos 3D	50
Fig. 3.1.	Nivelación de los planos imagen del par estereoscópico	52
Fig. 3.2.	Posibles rotaciones de la cámara móvil para alineación de planos imagen	53
Fig. 3.3.	Rotación del eje de la cámara generada con el tornillo micrométrico	54
Fig. 3.4.	Diseño del soporte de visión estereoscópica en <i>Solid Edge</i>	55
Fig. 3.5	Piezas del sistema estéreo implementado	56
Fig. 3.6	Poses del tablero para la calibración de cámara	60
Fig. 3.7	Tableros para rectificación de cámaras	63
Fig. 3.8	Pares de imágenes usadas en el proceso de rectificación	64
Fig. 3.9	Proceso de rectificación	65

Fig. 3.10	Disparidad densa con <i>baseline</i> de 4cm	66
Fig. 3.11	Efecto del filtro de media en los mapas de disparidad	67
Fig. 3.12	Representación del mapa de disparidad en pseudo-color	68
Fig. 3.13	Perfil de disparidad	69
Fig. 3.14	Extracción de características con filtro Sobel	70
Fig. 3.15	Eliminación uniforme de puntos característicos	71
Fig. 3.16	Búsqueda de correspondencia basada en flujo óptico	72
Fig. 3.17	Correspondencia usando flujo óptico aplicando restricción epipolar	73
Fig. 3.18	Correspondencia estéreo (pirámide con textura)	74
Fig. 3.19	Correspondencia de características relevantes de la escena	75
Fig. 3.20	Escena a reconstruir (con textura)	76
Fig. 3.21	Reconstrucción tridimensional en Matlab	77
Fig. 3.22	Reconstrucción tridimensional con texturas en Matlab	78
Fig. 3.23	Reconstrucción tridimensional con texturas con Glut	79
Fig. 3.24	Reconstrucción tridimensional con texturas con Glut	80
Fig. 3.25	Reconstrucción tridimensional partiendo de disparidad de características con Glut	81
Fig. 3.26	Profundidad vs Disparidad, <i>baseline</i> =40mm	82
Fig. 3.27	Profundidad vs Disparidad, <i>baseline</i> =60mm	83
Fig. 3.28	Profundidad vs Disparidad	83
Fig. 3.29	Profundidad vs 1/Disparidad, <i>baseline</i> =40mm	84
Fig. 3.30	Profundidad vs 1/Disparidad, <i>baseline</i> =60mm	84
Fig. 3.31	Linealización, <i>baseline</i> =40mm	85
Fig. 3.32	Linealización, <i>baseline</i> =60mm	85
Fig. 3.33	Variación del ángulo de calibración, <i>baseline</i> =60mm	87
Fig. 3.34	Variación del ángulo entre ejes ópticos, <i>baseline</i> =60mm	87
Fig. 3.35	Función sensibilidad	89
Fig. 3.36	Coste computacional, método disparidad por extracción de características	90

Fig. 3.37 Variación del tiempo de procesamiento respecto al número de puntos de búsqueda	91
Fig. 3.38 Coste computacional, método disparidad por regiones	92
Fig. 4.1 Trayectoria cerrada generada en Stage, usando la visión estéreo como sensor detector de obstáculos	98
Fig. 4.2. Trayectoria con bifurcaciones generada en Stage, usando el sistema de visión estéreo como elemento sensor	99
Fig. 4.3 Reconstrucción tridimensional con disparidad densa.	100
Fig. 4.4 Planos de profundidad, poca resolución para lugares con mayor profundidad	101
Fig. 4.5 Respuesta del algoritmo ante objetos lejanos	102
Fig. 4.6 Respuesta del algoritmo un obstáculo dentro del umbral establecido	103

---

# INTRODUCCIÓN

---

La determinación precisa de coordenadas tridimensionales, de uno o varios objetos en una escena, es de gran interés en aplicaciones en las que se requiera detectar obstáculos y cambios de elevación en un terreno, como por ejemplo en el guiado de robots móviles. Existe un buen número de aplicaciones relacionadas con el anterior aspecto como la navegación autónoma [1], fotogrametría [2], aplicaciones médicas [3], control de calidad y operación remota. La visión estereoscópica, basada en la triangulación entre un punto de la escena y al menos dos proyecciones de este punto sobre imágenes tomadas de distintas perspectivas, es una de las técnicas más utilizadas para la reconstrucción tridimensional. Hoy en día los sistemas de visión estéreo han cobrado gran fuerza gracias a los adelantos tecnológicos como cámaras de menor costo y computadores personales con mayor capacidad de procesamiento.

A pesar de que la estereoscopia es una técnica atractiva para la percepción de la profundidad, porque reduce la ambigüedad de las mediciones monoculares y, a diferencia de los métodos activos (radares y *laser range finders*), es utilizable en un mayor rango de aplicaciones [4], pero posee ciertas desventajas, como por ejemplo:

- \_ La incapacidad de los algoritmos binoculares en la obtención de correspondencias en zonas ocluidas.
  
- \_ Correspondencias correctas que pueden ser rechazadas por el algoritmo de búsqueda de correspondencias.
  
- \_ Correspondencias falsas aceptadas por error en el algoritmo de búsqueda de correspondencia.

\_ Ambigüedad causada por la presencia de patrones periódicos en la escena.

\_ La falta de información en la escena causada por la ausencia de texturas.

Limitaciones que, igualmente, son fuertemente influenciados por la incertidumbre generada en la discretización de las imágenes y la desalineación de los sensores en un sistema estéreo real.

Es así como en el presente trabajo se pretende contribuir al mejoramiento de las técnicas basadas en estereoscopia para la reconstrucción tridimensional de escenas reales. En principio, se realizó el diseño de un soporte para cámaras tipo webcam en configuración estereoscópica, el soporte permite fijar una de las cámaras y mover la otra en dos grados de libertad, uno de ellos de forma manual y el otro usando un tornillo micrométrico. El movimiento de la cámara permite alinear *off-line* el sistema, de esta forma se pretende obtener ejes ópticos paralelos. Adicionalmente el soporte permite ubicar las cámaras a 4cm y 6cm de separación, ya que es de interés para el trabajo de investigación conocer la relación entre la separación de cámaras y las características del sistema de visión como el rango y la resolución.

En un sistema de visión que pretende medir distancias en una escena real, es de gran importancia la calibración de la cámara [5], [6] y fundamentalmente cumplir, si es un sistema estéreo, con la geometría de ejes ópticos paralelos. Para ello, finalmente, se diseña un procedimiento software de “rectificación de imágenes”. Para facilitar el cumplimiento de estos dos requisitos se realizó una interfaz hombre máquina que permite: calibrar las dos cámaras, izquierda y derecha, tomar las imágenes para el proceso de rectificación, realizar el proceso de rectificación de imágenes, corregir las distorsiones de las dos cámaras y verificar la rectificación del par estereoscópico. Son siete procedimientos necesarios para la calibración y alineación de cámaras que se



integraron en una única interfaz, que permite en poco tiempo, alrededor de 3 minutos, obtener imágenes aceptables para los algoritmos de visión estéreo.

Existen dos técnicas sobresalientes en los sistemas de visión estéreo, basadas en disparidad por regiones y en disparidad por características. En el presente trabajo se presentan resultados de los dos métodos. A partir de los mapas de disparidad se realiza la reconstrucción tridimensional de escenas reales usando mallas (procedimiento *off-line*) y usando representación de puntos (procedimiento *on-line*).

En robótica es de gran interés la integración de dispositivos bajo una única plataforma de procesamiento, que le permita al sistema de mando del robot tomar y compartir datos de diversos sensores simultáneamente, y bajo el mismo estándar de mensajes. Es por esta razón que se decidió hacer uso de la plataforma robótica Player para tal fin y se integró, a la misma, el sistema estéreo desarrollado.

Finalmente se caracterizó el sistema determinando su resolución y rango, analizando la variación que sufren estos parámetros debido a la separación entre los centros ópticos de las cámaras.

---

## Capítulo 1

# MODELO GEOMETRICO Y CALIBRACIÓN DE CÁMARA

---

### 1.1 INTRODUCCIÓN

Un alto número de aplicaciones de visión de máquina requieren realizar el proceso denominado calibración de cámara. Este proceso consiste en determinar los parámetros intrínsecos de la cámara como: distancia focal, coeficientes de distorsión y puntos centrales del plano imagen. En la literatura se exponen varios tipos de procesos de calibración de imágenes [6], del mismo modo existen varios tipos de modelos de la cámara.

Las dos principales causas que generan distorsiones en las imágenes de las cámaras comunes son:

- \_ La curvatura de las lentes usadas tienen imperfecciones, lo cual ocasiona que la reflexión de la luz en algunos lugares de la lente sea defectuosa.
- \_ En el proceso de ensamblaje de las cámaras, muchos de sus componentes no quedan perfectamente alineados, por ejemplo los arreglos de los sensores de las cámaras pueden tener deformación en su geometría.

En todo sistema de visión por computador en el que se requiera extraer información métrica a partir de imágenes 2D del mundo 3D, se hace necesario realizar el proceso de calibración de cámara [7].

En la sección 1.2 se describe el modelo de la cámara libre de distorsiones y el modelo de la cámara con distorsiones geométricas. En la sección 1.3 se presentan algunos aspectos

generales sobre las cámaras CCD y CMOS. En la sección 1.4 se presenta la cámara utilizada en el trabajo. En la sección 1.5 se exponen los dos métodos de calibración geométricas.

## 1.2 MODELOS DE CÁMARAS

### 1.2.1. Modelo libre de distorsiones (cámara *pin-hole*)

El modelo de cámara *pin-hole* es el modelo de cámara más simple y permite expresar la proyección del espacio (la escena) 3D a través de un punto (el foco) a un punto 2D (plano imagen), supone que para todo punto que impacta en el plano, el rayo de luz que sale rebotado de un cuerpo y llega a la cámara, atraviesa un único punto, independientemente del punto de origen y del punto de impacto en el plano de la imagen [8]. Es un modelo sencillo y ampliamente usado.

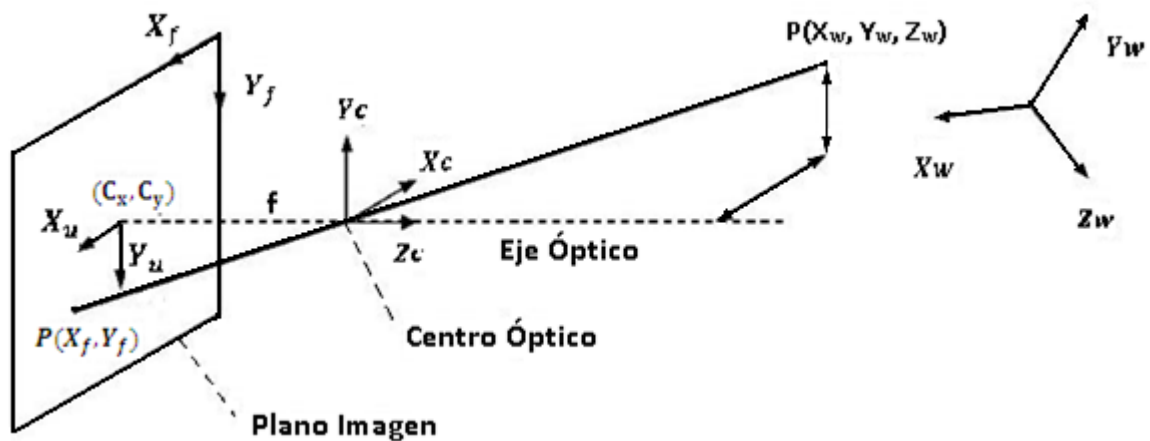


Fig. 1.1. Modelo pin-hole.

Antes de iniciar con la descripción matemática del modelo *pin-hole* se deben tener en cuenta las siguientes definiciones:

- Centro de la cámara ( $O_c$ ), o centro óptico. Punto donde se localiza la cámara. Por defecto se considera en el centro de coordenadas, observando en dirección de Z negativo y con Y positivo hacia arriba.
- Plano focal o de la imagen. Plano donde se forma la imagen, a una distancia  $f$  del origen.
- Eje óptico. Línea perpendicular al plano de la imagen que pasa por el centro de la cámara.

Sea  $P(X_w, Y_w, Z_w)$  un punto cualquiera del mundo real con  $O$  como punto de referencia, y  $Q(X_c, Y_c, Z_c)$  la representación del punto  $P$  en el sistema de coordenadas con origen en el centro de la cámara ( $O_c$ ), donde  $Z_c$  coincide con el eje óptico, entonces la representación matricial de un punto  $P$  en las coordenadas de la cámara ( $X_c, Y_c, Z_c$ ) se puede realizar a través de una rotación y una traslación, según la ecuación 1.1:

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix} = R \begin{bmatrix} X_w \\ Y_w \\ Z_w \end{bmatrix} + T \quad (1.1)$$

Donde  $R$  denota una matriz de rotación, que indica la orientación de la cámara y  $T$  un vector de traslación que indica la posición de la cámara. La ecuación 1.1 también puede expresarse en forma reducida según indica la ecuación 1.2:

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} \quad (1.2)$$

Si ahora se quiere expresar el punto  $P$  en las coordenadas del plano imagen ( $X_u, Y_u$ ) con origen en el punto principal ( $O_i, (C_x, C_y)$ ), entonces tomando la proyección de la línea tridimensional en el plano ( $Y_c, Z_c$ ), como en la Fig. 2, y haciendo unos pequeños cálculos, se obtiene la ecuación 1.3:

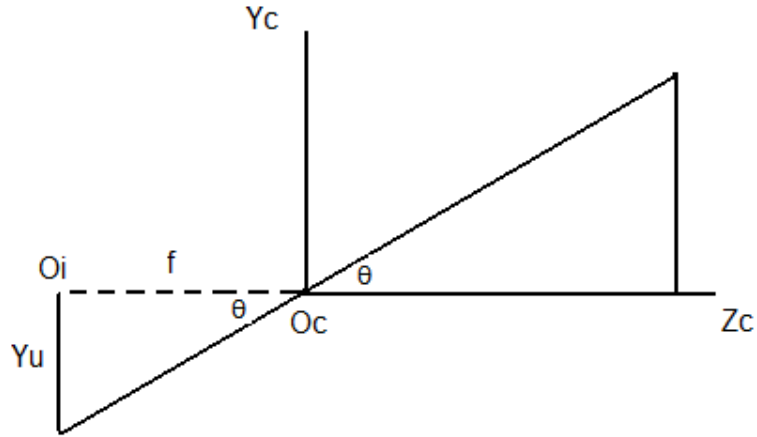


Fig. 1.2. Proyección en el plano  $(Y_c, Z_c)$ . Fuente propia.

Como:

$$\tan \theta = \frac{Y_c}{Z_c} = \frac{Y_u}{f} \quad \Rightarrow \quad Y_c = \frac{Z_c}{f} Y_u$$

Procediendo de la misma forma, pero ahora proyectando en el plano  $(X_c, Z_c)$  se tiene:

$$\frac{X_c}{Z_c} = \frac{X_u}{f} \quad \Rightarrow \quad X_c = \frac{Z_c}{f} X_u$$

Entonces, definiendo un factor de escala  $n = \frac{Z_c}{f}$ , se obtiene 1.3:

$$\begin{bmatrix} nX_u \\ nY_u \\ n \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{1}{f} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} \cong \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} \quad (1.3)$$

Unificando las ecuaciones 1.2 y 1.3 se obtiene las coordenadas centrales de la cámara en función de las coordenadas del mundo, según ilustra la ecuación 1.4:

$$\begin{bmatrix} nX_u \\ nY_u \\ n \\ 1 \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} \quad (1.4)$$

Por último se transforma del sistema coordenado  $(X_u, Y_u)$  centrado en  $O_i$  al sistema coordenado  $(X_f, Y_f)$ , las cuales se denominan coordenadas laterales y están dadas en pixeles, para lo cual se necesitan las constantes  $K_x$  y  $K_y$  con el fin de hacer posible el cambio de coordenadas; además el punto  $O_i$  se representa en el sistema  $(X_f, Y_f)$  como  $(C_x, C_y)$ , teniendo en cuenta esto se obtiene la ecuación 1.5:

$$\begin{aligned} X_f &= K_x X_u + C_x \\ Y_f &= K_y Y_u + C_y \end{aligned} \tag{1.5}$$

En notación matricial 1.5 se representa según la ecuación 1.6:

$$\begin{bmatrix} nX_f \\ nY_f \\ n \end{bmatrix} = \begin{bmatrix} K_x f & 0 & C_x & 0 \\ 0 & K_y f & C_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} \tag{1.6}$$

A la ecuación 1.6 se la denomina la matriz de proyección M.

### 1.2.2. Modelos de cámara con distorsiones geométricas

La ecuación 1.6 describe una situación ideal pero esto no es cierto en la mayoría de las cámaras, debido a imperfecciones en las lentes y a que el chip sensor matricial no está totalmente paralelo a la lente. Es por esta razón que se generan distorsiones geométricas, las cuales se refieren a las posiciones erróneas de los puntos en el plano imagen [9].

Estas distorsiones generan errores en las posiciones reales  $(X_u, Y_u)$  y se simbolizan  $(\xi x, \xi y)$ , para representar el error en la coordenada X y el error en la coordenada Y respectivamente, ver ecuación 1.7.

$$\begin{aligned} X_d &= X_u + \xi x \\ Y_d &= Y_u + \xi y \end{aligned} \tag{1.7}$$

Por lo tanto las coordenadas  $(X_f, Y_f)$  con distorsión se representan por  $(X_f', Y_f')$ , según la ecuación 1.8:

$$\begin{aligned} X_f' &= K_x X_d + C_x \\ Y_f' &= K_y Y_d + C_y \end{aligned} \tag{1.8}$$

Relacionando las ecuaciones 1.6 y 1.7 se obtiene la ecuación 1.9:

$$\begin{bmatrix} nX_f' \\ nY_f' \\ n \end{bmatrix} = \begin{bmatrix} K_x f & 0 & (C_x + K_x \xi_x) & 0 \\ 0 & K_y f & (C_y + K_y \xi_y) & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} \tag{1.9}$$

Por lo tanto las coordenadas en el plano imagen, teniendo en cuenta las distorsiones geométricas se expresan como en la ecuación 1.10:

$$\begin{aligned} X_f' &= C_x + K_x \xi_x + K_x f \frac{r_{11}X_w + r_{12}Y_w + r_{13}Z_w + t_x}{r_{31}X_w + r_{32}Y_w + r_{33}Z_w + t_z} \\ Y_f' &= C_y + K_y \xi_y + K_y f \frac{r_{21}X_w + r_{22}Y_w + r_{23}Z_w + t_y}{r_{31}X_w + r_{32}Y_w + r_{33}Z_w + t_z} \end{aligned} \tag{1.10}$$

A continuación se analizan los tipos de distorsión, para encontrar el modelo matemático de  $\xi_x$  y  $\xi_y$ .

### 1.2.2.1. Distorsión radial

Una cámara que presenta el fenómeno de distorsión radial, distorsiona las líneas rectas de la escena real, representándolas en sus imágenes como líneas curvas. Debido a que esta distorsión falsifica la información geométrica de la escena real, es importante corregir esta aberración antes de abordar el problema de extraer información útil de la imagen.

Este tipo de distorsión se presenta por el defecto en el radio de curvatura de las lentes y causa un desplazamiento de los pixeles hacia dentro o hacia fuera a lo largo del radio. La cantidad de distorsión radial que se genera al eje coordenado X y al eje coordenado Y se representa como  $\xi_{xr}$  y  $\xi_{yr}$  respectivamente, y las ecuaciones que representan estas cantidades se representan en 1.11:

$$\begin{aligned}\xi_{xr} &= K_1 X_d (X_d^2 + Y_d^2) + K_2 (X_d^2 + Y_d^2)^2 + O[(X_d, Y_d)^5] \\ \xi_{yr} &= K_1 Y_d (X_d^2 + Y_d^2) + K_2 Y_d (X_d^2 + Y_d^2)^2 + O[(X_d, Y_d)^5]\end{aligned}\tag{1.11}$$

donde  $K_1$  y  $K_2$  son constantes que determinan la cantidad y dirección de la distorsión y  $O[(X_d, Y_d)^5]$  representan los términos de orden mayor que cuatro.

### 1.2.2.2. Distorsión descentral

Se debe a defectos en la alineación de los centros ópticos de las lentes que conforman la óptica de la cámara y causa desplazamientos radiales y tangenciales. La imagen tendrá un eje para el que la distorsión tangencial sea nula (eje libre de distorsión) y otro eje para el que esta distorsión sea máxima. Esta distorsión responde a una expresión con dos parámetros  $P_1$  y  $P_2$  que se refleja en la ecuación 1.12.

$$\begin{aligned}\xi_{xd} &= P_1 (3X_d^2 + Y_d^2) + 2P_2 X_d Y_d + O[(X_d, Y_d)^4] \\ \xi_{yd} &= 2P_1 X_d Y_d + P_2 (X_d^2 + 3Y_d^2) + O[(X_d, Y_d)^4]\end{aligned}\tag{1.12}$$



donde  $P_1$  y  $P_2$  son constantes que determinan la cantidad y dirección de la distorsión así como el eje máximo de distorsión tangencial.

### 1.2.2.3. Distorsión prismática

Se origina por imperfecciones en la fabricación y montaje de las lentes y provoca desplazamientos radiales y tangenciales. Este tipo de distorsión puede ser modelado por adición de un delgado prisma al eje óptico del sistema y se puede expresar según 1.13:

$$\begin{aligned}\xi_{xp} &= S_1(X_d^2 + Y_d^2) + O[(X_d, Y_d)^4] \\ \xi_{yp} &= S_2(X_d^2 + Y_d^2) + O[(X_d, Y_d)^4]\end{aligned}\tag{1.13}$$

donde  $S_1$  y  $S_2$  son constantes que determinan la cantidad y dirección de la distorsión.

### 1.2.2.4. Distorsión total

Las distorsiones descentral y prismática tienen unos coeficientes similares, pero son modelos de distorsiones diferentes y tienen distintos ejes de máxima distorsión. La distorsión total se puede expresar como suma de estos tres tipos, tal y como se expresa en la ecuación 1.14 en las que se han agrupado los efectos vistos, teniendo solo en cuenta las componentes de primer orden.

$$\begin{aligned}\xi_x &= S_1(X_d^2 + Y_d^2) + 3P_1X_d^2 + P_1Y_d^2 + 2P_2X_dY_d + K_1X_d(X_d^2 + Y_d^2) \\ \xi_y &= S_2(X_d^2 + Y_d^2) + 2P_1X_dY_d + P_2X_d^2 + 3P_2Y_d^2 + K_1Y_d(X_d^2 + Y_d^2)\end{aligned}\tag{1.14}$$

Es común agrupar las distorsiones en radiales y tangenciales. De las expresiones conocidas se deducen las ecuaciones 1.15. Éstas son las ecuaciones que se van a utilizar como caso más general de distorsión. En ellas aparecen dos coeficientes  $K_1$  y  $K_2$  para la distorsión radial y otros dos  $P_1$  y  $P_2$  para la tangencial. Estos cuatro coeficientes son típicamente suficientes para describir la distorsión total.

$$\xi_{xr} = X_d(K_1 r_d^2 + K_2 r_d^4)$$

$$\xi_{yr} = Y_d(K_1 r_d^2 + K_2 r_d^4)$$

$$\xi_{xt} = 2P_1 X_d Y_d + P_2(2X_d^2 + r_d^2)$$

$$\xi_{yt} = P_1(r_d^2 + 2Y_d^2) + 2P_2 X_d Y_d$$

(1.15)

donde  $\xi_{xr}$  y  $\xi_{yr}$  son las distorsiones radiales en el eje coordenado X e Y respectivamente.  $\xi_{xt}$  y  $\xi_{yt}$  son las distorsiones tangenciales en el eje coordenado X e Y respectivamente, con  $r_d = \sqrt{(X_d^2 + Y_d^2)}$ .

Para ver el efecto de la distorsión radial y tangencial se utilizó la herramienta Matlab, para generar distorsiones radiales y tangenciales en una imagen sin distorsiones como se muestra en la Fig. 1.3.

### 1.2.3 Simulación de los modelos de distorsiones geométricas

Para entender como las distorsiones afectan una imagen libre de distorsiones, se realizó la simulación de las 4 distorsiones antes detalladas. En la Fig. 1.3 se muestra la imagen original junto con los dos tipos de distorsiones mencionados y la distorsión total.

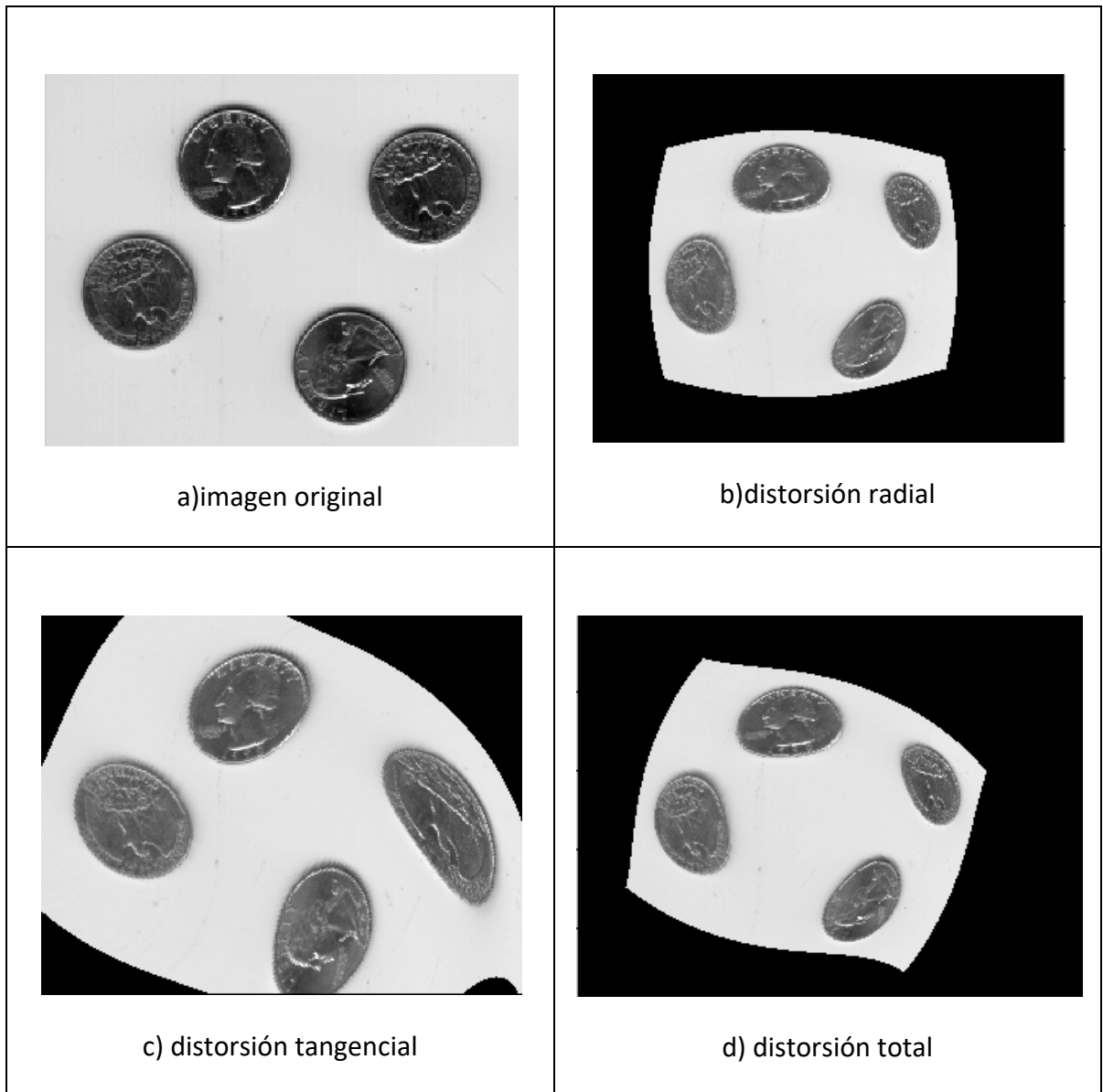


Fig. 1.3 Simulación de distorsiones, a) imagen sin distorsión, b) con distorsión radial, c) con distorsión tangencial y d) con distorsión total. Fuente propia.

### 1.3 CÁMARA CCD y CMOS

Actualmente las cámaras se pueden clasificar según su tecnología principalmente en cámaras CCD (*Charge Coupled Device* o Dispositivo de Cargas Acopladas), que fueron los primeros en aparecer en el mercado, y las más recientes CMOS (*Complementary Metal Oxide Semiconductor* - Semiconductor de Óxido de Metal Complementario). Cada una de

estas tecnologías tiene sus ventajas y desventajas y por lo tanto se puede elegir entre cada una de ellas, dependiendo de la aplicación [10].

Tanto los sensores CCD como los CMOS están fabricados con materiales semiconductores, concretamente de Metal-Óxido (MOS) y están estructurados en forma de una matriz, con filas y columnas. Funcionan al acumular una carga eléctrica en cada celda de esta matriz en proporción a la intensidad de la luz que incide sobre ella localmente. A mayor intensidad luminosa, mayor carga acumulada.

El CCD convierte finalmente estas cargas en voltajes y entrega una señal analógica a su salida, que debe ser digitalizada y procesada por la circuitería de la cámara. Una forma de entender cómo funciona es imaginarse al sensor como un arreglo (matricial) en dos dimensiones con miles (o millones) de celdas solares en miniatura, donde cada una de las celdas convierte la luz de una pequeña porción de la imagen (un píxel) en electrones. Lo siguiente es realizar la lectura del valor correspondiente en cada una de las celdas. En un sensor CCD, la información de cada una de las celdas es enviada a través del chip hacia una de las esquinas del arreglo, y ahí un convertidor analógico a digital traduce el valor de cada una de las celdas. De esta manera, se mantiene simple la estructura del sensor, a costa de la necesidad de una circuitería adicional importante que se encargue del tratamiento de los datos recogidos.

Al contrario que en los sensores de tecnología CCD, las celdas de la matriz CMOS son totalmente independientes de sus vecinas. La principal diferencia radica en que en estos sensores la digitalización se realiza píxel a píxel dentro del mismo sensor, por lo que la circuitería es mucho más sencilla. En cada celda de una matriz CMOS se encuentran varios transistores, conformando cada uno de los pixeles del sensor, que amplifican y procesan la información recogida. Esta manera de efectuar la lectura de la imagen es más flexible, ya que cada pixel se lee de manera individual.

La primera diferencia, y la más obvia, es el precio. En general, los sensores CMOS tienden a ser más económicos que los CCD. Por supuesto, no se puede comparar una tecnología con otra solamente basándose en su costo, por lo que es necesario tener en cuenta otros factores, como los descritos a continuación.

### **1.3.1 Responsividad**

Se define con este término al nivel de señal que es capaz de ofrecer el sensor por cada unidad de energía óptica incidente. Dado que interesa que el sensor tenga una responsividad elevada, se necesita que con poca cantidad de luz, se obtenga una señal aceptable. Con esto en mente, se puede ver que los sensores CMOS son superiores a los de tecnología CCD, debido a que integran elementos amplificadores en cada celda. Además el sistema de construcción CMOS permite una alta amplificación con un bajo consumo, mientras que en CCD la amplificación al ser externa al sensor supone un consumo más elevado.

### **1.3.2 Rango Dinámico (RD)**

Es el cociente entre el nivel de saturación de los píxeles y el umbral por debajo del cual no captan señal. En este aspecto CCD supera a CMOS en un factor de dos, dado que en la actualidad el RD de un sensor CCD es típicamente del doble que el de un CMOS.

### **1.3.3 Ruido**

Los sensores CCD aventajan a los de tecnología CMOS en este apartado, ya que debido a su construcción todo el procesado de señal se da fuera del CCD, con lo que se podrá aprovechar la mejor tecnología disponible en convertidores A/D, pudiendo elegir los más rápidos o mejores procesadores. Por su parte, los CMOS al realizar la gran mayoría de las

funciones dentro del sensor (amplificación, conversión A/D, etc.) destinan menos espacio para los fotodiodos encargados de recoger la luz, por lo que se produce más ruidos en la lectura.

### **1.3.4 Respuesta Uniforme**

Idealmente se espera que un píxel sometido al mismo nivel de excitación de luz que sus vecinos no presente cambios apreciables respecto de ellos. En este aspecto la individualidad de cada píxel de un sensor CMOS los hace más sensibles a sufrir fallos, siendo mayor la uniformidad en CCD. No obstante mediante la adición de circuitos con realimentación se ha conseguido subsanar en gran medida este problema, aunque los CCD siguen ostentando una ligera ventaja.

### **1.3.5 Velocidad**

Concentrándose en la velocidad con la que se captura la imagen, se nota que los CMOS son bastante superiores a los CCD, debido a que muchas funciones, como la propia conversión analógico-digital son realizadas en el propio sensor. Si bien por el momento esta ventaja es ligera, se espera que aumente con el tiempo.

### **1.3.6 Blooming**

Esta palabra define el fenómeno por el cual un píxel que ha alcanzado la saturación empieza a saturar a sus vecinos, creando efectos y patrones no deseados. Esto se da principalmente en sensores CCD, que necesitan de algunos trucos de construcción para evitarlos. En cambio, gracias a su construcción los sensores CMOS no sufren de este defecto.

## 1.4 CAMARA UTILIZADA EN EL PROYECTO

Para este trabajo se utilizaron dos cámaras tipo webcam ST-HC-328 de marca STAR TEC, ver Fig. 1.4, las especificaciones de estas cámaras se muestran en la tabla 1.1.



Fig. 1.4. Cámara webcam ST-HC-328. Fuente [www.starktechnology.com](http://www.starktechnology.com)

**Tabla 1.1** Características de la cámara utilizada. Fuente propia.

CARACTERÍSTICA	ESPECIFICACIÓN
Fabricante	STAR TEC
Tecnología	CMOS
Definición	1.3 Mega pixeles
Rango del foco	5 cm – infinito
Formato de video	24 bit RGB
Velocidad de frame	320x240 hasta 30 frame/s
Tamaño del sensor	4.86 x 3.64 mm
Rango dinámico	72 db
Interface	Usb

## **1.5 PROCESO DE CALIBRACIÓN GEOMÉTRICA**

Anteriormente se obtuvo un modelo matemático general, ecuación 1.10, que se aproxima a la representación de una cámara real, ya que tiene en cuenta las distorsiones geométricas que esta presenta. Con este modelo se puede encontrar la ubicación de los píxeles en el chip conociendo puntos del mundo real y referenciados a un marco arbitrario, esta ecuación posee varias incógnitas. Las incógnitas que representan el comportamiento interior de la cámara, como distancia focal  $f$  y los coeficientes de distorsión se denominan parámetros intrínsecos, mientras que las incógnitas que determinan la posición de la cámara respecto de un sistema de coordenadas del mundo, como la matriz de rotación y el vector de traslación, se denominan parámetros extrínsecos.

Al proceso de obtención de los parámetros intrínsecos y extrínsecos es a lo que se denomina calibración de cámara. Existen variedad de métodos para hacer calibración de cámara, pero en este trabajo se estudiará la calibración basada en el método de Zhang [11] para encontrar las longitudes focales y en el método de Brown [12] para resolver los parámetros de la distorsión.

### **1.5.1 Métodos de calibración**

A continuación se presenta el mecanismo de los dos métodos utilizados para la obtención de los parámetros intrínsecos, método de Zhang, y para determinar los coeficientes de distorsión geométrica, método de Brown.

#### **1.5.1.1 Método de Zhang**

Zhang [11] propone una técnica de calibración basada en la observación de una plantilla plana desde varias posiciones. La ventaja de este método de calibración es que permite



obtener los parámetros de la cámara fácilmente a partir de una plantilla plana en la cual no es necesario conocer las posiciones de los puntos de interés, ni tampoco es necesario conocer las posiciones de la cámara desde donde se han tomado las imágenes de la misma.

El modelo de la cámara que utiliza es el expresado en la ecuación 1.6, la cual se puede reescribir según la ecuación 1.16:

$$q = sHQ \quad (1.16)$$

donde:

$Q = [X \ Y \ Z \ 1]^T$ , aquí  $X = X_w, Y = Y_w, Z = Z_w$  y representan las coordenadas del mundo.

$q = [x \ y \ 1]^T$ , aquí  $x = x_f, y = y_f$  y representan las coordenadas en el plano Imagen y están dadas en pixeles.

s es un factor de escala arbitrario.

$$H = \begin{bmatrix} K_x f & 0 & C_x & 0 \\ 0 & K_y f & C_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & C_x \\ 0 & f_y & C_y \\ 0 & 0 & 1 \end{bmatrix} [r_1 \ r_2 \ r_3 \ t]$$

donde  $f_x = K_x f, f_y = K_y f$

Pero el interés en la práctica no es todo el espacio Q, si no Q' que se define solo en el plano que se está viendo. Esto permite hacer una breve simplificación. Sin pérdida de generalidad, se puede elegir el plano objeto de modo que  $Z = 0$ . Entonces la ecuación 1.16 queda según 1.17:

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = s \begin{bmatrix} f_x & 0 & C_x \\ 0 & f_y & C_y \\ 0 & 0 & 1 \end{bmatrix} [r_1 \ r_2 \ r_3 \ t] \begin{bmatrix} X \\ Y \\ 0 \\ 1 \end{bmatrix} = s \begin{bmatrix} f_x & 0 & C_x \\ 0 & f_y & C_y \\ 0 & 0 & 1 \end{bmatrix} [r_1 \ r_2 \ t] \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} \quad (1.17)$$

Por lo que  $q = HQ'$

Ahora H queda representada por la expresión 1.18:

$$H = s \begin{bmatrix} f_x & 0 & C_x \\ 0 & f_y & C_y \\ 0 & 0 & 1 \end{bmatrix} [r_1 \ r_2 \ t] \quad (1.18)$$

H representa la matriz de homografía, que relaciona las coordenadas de los puntos de la plantilla plana del escenario con sus correspondientes en la imagen.

Si ahora se hace  $M = \begin{bmatrix} f_x & 0 & C_x \\ 0 & f_y & C_y \\ 0 & 0 & 1 \end{bmatrix}$ , y se escribe  $H = [h_1 \ h_2 \ h_3]$ , donde cada h es un

vector columna de 3 por 1, se obtiene la ecuación 1.19:

$$H = [h_1 \ h_2 \ h_3] = sM[r_1 \ r_2 \ t] \quad (1.19)$$

Entonces, igualando términos correspondientes se llega a la ecuación 1.20.

$$h_1 = sMr_1 \text{ o } r_1 = \lambda M^{-1}h_1$$

$$h_2 = sMr_2 \text{ o } r_2 = \lambda M^{-1}h_2 \quad (1.20)$$

$$h_3 = sMr_3 \text{ o } r_3 = \lambda M^{-1}h_3$$

donde  $\lambda = \frac{1}{s}$

Los vectores  $r_1$  y  $r_2$  son ortonormales, es decir son ortogonales y la norma de cada uno de ellos es 1. Esto implica que el producto punto entre estos vectores de rotación es cero, y se expresa en la ecuación 1.21.

$$r_1^T r_2 = 0 \quad (1.21)$$

Para cualquier par de vectores  $a$  y  $b$  se tiene  $(ab)^T = a^T b^T$ , entonces  $r_1^T = (\lambda M^{-1} h_1)^T = \lambda M^{-T} h_1^T$  por lo que la restricción de 1.21 se transforma en 1.22.

$$h_1^T M^{-T} M^{-1} h_2 = 0 \quad (1.22)$$

donde  $M^{-T}$  es la abreviatura de  $(M^{-1})^T$ .

También se tiene que las magnitudes de  $r_1$  y  $r_2$  son iguales, ecuación 1.23.

$$||r_1|| = ||r_2|| \quad \text{O} \quad r_1^T r_1 = r_2^T r_2 \quad (1.23)$$

Sustituyendo la ecuación 1.20 en 1.21, se obtiene 1.24.

$$h_1^T M^{-T} M^{-1} h_1 = h_2^T M^{-T} M^{-1} h_2 \quad (1.24)$$

Para facilitar las cosas, se denota  $B = M^{-T} M^{-1}$  como en la ecuación 1.25.

$$B = M^{-T} M^{-1} = \begin{bmatrix} B_{11} & B_{12} & B_{13} \\ B_{21} & B_{22} & B_{23} \\ B_{31} & B_{32} & B_{33} \end{bmatrix} \quad (1.25)$$

Al resolver B, se obtiene la expresión 1.26.

$$B = \begin{bmatrix} \frac{1}{f_x^2} & 0 & \frac{-c_x}{f_x^2} \\ 0 & \frac{1}{f_y^2} & \frac{-c_y}{f_y^2} \\ \frac{-c_x}{f_x^2} & \frac{-c_y}{f_y^2} & \frac{-c_x^2}{f_x^2} + \frac{-c_y^2}{f_y^2} + 1 \end{bmatrix} \quad (1.26)$$

Analizando las restricciones 1.22 y 1.23 se observa que tienen la forma general  $h_i^T B h_j$  y si se tiene en cuenta que B es una matriz simétrica, es decir  $B = B^T$  ecuación 1.26, por lo que se puede expresar esta matriz como un vector de seis dimensiones, entonces se obtiene 1.27.

$$h_i^T B h_j = v_{ij}^T b = \begin{bmatrix} h_{i1} h_{j1} \\ h_{i1} h_{j2} + h_{i2} h_{j1} \\ h_{i2} h_{j2} \\ h_{i3} h_{j1} + h_{i1} h_{j3} \\ h_{i3} h_{j2} + h_{i2} h_{j3} \\ h_{i3} h_{j3} \end{bmatrix}^T \begin{bmatrix} B_{11} \\ B_{12} \\ B_{22} \\ B_{13} \\ B_{23} \\ B_{33} \end{bmatrix} \quad (1.27)$$

Teniendo en cuenta ahora la notación  $v_{ij}^T$ , las restricciones 1.22 y 1.24 se pueden escribir simultáneamente como en 1.28.

$$\begin{bmatrix} v_{12}^T \\ (v_{11} - v_{22})^T \end{bmatrix} b = 0 \quad (1.28)$$

En la práctica se recopila información de K imágenes de tableros de ajedrez, formando un conjunto de ecuaciones gobernadas por la ecuación 1.29.

$$Vb = 0 \quad (1.29)$$

donde V es una matriz de  $2K \times 6$ , con esta ecuación se pueden resolver los parámetros  $b = [B_{11}, B_{12}, B_{22}, B_{13}, B_{23}, B_{33}]^T$ , con estos valores se recurre a la ecuación 1.26 y finalmente se pueden obtener las características intrínsecas de la cámara según las ecuaciones 1.30 y 1.31:

$$\begin{aligned} f_x &= \sqrt{\lambda/B_{11}} \\ f_y &= \sqrt{\lambda B_{11}/(B_{11}B_{22} - B_{12}^2)} \\ C_x &= -B_{13}f_x^2/\lambda \\ C_y &= (B_{12}B_{13} - B_{11}B_{23})/(B_{11}B_{22} - B_{12}^2) \end{aligned} \quad (1.30)$$

$$\text{donde } \lambda = B_{33} - \left( B_{13}^2 + C_y(B_{12}B_{13} - B_{11}B_{23}) \right) / B_{11}$$

$$\begin{aligned}
r_1 &= \lambda M^{-1} h_1 \\
r_2 &= \lambda M^{-1} h_2 \\
r_3 &= r_1 \times r_2 \\
t &= \lambda M^{-1} h_3
\end{aligned}
\tag{1.31}$$

### 1.5.1.2 Método de Brown

Para encontrar los coeficientes de distorsión se usan los parámetros intrínsecos obtenidos anteriormente, con estos se genera un conjunto de ecuaciones que representan las proyecciones de los puntos sin distorsión.

Sea  $(X_p, Y_p)$  la ubicación de los puntos en la imagen si la cámara fuera perfecta, ecuación 1.32:

$$\begin{bmatrix} X_p \\ Y_p \end{bmatrix} = \begin{bmatrix} f_x X_w / Z_w + C_x \\ f_y Y_w / Z_w + C_y \end{bmatrix}
\tag{1.32}$$

Sean  $(X_d, Y_d)$  los mismos puntos pero con distorsión [12], si ahora se utilizan los parámetros sin distorsión obtenidos en la calibración, se obtiene la ecuación 1.33.

$$\begin{bmatrix} X_p \\ Y_p \end{bmatrix} = (1 + K_1 r^2 + K_2 r^4 + K_3 r^6) \begin{bmatrix} X_d \\ Y_d \end{bmatrix} + \begin{bmatrix} 2P_1 X_d Y_d + P_2 (r^2 + 2X_d^2) \\ P_1 (r^2 + 2Y_d^2) + 2P_2 X_d Y_d \end{bmatrix}
\tag{1.33}$$

Finalmente se recoge una lista de estas ecuaciones correspondientes a un conjunto de puntos en la imagen y se estiman los parámetros de distorsión, con estos datos se estiman nuevamente los parámetros intrínsecos y los parámetros extrínsecos.

---

## Capítulo 2

# TÉCNICA DE VISIÓN ESTEREOSCÓPICA

---

### 2.1 INTRODUCCION

En la actualidad existen diversos métodos para extraer la información espacial de una escena (ultrasonidos, infrarrojos, luz láser estructurada, etc.) [13]. Las técnicas de visión artificial son ampliamente utilizadas, sin embargo la mayoría de trabajos se han implementado bajo imágenes bidimensionales, suponiendo una fuerte restricción en el amplio campo de posibilidades de la visión de máquina. Para dar solución a esta limitación, se han desarrollado durante la última década estructuras que permiten obtener la información tridimensional de una escena, siendo una de las más utilizadas la constituida por un par estereoscópico de cámaras a semejanza de la visión humana.

El objetivo de la Visión Estéreo es resolver dos problemas el de la correspondencia y el de la reconstrucción 3D. El problema de correspondencia consiste en decidir para un punto del plano imagen de una de las cámaras, izquierda o derecha, qué punto en el plano imagen de la otra cámara es correspondiente (son imágenes del mismo punto físico). El segundo problema es el problema de reconstrucción que trata de obtener, dados dos puntos correspondientes en ambos planos imagen, las coordenadas 3D del punto en el espacio respecto a un sistema de coordenadas del mundo [14].

En este capítulo se estudian las relaciones algebraicas y geométricas que existen cuando se toman dos vistas de una escena (geometría bifocal). En la sección 2.2 se describe detalladamente la geometría que gobierna la visión estereo. En la sección 2.3 se consideran los principales procesos de pre-procesamiento utilizados antes de someter las imágenes a los algoritmos estereo. En la sección 2.4 se presentan los métodos de

correspondencia estéreo más utilizados. Finalmente en la sección 2.5 se muestra el proceso de determinación de la profundidad y la reconstrucción 3D de la escena.

## 2.2 GEOMETRIA ESTEREOSCÓPICA

El sistema estéreo básico consta de dos cámaras *pin-hole* idénticas y sus respectivos sistemas de coordenadas que difieren únicamente en la posición de sus orígenes. Los planos imagen izquierdo y derecho son coplanares y están representados por  $I_l$  y  $I_r$  respectivamente, los centros de proyección son  $C_l$  y  $C_r$ , los ejes ópticos son paralelos y por ende el punto de intersección de estos ejes esta en el infinito [15]. Si ahora se quiere obtener la posición de un punto P, tal como se muestra en la Fig. 2.1, a partir de sus proyecciones  $p_l = (x_l, y_l)$  y  $p_r = (x_r, y_r)$  y teniendo en cuenta la semejanza de triángulos se obtiene la ecuación 2.1.

$$\frac{B+x_l-x_r}{z-f} = \frac{B}{z} \quad (2.1)$$

donde  $B$  es la distancia entre los centros de proyección  $C_l$  y  $C_r$ , y se denomina línea base del sistema estéreo,  $x_l$  y  $x_r$  son las coordenadas de  $p_l$  y  $p_r$  con respecto a los puntos principales  $C_l$  y  $C_r$ ,  $f$  es la longitud focal y  $Z$  la distancia entre el punto P y la línea base.

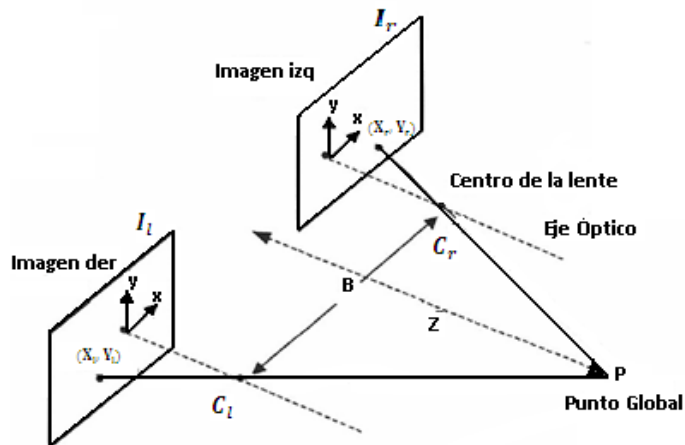


Fig. 2.1. Modelo básico de visión estéreo [15].

A partir de la ecuación 2.1 se obtiene 2.2:

$$Z = f \frac{B}{d} \quad (2.2)$$

donde  $d = x_l - x_r$  se denomina disparidad y mide la diferencia entre los correspondientes puntos proyectados en ambos planos imagen. De la ecuación 2.2 se deduce que la profundidad es inversamente proporcional a la disparidad.

Si ahora se tiene en cuenta un sistema de visión estéreo real, la configuración no será exactamente como se mostró en la Fig. 2.1, los ejes ópticos de las cámaras no estarán paralelos y esto conllevará a errores grandes en los cálculos de la disparidad. Por lo tanto se requiere de una alineación matemática. En la Fig. 2.2 se muestra la configuración estéreo real de dos cámaras y la alineación matemática que se requiere [16], donde R representa una matriz de rotación y T un vector de translación.

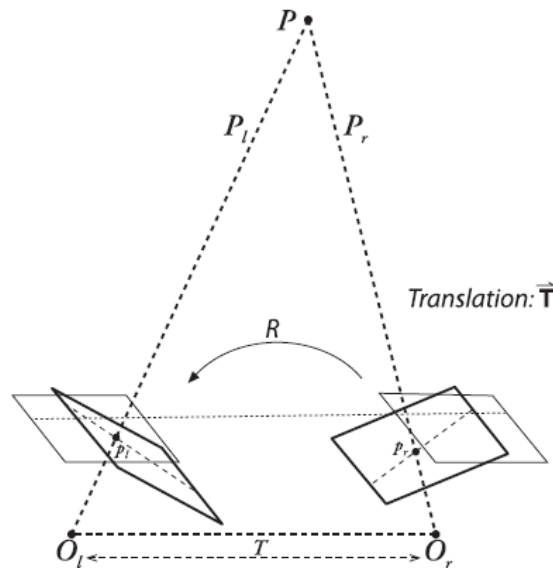


Fig. 2.2. Configuración estéreo real y alineación matemática necesaria para alineación. [16]



En la práctica es adecuado hacer el arreglo estéreo de cámaras lo más cercano al ideal mediante hardware, ya que si el sistema está muy desorientado la alineación matemática puede producir distorsiones extremas y reducir o eliminar el área de trabajo estéreo. También hay que tener en cuenta la sincronización de las cámaras ya que si no se capturan las imágenes simultáneamente se presentarán errores en el proceso de emparejamiento.

### 2.2.1 Cámaras no paralelas

En un sistema de visión estéreo real es posible que las cámaras no estén paralelas. El caso más crítico es que alguna de ellas rote alrededor del eje Y. En este caso la expresión 2.2 no es válida y los ejes ópticos se cruzan en el plano XY, de forma que se cortan en un punto  $(0, 0, Z_0)$ . Este punto es conocido como *fixation point*, y puede encontrarse detrás de las cámaras  $Z_0 < 0$  [17].

Si  $\theta$  es el ángulo de rotación, entonces la profundidad para disparidad cero se expresa como en la ecuación 2.3:

$$Z_0 = \frac{B}{\tan(\theta)} \quad (2.3)$$

Bajo pequeñas rotaciones, es posible asumir que el plano de la imagen en la cámara de la derecha permanece paralelo al plano de la imagen en la cámara izquierda, y por tanto al plano XY.

Para este caso la ecuación 2.2 se modifica como se muestra en 2.4:

$$Z = f \frac{B}{\left(d + \frac{fB}{Z_0}\right)} \quad (2.4)$$

### 2.2.2 Geometría Epipolar

El término epipolar viene del griego *epi* que significa encima, y *polos* cuyo significado es punto de atracción o uno de los dos puntos de una esfera que son interceptados por su eje de rotación [18].

En la Fig. 2.3a,  $M$  es un punto en el mundo real,  $m_1$  y  $m_2$  son las respectivas proyecciones en la imagen1 e imagen2. En el capítulo 1 se presentó la proyección de un punto 3D en un plano 2D mediante el modelo *pin-hole* de la cámara. En este caso los centros ópticos son  $C_1$  y  $C_2$ . La posición de  $m_1$  en la imagen1 no resuelve unívocamente la posición de  $M$ , ya que en el proceso de proyección se ha perdido la información de profundidad.

Sin embargo, se puede afirmar que  $M$  debe estar en el rayo que nace en el centro óptico  $C_1$  para formar  $m_1$ , es decir,  $M$  pertenece a la recta  $m_1C_1$ . Esta situación es mostrada en la Fig. 2.3b, donde varios puntos, incluido  $M$ , pertenecientes a la recta pueden ser los que forman el punto  $m_1$  en la primera imagen. Si, a partir de  $m_1$ , se desea conocer la ubicación de  $m_2$  es necesario proyectar en la imagen 2 los posibles puntos que pueden formar  $m_1$  Fig. 2.3c. Se observa que  $m_2$  es uno de estos puntos proyectados, sin embargo a partir de  $m_1$ , solamente, no se puede saber la ubicación exacta de  $m_2$ , sólo se puede afirmar que  $m_2$  pertenece a la proyección de la recta  $m_1C_1$  realizada por el segundo centro óptico  $C_2$  en la imagen 2. La proyección de esta recta se denomina *línea epipolar* y se puede apreciar en la Fig. 2.3d.

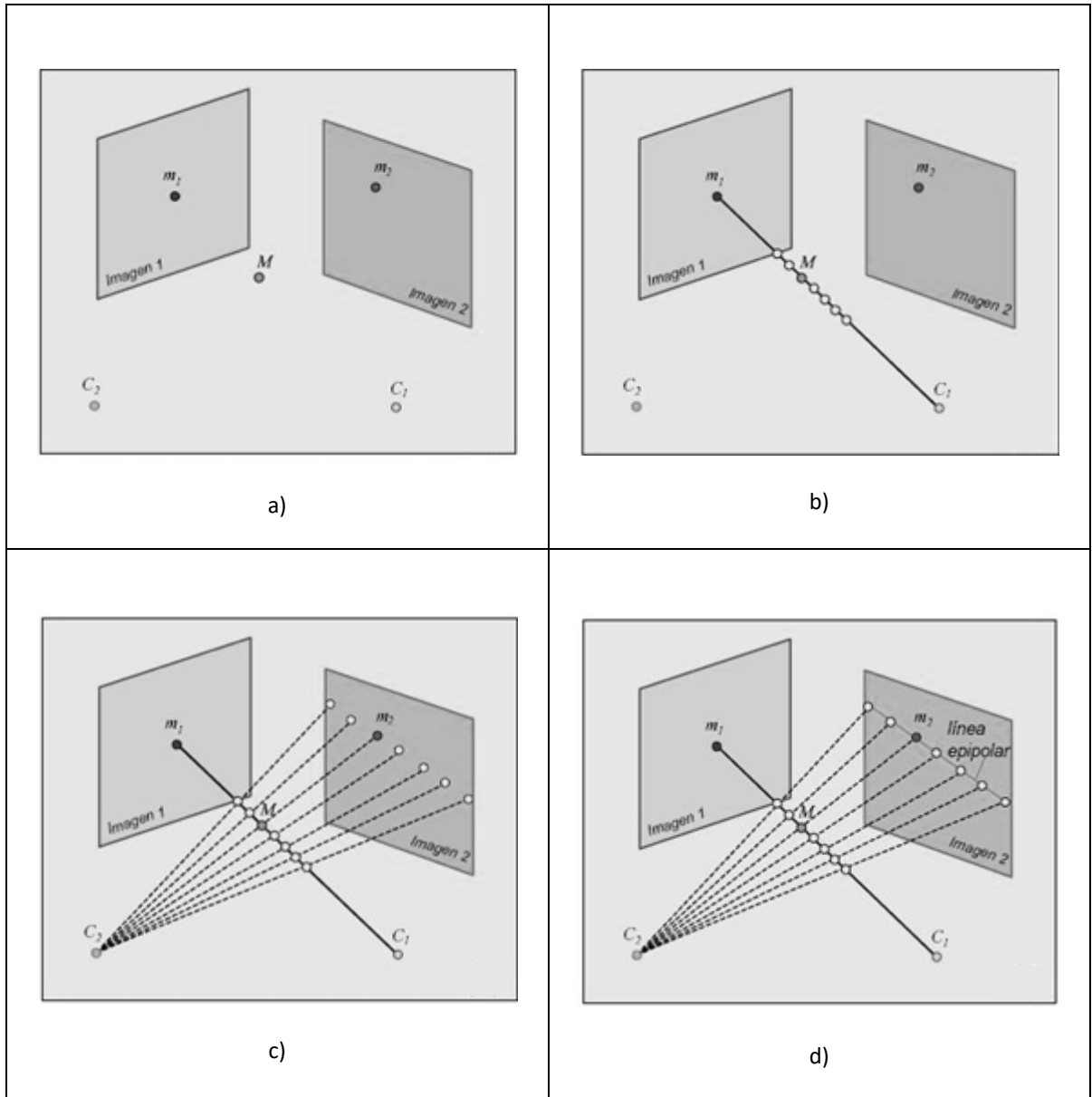


Fig. 2.3. Geometría epipolar bifocal [18].

La *restricción epipolar* señala que para que  $m_1$  y  $m_2$  sean puntos correspondientes, el punto  $m_2$  debe estar en la línea epipolar de  $m_1$ . Esto no quiere decir que todos los puntos en la línea epipolar de  $m_1$  son correspondientes a  $m_1$ , ya que, como bien se puede observar de la Fig. 2.3, sólo un punto en la imagen 2 es correspondiente a  $m_1$ , y en este caso es la proyección de  $M$  en la segunda imagen. La restricción epipolar es entonces una condición necesaria, más no suficiente. A pesar de que no sea una condición suficiente, es

de gran utilidad saber que el punto correspondiente a  $m_1$  en la segunda imagen esta sobre una línea y no está ubicado en cualquier parte de la imagen. Esto representa una reducción considerable en la dimensionalidad del problema de búsqueda de puntos correspondientes, ya que en vez de buscar en toda la imagen 2 (de dos dimensiones) se busca solo a lo largo de una línea (una dimensión).

Conociendo la geometría de la cámara es posible hacer una rectificación de las imágenes, proceso que consiste en transformar un conjunto de líneas epipolares genéricas en líneas paralelas, por lo que la búsqueda de correspondencia de puntos se convierte en una búsqueda a través de las líneas epipolares.

### **2.2.3 La matriz esencial E y la matriz fundamental F**

Existen relaciones geométricas entre puntos del plano imagen del par estereoscópico, y entre líneas epipolares correspondientes a las dos imágenes. Estas relaciones se pueden explicar en función de la matriz esencial E y la matriz fundamental F [16]. A continuación se presenta la descripción matemática de esta geometría.

#### **2.2.3.1 Matriz esencial E**

Dado un punto P en el mundo real, se desea conocer la relación entre  $p_l$  y  $p_r$  en los dos planos imagen, para hallar esto se utiliza la geometría epipolar. Antes que todo se escoge el sistema de referencia, para este caso se toma el sistema coordenado centrado en  $O_l$  en la cámara izquierda, Fig. 2.4, por lo tanto un punto P proyectado en la cámara derecha  $p_r$  se puede expresar en términos de  $p_l$  el punto proyectado en la cámara izquierda, ecuación 2.5.

$$p_r = R(p_l - T) \tag{2.5}$$

Si se recuerda que el conjunto de puntos  $x$  puede generar un plano que pasa por el punto  $a$  y tiene vector normal  $n$ , se cumple la ecuación 2.6.

$$(\mathbf{x} - \mathbf{a}) \cdot \mathbf{n} = 0 \quad (2.6)$$

Donde  $\mathbf{x}$  representa el vector posición de  $x$  y  $\mathbf{a}$  el vector posición de  $a$ .

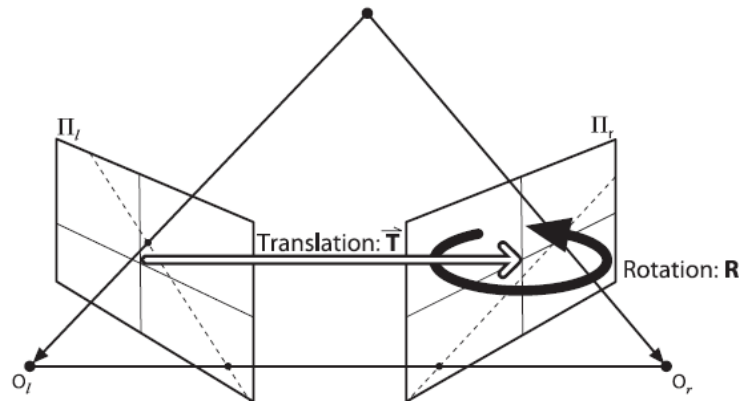


Fig. 2.4. Transformación geométrica de un punto en  $O_l$  a su correspondiente en  $O_r$ . [16].

Si ahora se relaciona la ecuación 2.6 con el plano epipolar, y teniendo en cuenta que este plano contiene a los vectores  $p_l$  y  $T$ , la ecuación del plano epipolar se puede expresar como en la ecuación 2.7.

$$(p_l - T)^T \cdot (T \times p_l) = 0 \quad (2.7)$$

donde  $(p_l \times T)$  es el vector normal al plano epipolar.

Si la ecuación 2.5 la multiplicamos por la inversa, se obtiene 2.8.

$$(p_l - T) = R^{-1}(p_r) \quad (2.8)$$

Haciendo esta sustitución en 2.7 y teniendo en cuenta que  $R^T = R^{-1}$ , se obtiene la ecuación 2.9.

$$(R^T p_r)^T (T \times p_l) = 0 \quad (2.9)$$

Si ahora se expande el producto cruz  $(T \times p_l)$ , se obtiene 2.10.

$$T \times p_l = S p_l \Rightarrow S = \begin{bmatrix} 0 & -T_z & T_y \\ T_z & 0 & -T_x \\ -T_y & T_x & 0 \end{bmatrix} \quad (2.10)$$

Entonces, la ecuación 2.9 se transforma en 2.11.

$$(p_r)^T R S p_l = 0 \quad (2.11)$$

Al producto  $R \times S$  se le conoce como la matriz esencial E, entonces 2.11 se puede reescribir como se muestra en 2.12.

$$(p_r)^T E p_l = 0 \quad (2.12)$$

La matriz E resulta ser de rango deficiente, porque es de dimensiones 3x3 pero sólo es de rango dos, es decir dos de sus columnas son linealmente independientes, en la práctica esto es la ecuación de una línea recta en el plano. Esta matriz contiene cinco parámetros, tres para la rotación y dos para la traslación en una escala no establecida.

La matriz esencial E, tiene además otras dos limitaciones, la primera muestra que el determinante de E es cero, porque es de rango deficiente, la segunda muestra que sus valores singulares distintos de cero son iguales, porque la matriz S es anti-simétrica y R es una matriz de rotación.

Hay que tener en cuenta que la matriz E no tiene ningún parámetro intrínseco, por lo que todos los puntos a los que se refiere dicha matriz están en coordenadas físicas, no en coordenadas en píxeles.

### 2.2.3.2 La matriz fundamental F

La matriz E tiene la información geométrica de una cámara respecto a la otra, pero no tiene la información de las propias cámaras. En la práctica es importante conocer las coordenadas de la matriz E en píxeles, por lo tanto para encontrar la relación entre un píxel en una imagen y la línea epipolar correspondiente en la otra imagen se debe tener en cuenta la información intrínseca de las dos cámaras. Recordando lo expuesto en el capítulo 1, se tiene la ecuación 2.13.

$$q = Mp \tag{2.13}$$

donde  $M$  Es la matriz de parámetros intrínsecos,  $p$  Es el punto en la imagen,  $q$  el punto en el espacio real. Multiplicando la ecuación 2.13 por la matriz inversa, se obtiene 2.14.

$$p = M^{-1}q \tag{2.14}$$

Si se usa 2.14, para los puntos  $p_r$  y  $p_l$ , y reemplazando en la ecuación 2.12, se obtiene la ecuación 2.15.

$$q_r^T (M_r^{-1})^T E M_l^{-1} q_l = 0 \tag{2.15}$$

Entonces se define a la matriz F como se muestra en la expresión 2.16.

$$F = (M_r^{-1})^T E M_l^{-1} \tag{2.16}$$

De modo que la ecuación 2.15 se transforma en 2.17.

$$q_r^T F q_l = 0 \quad (2.17)$$

Por lo tanto la matriz fundamental F es igual a la matriz esencial E, salvo que la matriz F opera con las coordenadas en pixeles, mientras que E opera con las coordenadas físicas. Por último hay que resaltar que la matriz F posee las mismas propiedades que la matriz E.

#### **2.2.4 Alineación *off-line***

Para que la expresión 2.2 sea válida y libre de errores, el par estereoscópico debe estar en su forma estándar, es decir, como lo muestra la Fig. 2.1. Para lograr esta disposición, debe realizarse una primera alineación *hardware off-line*. Esta alineación consiste en ubicar, físicamente, las cámaras con sus ejes ópticos paralelos. Para este propósito se diseñó un soporte que permite modificar la posición de una de las cámaras. Este procedimiento se detallará en el capítulo 3.

En el proceso de alineación *off-line*, existen dos procedimientos software, que son:

\_ Determinación de parámetros intrínsecos de las cámaras, indispensables para realizar la corrección con el sistema en línea.

\_ Determinación de la matriz esencial E y la matriz fundamental F, indispensables para realizar la alineación *hardware* con el sistema en línea.



## 2.2.5 Proceso de Rectificación

Con la calibración estéreo se conoce la relación geométrica de una cámara con respecto a la otra. Ahora el propósito es corregir las imágenes de tal manera que sean equivalentes a imágenes tomadas con dos cámaras con los ejes ópticos paralelos, como se muestra en la Fig. 2.1.

### 2.2.5.1 Algoritmo de Hartley

El algoritmo de Hartley intenta encontrar homografías con el mapa de epipolos en el infinito y reducir al mínimo la disparidad entre las dos imágenes estéreo. Para esto se calculan los parámetros intrínsecos de cada cámara, ya que la información intrínseca está implícitamente contenida en la proyección de los puntos del tablero de calibración. Por lo tanto, solo se calcula la matriz fundamental  $F$ , la cual se puede obtener de cualquier conjunto combinado de siete o más puntos entre los dos puntos de vista de la escena.

La ventaja del algoritmo de Hartley es que la calibración estéreo en línea, se puede realizar simplemente mediante la observación de los puntos de la escena. El inconveniente es que no se tiene información sobre la escala de la imagen, por ejemplo, si se usa un tablero de ajedrez para la generación de un punto en el centro, el sistema no sería capaz de saber si el tablero es grande y se encuentra lejos, o si es pequeño y se encuentra cerca. Tampoco aparecen explícitamente los parámetros intrínsecos de la cámara, sin los cuales las cámaras podrían tener distintas longitudes focales, diferentes centros de proyección y puntos principales diferentes. Como resultado de lo anterior se puede determinar la reconstrucción 3D de objetos solo con una transformación proyectiva. Esto significa que la proyección de objetos de diferentes escalas puede ser igual (los puntos tienen la mismas coordenadas en 2D a pesar de que los objetos 3D son diferentes) [19].

Suponiendo que se tiene la matriz fundamental  $F$  el algoritmo de *Hartley* procederá como se describe a continuación:

1. Se utiliza la matriz fundamental  $F$ , para calcular los dos epipolos a través de las relaciones:  $F e_l = 0$  y  $(e_r)^T F = 0$ , donde  $e_l$  y  $e_r$  son los epipolos izquierdo y derecho, respectivamente.

2. Se busca una primera homografía  $H_r$ , que seguirá de cerca el mapa de los epipolos derechos a los puntos 2D homogéneos en el infinito  $(1,0,0)^T$ . Ya que una homografía tiene siete constantes, se usan tres para hacer el mapeo hasta el infinito, se disponen de 4 grados de libertad para elegir  $H_r$ . Con estos cuatro grados de libertad la mayoría de las opciones  $H_r$  darán lugar a imágenes muy distorsionadas. Para encontrar una buena  $H_r$ , se elige un punto de la imagen donde se quiere que ocurra una distorsión mínima y solo se permite la rotación. La traslación no es permitida. Una opción razonable para este punto es el origen de la imagen y además se supondrá que el epipolo  $(e_r)^T = (f, 0, 1)$  se encuentra en el eje X. Teniendo en cuenta estas coordenadas, la matriz  $G$  que permite llevar los epipolos hasta el infinito queda como:

$$G = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -1/f & 0 & 1 \end{bmatrix}$$

3. Para un punto de interés seleccionado en la imagen de la derecha (se elige el origen), se calcula la transformación de traslación que llevará a ese punto al origen de la imagen (0 en este caso) y  $R$  la rotación que tomará el epipolo  $(e_r)^T = (f, 0, 1)$ . Entonces, la homografía será  $H_r = GRT$ .

4. Se busca una próxima homografía  $H_l$ , que envía el epipolo izquierda al infinito y se alinean las filas de las dos imágenes. El envío del epipolo izquierda hasta el infinito se hace fácilmente mediante el uso de hasta tres restricciones como en el paso 2. Para alinear las filas, sólo se tiene que utilizar el hecho de que la alineación de las filas minimiza la distancia total entre todos los puntos coincidentes entre las dos imágenes. Es decir, se consigue la  $H_l$  que minimice la disparidad total en los puntos coincidentes de izquierda a derecha,  $\sum_i d(H_l p_i^l, H_r p_i^r)$ . Estas dos homografías  $H_r$  y  $H_l$  definen la rectificación estéreo.

### 2.3 PREPROCESAMIENTO DE IMÁGENES

Una vez se ha rectificado las imágenes del par estereoscópico, es necesario hacer un pre-procesamiento con el propósito de preparar las imágenes para que sean tratadas por los algoritmos estéreo [20].

La razón de que un paso de pre-procesamiento sea a menudo necesario, es debido a que las imágenes, en general, contienen ruido de alta frecuencia producido por el procesamiento de imagen, y variaciones de baja frecuencia debido a diferentes características de la cámara, tales como diferencias en ganancia y distribuciones desiguales de intensidad. Estos componentes de frecuencia indeseables pueden ser filtrados usando transformadas de imágenes pasa bajas y pasa banda.

Otros ejemplos de pre-procesamiento incluyen extracción de elementos característicos de la imagen tales como bordes [21], esquinas, lugares de alto gradiente espacial, signo del laplaciano [22] y las transformaciones de rango.

### 2.3.1 Filtros pasa bajas

El objetivo de los filtros pasa bajo es suavizar la imagen, son útiles cuando se supone que la imagen tiene gran cantidad de ruido y se quiere eliminar. También pueden utilizarse para resaltar la información correspondiente a una determinada escala (tamaño de la matriz de filtrado). Existen varias posibilidades: filtro de media, filtro de media ponderada, filtro de mediana, etc. En la Fig. 2.5 se muestra el mecanismo de funcionamiento de los filtros de media, media ponderada y mediana. Además se muestra un *kernel* de convolución para un filtro gaussiano.

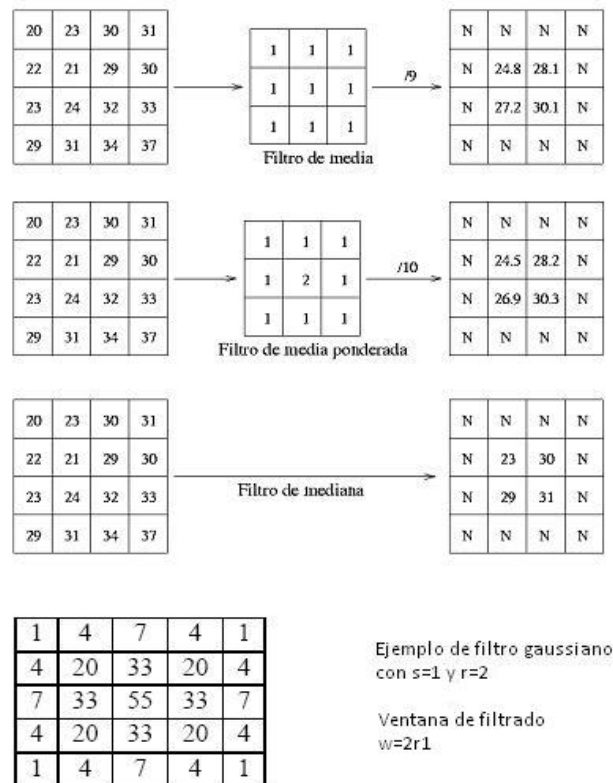


Fig. 2.5 Filtros pasa bajas, mecanismo de aplicación.

Los filtros adaptativos son considerablemente más complejos, ya que los coeficientes de ponderación se re-calculan para cada uno de los pixeles, en función del histograma de los

niveles de intensidad que aparecen en la ventana. Se han utilizado con gran éxito filtros adaptativos para eliminar el *speckle* de las imágenes de radar y para detectar, con un solo filtro, diferentes elementos.

Los filtros gaussianos simulan una distribución gaussiana bivalente. El valor máximo aparece en el pixel central y disminuye hacia los extremos tanto más rápido cuanto menor sea el parámetro de desviación típica. El resultado de aplicar el filtro gaussiano será un conjunto de valores entre 0 y 1. Para transformar la matriz a otra de números enteros se divide toda la matriz por el menor de los valores obtenidos.

### **2.3.2 Filtros pasa altas**

Su objetivo es resaltar las zonas de mayor variabilidad eliminando lo que sería la componente media, precisamente la que detectan los filtros de paso bajo. Por otra parte la respuesta de cada pixel está contaminada por la de los pixeles vecinos, ya que la radiación reflejada por un pixel se reparte hacia los pixeles vecinos. Los filtros pasa alto consiguen también eliminar en parte esta contaminación. Existen diversos métodos:

\_El método de la sustracción de la media, si se considera que un filtro pasa bajo sirve para resaltar componentes a gran escala eliminando la variabilidad local, si a la imagen original se le resta el resultado de pasarle un filtro pasa bajo se consigue resaltar esa variabilidad local.

\_El filtro basado en las derivadas, donde La derivada de una función  $y = f(x)$  es el incremento de  $f(x)$  para cada incremento infinitesimal de  $x$ . A continuación aparecen las ecuaciones de las derivadas respecto a  $x$  e  $y$ , las segundas derivadas respecto a  $x$  e  $y$ , la derivada compuesta y la derivada segunda compuesta denominada también Laplaciana, ver ecuación 2.16.

$$\Delta xp(i, j) = p(i, j) - p(i - 1, j)$$

$$\Delta yp(i, j) = p(i, j) - p(i, j - 1)$$

$$\Delta x_2p(i, j) = \Delta xp(i + 1, j) - \Delta xp(i, j) = p(i + 1, j) + p(i - 1, j) - 2p(i, j)$$

$$\Delta y_2p(i, j) = \Delta yp(i, j + 1) - \Delta yp(i, j) = p(i, j + 1) + p(i, j - 1) - 2p(i, j) \quad (2.16)$$

$$\Delta xyp(i, j) = \sqrt{\Delta xp(i, j)^2 + \Delta yp(i, j)^2}$$

$$\Delta xy_2p(i, j) = [p(i + 1, j) + p(i - 1, j) + p(i, j + 1) + p(i, j - 1)] - 4p(i, j)$$

En la Fig. 2.6 se muestran algunos *kernel* para aplicar los filtros pasa alto.

0	-1	0
-1	5	-1
0	-1	0

a)

1	1	1
1	-2	1
-1	-1	-1

b)

-1	1	1
-1	-2	1
-1	1	1

c)

-1.0	0	1.0	-1.0	-2.0	-1.0
-2.0	0	2.0	0	0	0
-1.0	0	1.0	1	2	1

d)

Fig. 2.6 Ejemplos de *kernel* pasa altas, a) Filtro menos Laplaciano. b) Filtro norte y este. c) Filtro Sobel

## 2.4 CORRESPONDENCIA ESTÉREO

En los últimos años se han desarrollado diversos métodos para solucionar el problema de correspondencia. Una descripción de estos métodos se presenta en [23], [24], [25]. Las dos principales clases son los métodos basados en áreas o regiones y los métodos basados

en características. En este trabajo de investigación se exploran los dos métodos mencionados.

Según [26] se pueden identificar 4 aspectos problemáticos en el proceso de búsqueda de correspondencia.

- Variación fotométrica: La intensidad de luz que llega a la cámara varia con respecto a su posición, de esta forma la luz que recoge la cámara izquierda será diferente a la que recoge la cámara derecha, tanto en su intensidad como en su distribución.
- Oclusiones: Debido a que cada cámara tiene diferente perspectiva de la escena, existen zonas que mientras son visibles para una de las cámaras para la otra están ocultas, de esta forma estos puntos de oclusión no se logran emparejar.
- Textura repetitiva: la presencia de texturas repetitivas en la escena, provocan la existencia de múltiples posibles emparejamientos, violando el principio de unicidad.
- Falta de textura: Los algoritmos de correspondencia tienen mayor probabilidad de error en zonas de la escena que carezcan de texturas.

### **2.4.1 Suma de diferencias absolutas SAD**

Este método es ampliamente conocido en la búsqueda de correspondencias. Este algoritmo es exitoso encontrando puntos correspondientes de dos imágenes en ambientes de alta textura.

El propósito del algoritmo es encontrar para cada píxel de una imagen su correspondiente en la otra imagen, ya que las imágenes de entrada se encuentran rectificadas, un píxel con coordenadas en la imagen  $(x, y)$  se encontrará sobre la misma línea y a una distancia o

disparidad horizontal  $d$  en el punto  $(x+d, y)$  de la otra imagen. Para emparejar los píxeles, se utiliza una ventana centrada en el punto de interés en  $(x, y)$  y otra ventana de igual dimensión en la segunda imagen cuyo centro será desplazado sobre la línea de exploración, línea que comparten las dos imágenes sobre la cual se busca la correspondencia. Para escoger el punto en la imagen recorrida que produzca el emparejamiento, se utiliza la sumatoria de diferencias absolutas SAD, según la ecuación 2.17. Cada píxel de la primera ventana se resta de su correspondiente en la otra y se toma el valor absoluto de esa diferencia. La suma de todos estos valores determina la similitud o costo de apareamiento entre las dos ventanas. Entre menor sea el valor SAD mayor será la similitud entre las dos ventanas.

$$SAD = \sum_{(x,y)} |I_1(x, y) - I_2(x + d, y)| \quad (2.17)$$

En la Fig. 2.7 se muestra la búsqueda de correspondencia sobre la línea epipolar. El círculo azul en la imagen izquierda representa el píxel que se desea emparejar. La región entre las líneas rojas paralelas representa la línea epipolar. Sobre cada punto de la línea epipolar de la imagen derecha se aplica el algoritmo SAD. De esta forma se obtiene un valor de similitud para cada punto sobre la línea epipolar como se muestra en la Fig. 2.8.

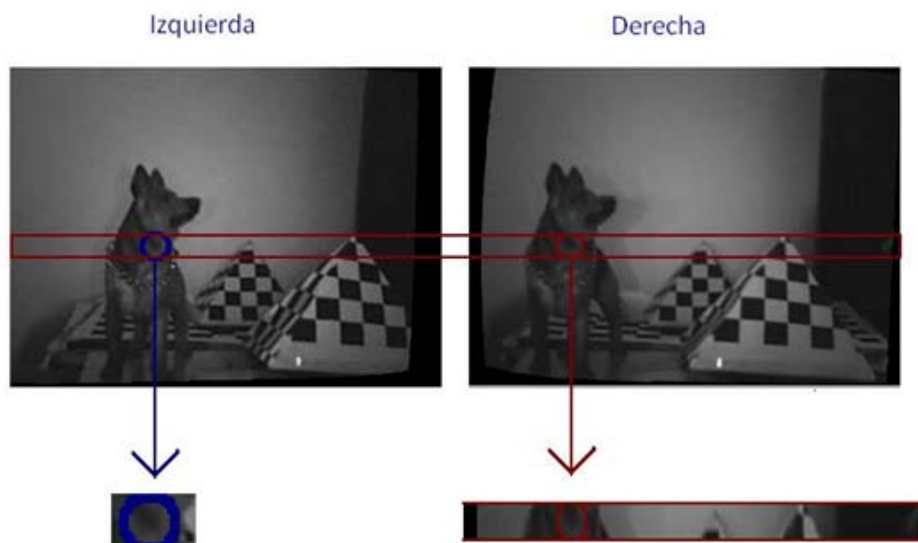


Fig. 2.7 Búsqueda de correspondencia, fuente propia.



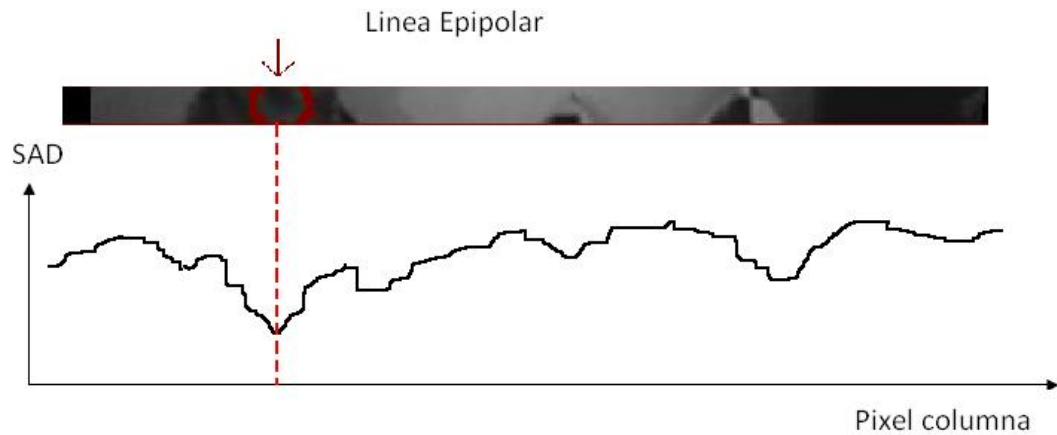


Fig. 2.8 Recorrido sobre la línea epipolar. Fuente propia.

## 2.4.2 Flujo óptico

El flujo óptico se define como el aparente movimiento de objetos en una secuencia de imágenes debido a cambios en la intensidad de los píxeles, aunque la intensidad de los píxeles puede variar por causa de otras variables que no son el movimiento de objetos, como cambio en la iluminación, etc. Se supone, en principio, estas variables constantes, un punto de la imagen conserva su intensidad al moverse, así que se considera que los cambios de intensidad en la imagen son debidos al movimiento de objetos en la escena, y entonces se considera el flujo óptico como el análisis de la velocidad de ese punto.

El flujo óptico deduce una ecuación que indica la componente, en dirección normal a los contornos de la velocidad. Para hallar la segunda componente de la velocidad deben hacerse suposiciones adicionales que se consideraran en el método de *Lucas-Kanade* [27].

Sean  $I(x, y, t)$  la intensidad del punto  $(x, y)$  de la imagen en el instante de tiempo  $t$ . Sean  $u(x, y)$  y  $v(x, y)$  las componentes en  $x$  e  $y$  del flujo óptico en ese punto. Después de un

pequeño intervalo de tiempo  $dt$ , este punto estará en  $(x + u \times dt, y + v \times dt)$ , conservando su intensidad como en 2.18.

$$I(x, y, t) = I(x + u \times dt, y + v \times dt) \quad (2.18)$$

Considerando una variación suave de la intensidad, se puede desarrollar el lado izquierdo de la ecuación en serie de Taylor, considerando además  $u = dx/dt$   $v = dy/dt$ ,  $I_x = dI/dx$ ,  $I_y = dI/dy$  e  $I_t = dI/dt$ , se obtiene finalmente la ecuación 2.19 de contención del flujo óptico.

$$I_x \times u + I_y \times v + I_t = 0 \quad (2.19)$$

Esta ecuación identifica una línea, perpendicular al gradiente de la intensidad del punto dado, sobre la cual estará su velocidad. Esta línea se denomina línea de condición. Por tanto, con esta ecuación, se conoce solamente una componente de la velocidad.

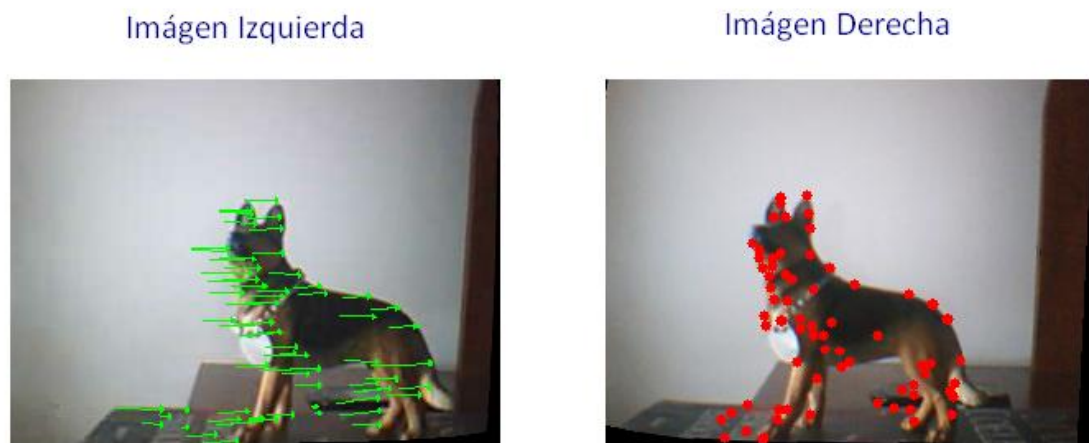
Como se ha visto, con esta ecuación sólo se obtiene información sobre la componente de la velocidad en la dirección del gradiente de intensidad, esto es, perpendicularmente a los contornos. No se podrá determinar la componente del flujo óptico en la dirección del contorno, inconveniente denominado problema de la apertura.

El problema de la apertura surge de la imposibilidad de asignar un valor unívoco al desplazamiento de un punto debido a que la región analizada es demasiado pequeña. Este es un caso particular de un problema más general conocido como el problema de la correspondencia. En la resolución del problema de la apertura se recurre a la aplicación de criterios de vecindad. Con ello se pretende mejorar la estación del campo en regiones

ambiguas mediante la observación del campo presente en regiones próximas que presenten mejores características. Existen diferentes tipos de algoritmos, entre los que destacan los algoritmos de Horn y Shunk [28], Lucas y Kanade [27].

Si bien se ha descrito el método de flujo óptico para imágenes en secuencia, se puede aplicar también como método de correspondencia estéreo. Las imágenes izquierda y derecha de un arreglo estereoscópico bifocal muestran una imagen desplazada respecto a la otra, esto es equivalente a tener un par de imágenes de una misma cámara en dos instantes consecutivos de tiempo, es por esto que el método de flujo óptico tiene aplicación en visión estéreo como método de correspondencia.

En la Fig. 2.9 se muestra un ejemplo del uso del flujo óptico para realizar la correspondencia estéreo. Los puntos rojos en la imagen derecha representan los pixeles que se desean emparejar. Las líneas verdes en la imagen izquierda son líneas de flujo óptico que representan el desplazamiento sufrido por los puntos de interés.



**Fig. 2.9 Flujo óptico aplicado al problema de búsqueda de correspondencia. Fuente propia.**

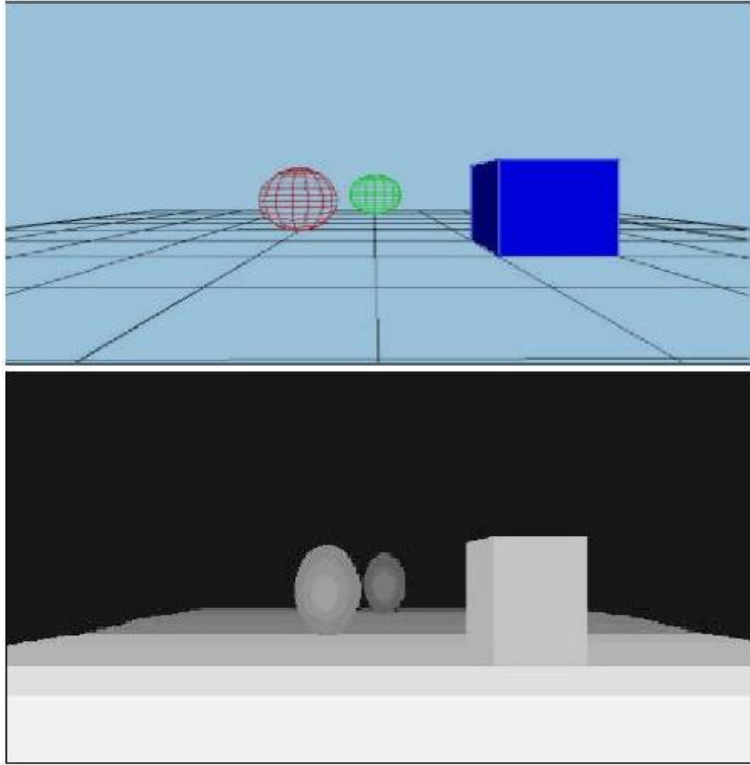
### **2.4.2.1 Método de Lucas-Kanade**

El algoritmo de *Lucas-Kanade* [27] permite la extracción de las dos componentes de la velocidad, utilizando una función ventana para tomar los valores de los gradientes espaciales y temporal de un entorno de vecindad del punto llevado a estudio. Propone una estimación del flujo óptico por bloques, es decir, estima el vector de movimiento para el bloque de píxeles. En la Fig. 2.9 se muestra el resultado de la búsqueda de correspondencias por flujo óptico usando el algoritmo de Lucas-Kanade.

## **2.5 DETERMINACIÓN DE PROFUNDIDAD Y RECONSTRUCCIÓN 3D**

Una vez culminado el proceso de búsqueda de correspondencias, se debe encontrar el mapa de disparidad, con el cual se establece la profundidad o coordenada Z. Las coordenadas X e Y se encuentran a partir de Z. De esta forma la reconstrucción 3D utilizando estereoscopía se entiende como el proceso de establecer coordenadas tridimensionales a partir de las proyecciones 2D de los emparejamientos y su visualización en un escenario tridimensional.

En la Fig. 2.10 se presenta un mapa de disparidad de una escena artificial. La imagen inferior representa la profundidad de la escena, en donde los niveles de gris más altos representan menor profundidad. En un caso real, se obtiene un mapa de disparidad en el cual existen regiones para las cuales los algoritmos no pueden encontrar asociado un valor de disparidad.



**Fig. 2.10. Superior imagen en simulación, inferior mapa de disparidad [29]**

Si se considera que el sistema estereoscópico está posicionado sobre el eje X en el sistema de referencia, con los planos imagen sobre el plano XY, entonces los ejes ópticos son paralelos al eje Z, como se muestra en la Fig. 2.11. La triangulación determina un punto sobre un objeto determinando el triángulo formado entre el punto observado P y los ejes ópticos de las cámaras. La información de profundidad del punto P está codificada en la diferencia de posición (disparidad) en los dos planos imagen.

Si se sitúa el origen de coordenadas en el centro de la lente de la cámara izquierda, según la Fig. 2.11, se puede encontrar el conjunto de coordenadas (X,Y,Z) utilizando la idea de triangulación.

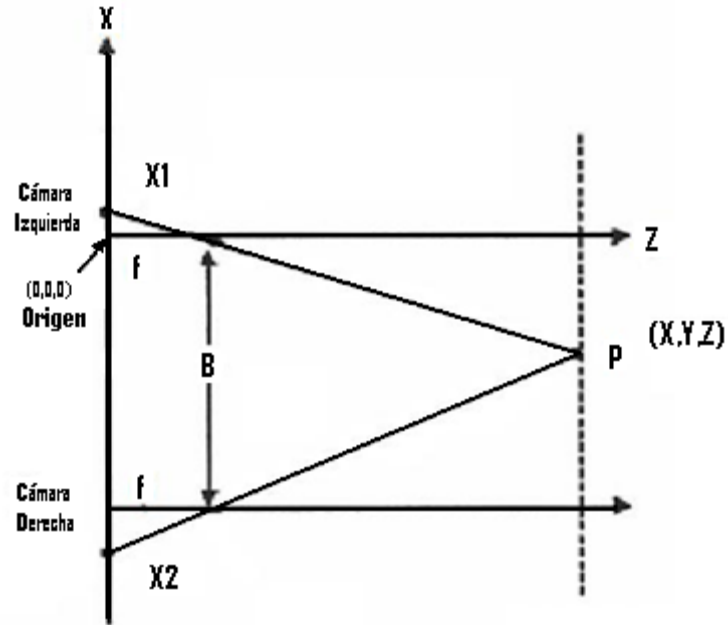


Fig.2.11. marco de referencia utilizado en la geometría de triangulación [15].

Las coordenadas reales sobre el eje X del punto de interés se proyectan en las dos imágenes según las ecuaciones 2.20. Donde el subíndice 1 equivale a la cámara izquierda y el subíndice 2 a la cámara derecha.

$$X_1 = \frac{X_{f1} \times Z}{f_x} \tag{2.20}$$

$$X_2 = \frac{X_{f2} \times Z}{f_x}$$

donde  $X_{f1}$  y  $X_{f2}$  están dados en pixeles, y  $f_x$  es el producto de la distancia focal  $f$  en mm y el factor de conversión  $S_x$  dado en pixeles/mm. De la Fig. 2.12 la coordenada Z del punto P es la misma en los sistemas de coordenadas de ambas cámaras, por lo que se tiene que:

$$X_2 = X_1 + B \tag{2.21}$$

$$Z_2 = Z_1 = Z \tag{2.22}$$

donde B es la distancia de la línea base. Reemplazando 2.20 en 2.21 se obtiene la ecuación 2.23:

$$X_1 + B = \frac{X_{f2} \times Z}{f_x} \quad (2.23)$$

Finalmente restando la primera ecuación 2.20 de 2.23 se obtiene la coordenada de profundidad Z, dada por la ecuación 2.24.

$$Z = \frac{f_x \times B}{X_{f2} - X_{f1}} \quad (2.24)$$

Determinado el valor de la coordenada Z, los valores para X e Y se obtienen de las ecuaciones 2.25.

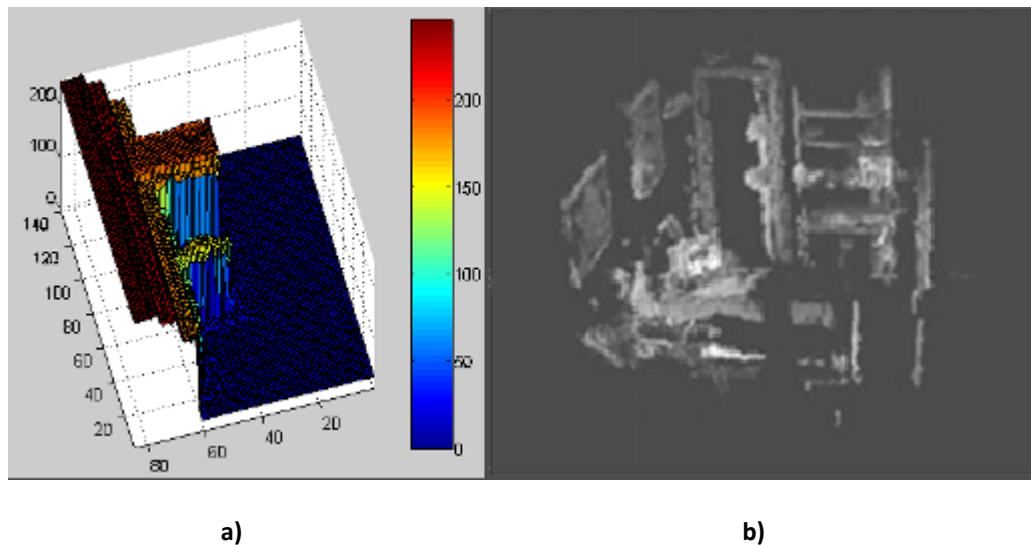
$$X = \frac{X_{f1} \times Z}{f_x} \quad (2.25)$$

$$Y = \frac{Y_{f1} \times Z}{f_y}$$

Las ecuaciones 2.24 y 2.25 se pueden expresar en forma matricial, donde dado un vector con componentes  $(X_{f1}, Y_{f1}, d)$  existe una matriz 3x3 tal que al multiplicarla con este vector genera un nuevo vector cuyas componentes son  $(X, Y, Z)$  que son las coordenadas del mundo real, como lo muestra la ecuación 2.26.

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \begin{pmatrix} \frac{B}{d} & 0 & -c_x \times \frac{B}{d} \\ 0 & \frac{f_x}{f_y} \times \frac{B}{d} & -c_y \times \frac{f_x}{f_y} \times \frac{B}{d} \\ 0 & 0 & f_x \times \frac{B}{d} \end{pmatrix} \begin{pmatrix} X_{f1} \\ Y_{f1} \\ 1 \end{pmatrix} \quad (2.26)$$

De esta forma queda determinado el conjunto de puntos (X,Y,Z) con origen de coordenadas en el centro de la lente de la cámara derecha. En la Fig. 2.12 se muestran dos posibles representaciones del conjunto de puntos en 3D [29].



**Fig. 2.12** Dos posibles representaciones de los datos 3D. a) Mallas, b) Nube de puntos. [29]

En el presente trabajo se muestran los datos 3D en 5 representaciones: 1. Mapa de disparidad, 2. Mapa en pseudocolor, 3. Perfil de línea, 4. Nube de puntos, 5. Mallas tridimensionales.



---

## Capítulo 3

# IMPLEMENTACIÓN DEL SISTEMA DE VISIÓN ESTÉREO

---

### 3.1 INTRODUCCIÓN

Existen actualmente en el mercado diversos tipos de dispositivos de visión estereoscópica, algunos de los cuales están integrados con algoritmos utilizados para obtener mapas de disparidad [30]. Sin embargo, el costo de estos dispositivos es alto, y muchos están limitados a las funciones para las que fueron diseñados, por ejemplo con *baseline* fija. En este trabajo de investigación se realizó un sistema de visión estéreo de bajo costo, conformado por un dispositivo (hardware) y un conjunto de algoritmos ejecutados sobre una plataforma robótica en el entorno de Linux.

En este capítulo se presenta la implementación del sistema de visión estéreo. En la sección 3.2 se muestra el diseño de la plataforma física de visión y su implementación. En la sección 3.3 se presenta el mecanismo de funcionamiento de un conjunto de algoritmos de visión estéreo, se explora el funcionamiento del sistema en la plataforma robótica Player, se muestran los resultados de disparidad por regiones y por extracción de características y finalmente, se muestra la representación 3D de las escenas. En la sección 3.4 se realiza la caracterización del sistema de visión, se encuentra la relación experimental entre profundidad y disparidad, se determina el rango, la resolución y finalmente el tiempo de ejecución de los algoritmos.

## 3.2 SISTEMA ESTÉREO HARDWARE

La técnica basada en visión estereoscópica usada para obtener la coordenada de profundidad de una escena requiere una geometría específica del par estereoscópico, esta geometría tiene que ver con la alineación de los ejes ópticos de las cámaras que conforman el sistema estéreo, ver Fig. 2.1. En el proceso de rectificación de imágenes se logra realizar esta operación por medio software, sin embargo el resultado es óptimo si se realiza pre-alineación hardware de los planos imagen.

### 3.2.1 Requisito de Alineación de ejes ópticos

Si en el proceso de alineación una de las cámaras es estática, existen 3 grados de libertad en los que se puede mover la segunda cámara, para alinear sus planos imagen. Esto se cumple si las cámaras se encuentran a la misma altura. En la Fig. 3.1 las flechas indican el movimiento que debe realizarse poner a la misma altura los centros ópticos de las cámaras.

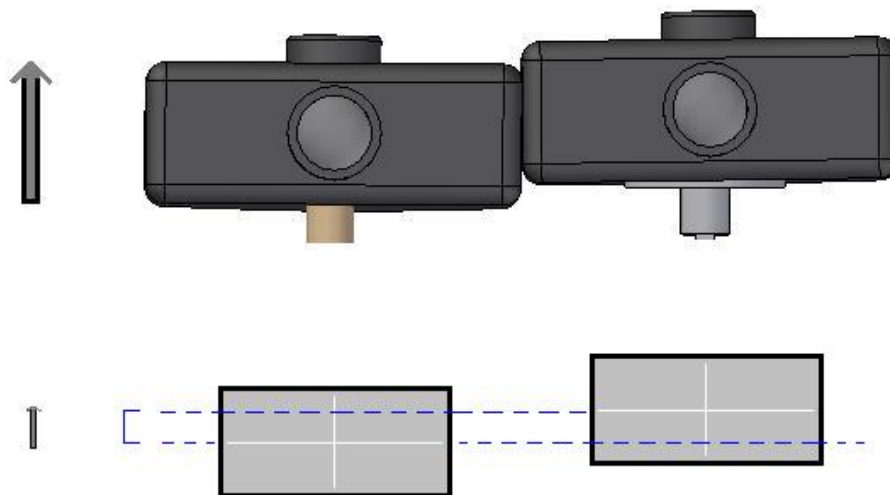
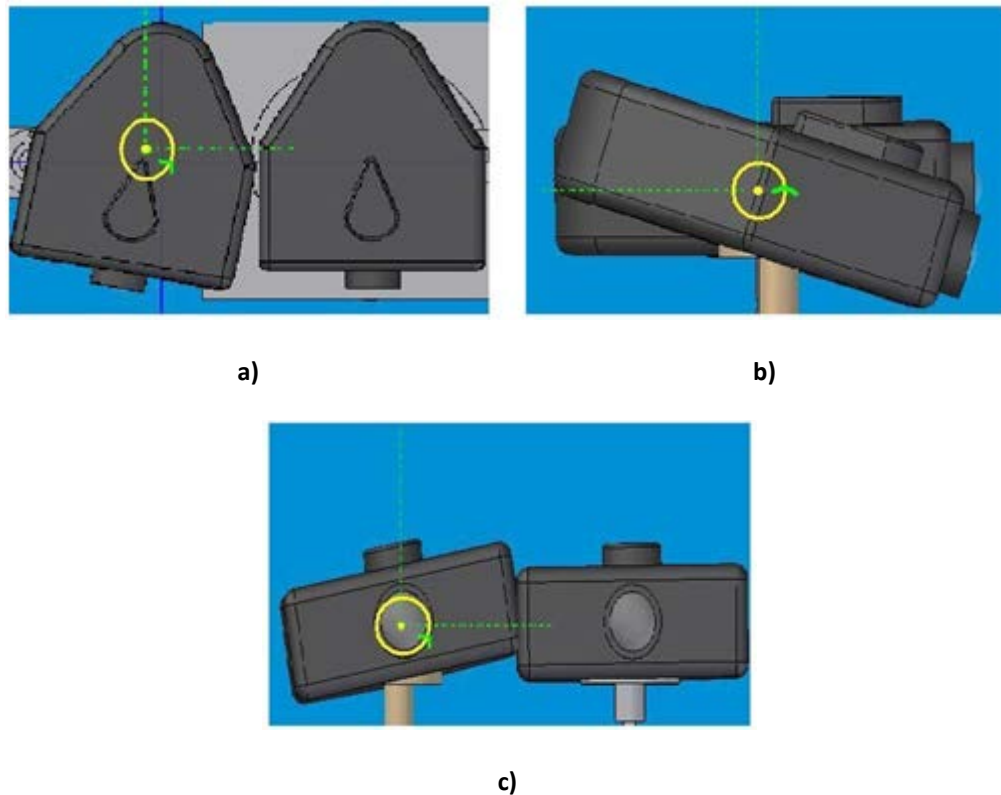


Fig. 3.1. Nivelación de los planos imagen del par estereoscópico. Fuente propia.

Dentro de los 3 grados de libertad posibles, existe uno que se debe considerar más que los otros dos. El movimiento rotacional más crítico se muestra en la Fig. 3.2a. A este movimiento en particular se ha prestado especial atención, con la ayuda del desplazamiento helicoidal de un tornillo micrométrico se ha obtenido alta resolución para el movimiento rotacional requerido. En cuanto a los otros 2 grados de libertad, uno de ellos se ajusta de forma manual, ver Fig. 3.2b, el otro movimiento se omite para evitar complicaciones en la implementación del soporte y por no tener mayor trascendencia en la calibración del par estereoscópico, mostrado en la Fig. 3.2c. Las trayectorias en amarillo indican las posibles rotaciones de la cámara.



**Fig. 3.2. Posibles rotaciones de la cámara móvil para alineación de planos imagen. a) Movimiento fino (con tornillo micrométrico). b) movimiento manual. c) movimiento omitido. Fuente propia.**

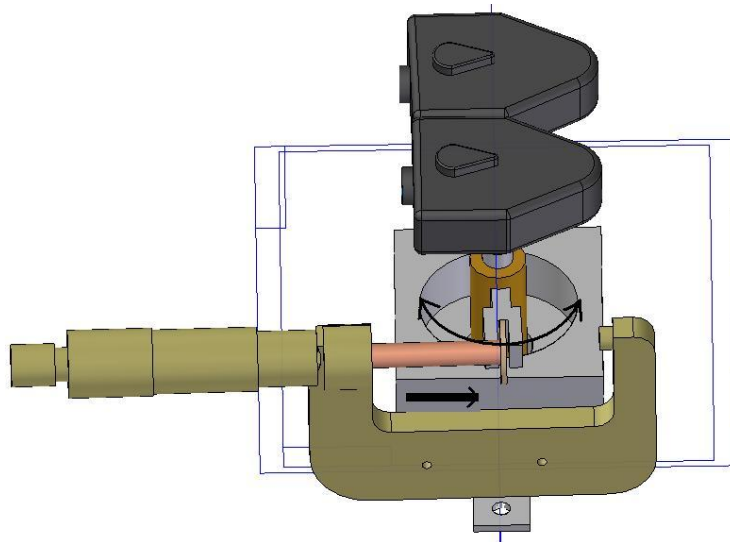
El movimiento rotacional fino mostrado en la Fig. 3.2a, se hace partiendo de un tornillo micrométrico. El tornillo tiene resolución de 0.001mm por cada paso, el eje del tornillo se

desplaza 25mm en forma lineal, este movimiento lineal se utiliza para generar el movimiento rotacional mencionado.

Se acopló el eje del tornillo micrométrico a una pestaña que se encuentra en el eje de la cámara forzando el giro y obteniendo una resolución proporcional a la sensibilidad del tornillo micrométrico, según la ecuación 3.1. En la Fig. 3.3 se muestra el acople entre las piezas, aquí las flechas en negro indican el movimiento lineal del eje del micrómetro y el movimiento rotacional del eje de la cámara.

$$\Delta S = r \times \Delta \theta \quad (3.1)$$

Donde  $\Delta S$  es la resolución del tornillo micrométrico (0.001mm/paso),  $r$  la distancia entre el eje de la cámara y el eje del tornillo (30mm), y  $\Delta \theta$  la resolución del movimiento rotacional de la cámara.



**Fig. 3.3. Rotación del eje de la cámara generada con el tornillo micrométrico. Fuente propia.**

La resolución obtenida para el movimiento rotacional de la cámara es 0,0191 grados por paso. Esto es válido siempre y cuando el ángulo de rotación de la cámara sea pequeño, lo cual no representa mayor inconveniente, ya que la cámara estará rotando unos pocos grados para lograr el paralelaje de los ejes ópticos.

Es de interés en el trabajo de investigación, conocer las variación en la resolución debida a la variación de la distancia *baseline* (B). Por este motivo, el diseño del soporte del par estereoscópico cuenta con la posibilidad de ubicar las cámaras a 4cm y a 6cm de distancia una de la otra.

### 3.2.2 Vista 3D del hardware

El diseño CAD (*Computer Assisted Design*) del soporte para las cámaras web se realizó en *Solid Edge V17*. El resultado final del diseño y la implementación del hardware de visión estereoscópica se muestra en la Fig. 3.4.

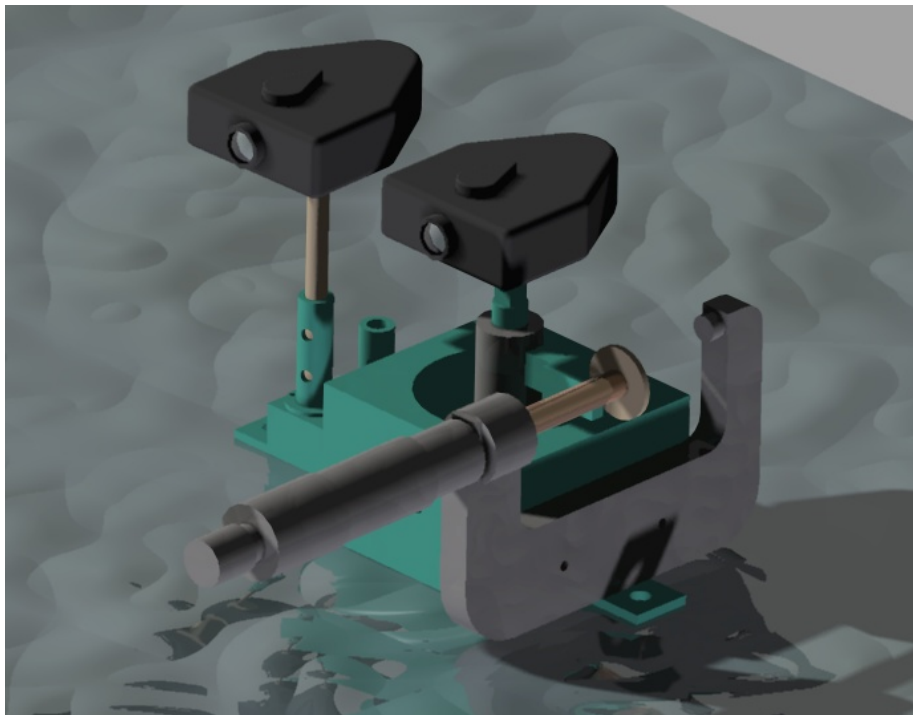


Fig. 3.4. Diseño del soporte de visión estereoscópica en *Solid Edge*. Fuente propia

### 3.2.3 Implementación del hardware

Para la construcción del sistema hardware que soporta el par estereoscópico fue necesario la fabricación de 7 piezas. Una base firme en la que se anclarán las demás piezas (*base*). Dos cilindros huecos en los que se sujeta el eje de la cámara estática (*cilindros*). Un eje acoplado a la cámara estática (eje estático). Un cilindro que sirve como eje de rotación para la cámara móvil (*eje móvil*). Una pieza anclada a la cámara móvil que a su vez se sujeta en un pivote unido a la pieza anterior (*pivote cámara*). Una pestaña que se acopla el eje rotatorio de la cámara móvil con el tornillo micrométrico (*pestaña*). En la Fig. 3.5 se muestra el soporte estéreo (construido por la empresa Metálicas Velasco de la ciudad de Popayán) y se identifican con flechas las 7 piezas que lo conforman.

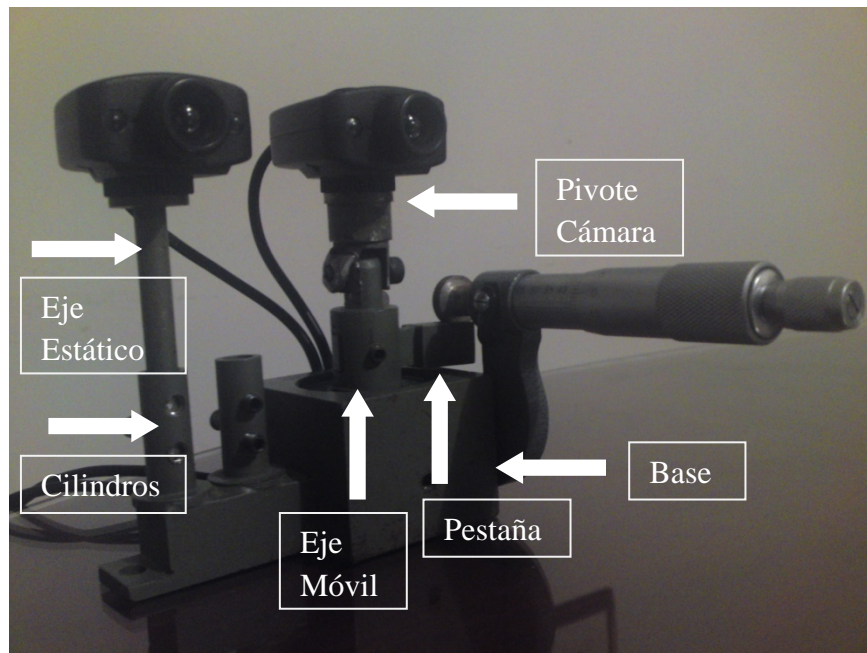


Fig. 3.5 Piezas del sistema estéreo implementado. Fuente propia.

### **3.3 SISTEMA ESTÉREO SOFTWARE**

Como el sistema estereoscópico está pensado para aplicaciones robóticas, el sistema operativo utilizado es Linux. Específicamente se utilizó la distribución Ubuntu 9.10, esto gracias a que reduce el tiempo de ejecución de los algoritmos y presenta mayor estabilidad en rendimiento de la memoria y del procesador. Además es software libre y las principales plataformas robóticas hacen uso de esta filosofía.

El sistema estéreo se implementó en la plataforma robótica Player 3.0.1. Esta es una de las plataformas más usadas en el campo de la robótica y es un servidor de red para el control de robots. Player ofrece una interfaz simple de todos los sensores y actuadores del robot sobre una red IP. Su programa de conversaciones cliente a servidor se hace sobre un socket TCP, leyendo los datos de los sensores, escribiendo a los actuadores y a los dispositivos de la configuración sobre la marcha [31].

#### **3.3.1 Captura de video**

La captura de video se hace a través de dos métodos, el primero es usando funciones de la librería de visión por computador libre OpenCV [32] y el segundo es usando funciones de la plataforma robótica Player.

Para tomar la señal de video de las cámaras con OpenCV, se utiliza la función `cvCaptureFromCAM`, esto se hace mediante un archivo punto c y la ventaja es que no hay que involucrarse con el controlador de las cámaras. Mientras que si el ingreso a estas, se realiza con funciones de Player, se debe crear anticipadamente el controlador, lo cual se hace en un archivo con extensión `cfg` y es el que permitirá acceder a las cámaras con protocolo TCP/IP, haciendo de ellas un cliente del servidor Player.

En la práctica se encontró que utilizar las funciones de OpenCV para acceder a las cámaras es más eficiente, en cuanto al tiempo de la captura de los marcos que utilizar las funciones de Player. La desventaja de utilizar el acceso mediante OpenCV radica en que el sistema de visión estéreo no sería un dispositivo de Player, generando problemas de sincronización con otros dispositivos (radar laser, motores, sistema de odometría, etc.). Además OpenCV no permite la comunicación directa mediante protocolo TCP/IP, lo cual complica el desarrollo de aplicaciones para robots tele-operados. Precisamente, las desventajas que se acaban de nombrar son las razones por las cuales se diseñó Player.

### **3.3.2 Corrección de distorsiones geométricas**

Las incógnitas que se quieren resolver a través de la calibración son cuatro parámetros intrínsecos ( $f_x$ ,  $f_y$ ,  $C_x$ ,  $C_y$ ) y cinco parámetros de distorsión: tres radiales ( $K_1$ ,  $K_2$ ,  $K_3$ ) y dos tangenciales ( $P_1$ ,  $P_2$ ). Los parámetros intrínsecos están directamente relacionados a la geometría 3D y por tanto a los parámetros extrínsecos, mientras que los parámetros de distorsión están vinculados a la geometría 2D en el plano imagen, de la distorsión de los puntos proyectados [16]. Por lo tanto estas dos clases de parámetros pueden tratarse de forma separada.

Para el cálculo de los parámetros de distorsión basta con la vista de un tablero de ajedrez y algunos cálculos.

Para los parámetros extrínsecos, se necesita saber en dónde está el tablero de ajedrez, esto requiere de 3 ángulos de rotación y tres parámetros para la traslación. Estos seis parámetros cambian constantemente con cada nueva imagen del tablero de ajedrez, en tanto que los cuatro parámetros intrínsecos ( $f_x$ ,  $f_y$ ,  $C_x$ ,  $C_y$ ) se mantienen constantes para cada vista del tablero. Es decir si  $K$  representa el número de imágenes que se toman al



tablero de ajedrez, entonces el total de parámetros que se encuentra para el conjunto de imágenes es  $6K + 4$ .

Dicho lo anterior se necesita saber el número de ecuaciones que se pueden obtener de las  $K$  imágenes, entonces si  $N$  representa el número de aristas del tablero, el número de ecuaciones que se obtienen del conjunto de imágenes es  $2NK$ , donde se usa el multiplicador 2 ya que cada punto de la imagen tiene una coordenada  $X$  y una coordenada  $Y$ .

Por lo tanto para resolver el problema se debe cumplir la restricción 3.2.

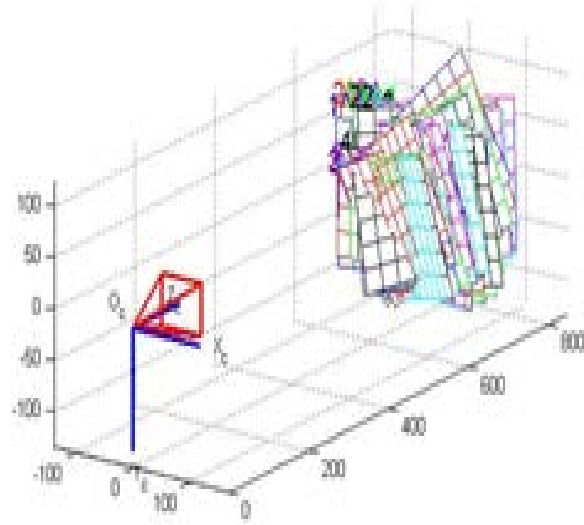
$$2NK \geq 6K + 4 \quad (3.2)$$

o equivalentemente se obtiene la ecuación 3.3.

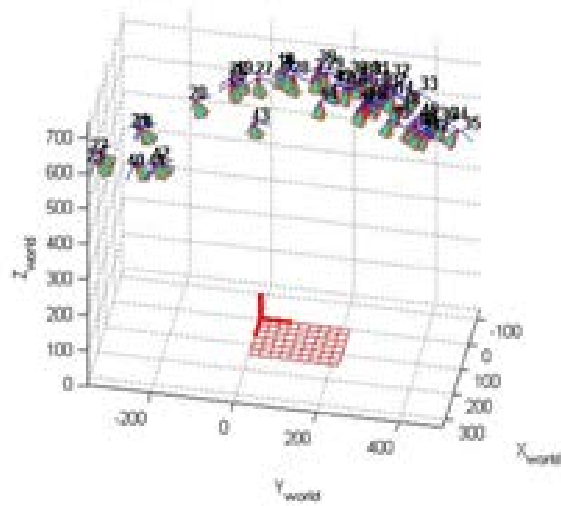
$$(N - 3)K \geq 2 \quad (3.3)$$

Hay que tener en cuenta que al utilizar tableros de ajedrez solo se está obteniendo información de un plano, por lo que basta con cuatro puntos para expresar toda la información de una vista en perspectiva, es decir para cada vista se tiene que  $N = 4$ , esto conlleva a que  $K \geq 2$ , con lo que se concluye que bastan solo 2 puntos de vista del tablero de ajedrez como mínimo para encontrar los parámetros.

En la práctica para obtener resultados de alta calidad se necesitan al menos diez imágenes de un tablero de ajedrez de 7x8 cuadros. Para el algoritmo empleado en este trabajo se emplearon 50 imágenes de un tablero de ajedrez de 8x12 esquinas internas. En la Fig. 3.6a se presenta las poses tomadas para la cámara derecha tomando como sistema de referencia la posición de la cámara, y en la Fig. 3.6b las poses para esta misma cámara tomando como sistema de referencia el tablero de calibración.



a)



b)

**Fig. 3.6 Poses del tablero para la calibración de cámara. a) Referencia en la cámara b) Referencia en el tablero. Fuente propia.**

Los parámetros intrínsecos y de distorsión para las dos cámaras después del proceso de calibración descrito se muestran en la tabla 3.1.

**Tabla 3.1.** Parámetros intrínsecos. Fuente propia.

Parámetro	Cámara Derecha	Cámara Izquierda
$f_x$	499.64377	488.37256
$f_y$	500.94366	487.97961
$C_Y$	117.07956	125.64400
$C_X$	159.32002	153.52606
$K_1$	-0.1256371	-0.1279371
$K_2$	-0.91264373	-1.4132898
$K_3$	2.30457926	8.2168178
$P_1$	2.35190e-003	7.31014e-004
$P_2$	1.186088e-003	4.29038e-004

Con los valores de la tabla 3.1 se obtienen las expresiones para las posiciones de los píxeles de la imagen corregida según la ecuación 1.33, las ecuaciones de corrección de la cámara derecha se presentan en la ecuación 3.4 para distorsión radial y en la ecuación 3.5 para distorsión tangencial.

$$\begin{aligned} X_p &= X(1 - 0.1256371 \times r^2 - 0.91264373 \times r^4 + 2.30457926 \times r^6) \\ Y_p &= Y(1 - 0.1256371 \times r^2 - 0.91264373 \times r^4 + 2.30457926 \times r^6) \end{aligned} \quad (3.4)$$

$$\begin{aligned} X_p &= X + [2 \times 2.35190e - 003Y + 1.186088e - 003(r^2 + 2X^2)] \\ Y_p &= Y + [21.186088e - 003 \times X + 2.35190e - 003(r^2 + 2Y^2)] \end{aligned} \quad (3.5)$$

Las distorsiones radiales son mayores que las distorsiones tangenciales, sin embargo las distorsiones radiales y tangenciales son pequeñas, y no implican un gran cambio en las imágenes. Las ecuaciones de corrección de la cámara izquierda son muy similares y tienen las mismas características.

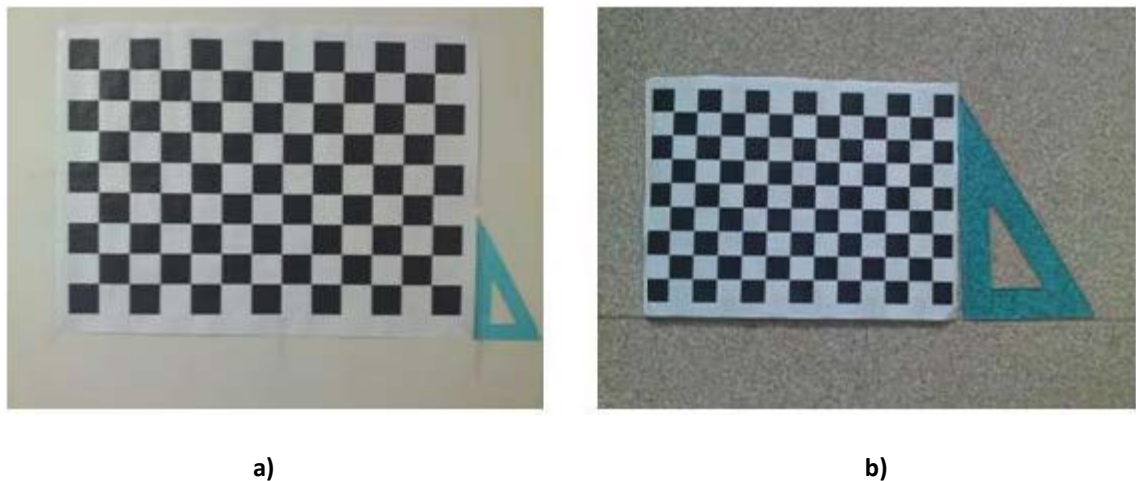
### 3.3.3 Rectificación software

Es relativamente fácil encontrar el mapa de disparidad si las imágenes estéreo están rectificadas, es por esta razón que el proceso de rectificación software adquiere gran importancia. La idea general es transformar la imagen de tal forma que las líneas epipolares coincidan con las filas de las imágenes estereoscópicas.

Existen muchos algoritmos para calcular los términos de rectificación. En el presente trabajo se utilizó el algoritmo de Hartley descrito en la sección 2.2.4.1, método que usa la matriz fundamental para la rectificación [19]. Para encontrar la matriz fundamental se usan varias vistas de un tablero de calibración. En este proceso además de la matriz fundamental es posible determinar la matriz esencial  $E$ , la matriz de rotación y el vector de traslación relacionados con la homografía.

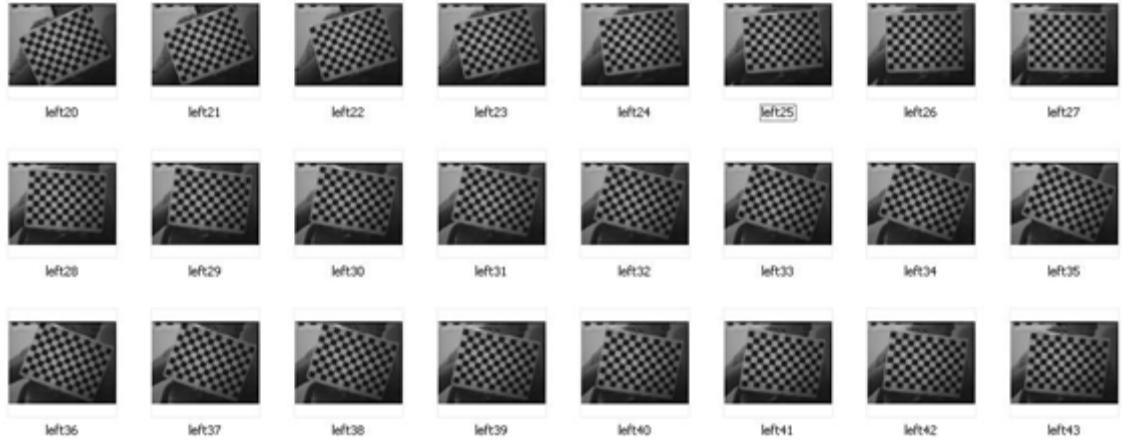
Como se mostró en la sección 3.2, el sistema estéreo tiene la posibilidad de variar la distancia entre las cámaras a 4cm y 6cm. A su vez el rango de funcionamiento del sistema estéreo disminuye con la disminución del *baseline*, es decir que el espacio de trabajo del sistema se encuentra cerca del par estereoscópico cuando la separación entre las cámaras es pequeña, y el espacio de trabajo se aleja del sistema estéreo a medida que se separa una cámara de la otra. Por ésta razón, es conveniente calibrar las cámaras en un lugar cercano al espacio de trabajo. Por esta razón se decidió utilizar dos tableros de calibración, uno pequeño para realizar el proceso de calibración de las cámaras aproximadamente a 1.5 metros alejado de ellas, para *baseline* de 4cm. Y un tablero de calibración más grande para realizar la calibración a 4 metros de distancia al sistema estéreo, para *baseline* de 6cm.

Se decidió rectificar las imágenes estéreo a distintas distancias dependiendo de la separación entre ellas. La razón principal es que el conjunto de algoritmos de calibración obtiene la solución a las respectivas ecuaciones para los puntos del tablero en particular, y la corrección de distorsiones arroja mejores resultados alrededor de los puntos cercanos a los utilizados para la calibración. En la Fig. 3.7 se muestran los dos tableros utilizados para la rectificación de cámaras.

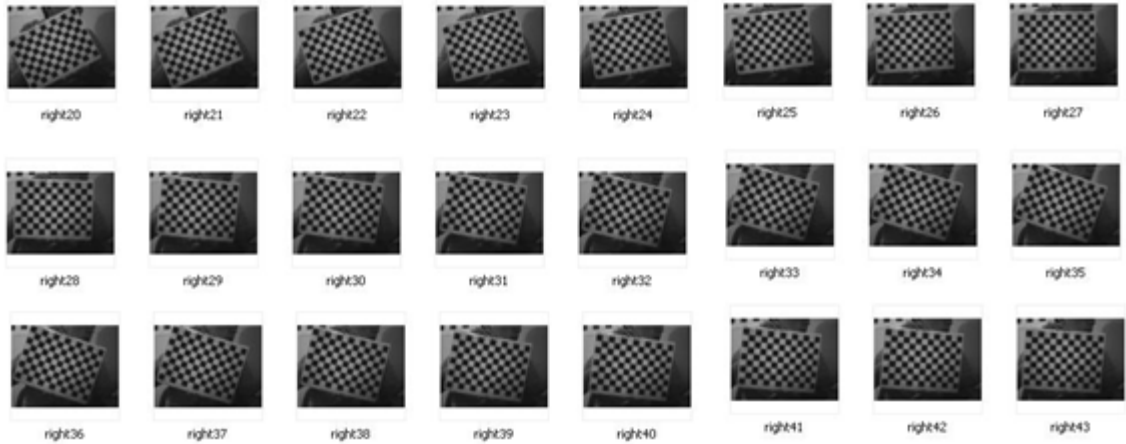


**Fig. 3.7 Tableros para rectificación de cámaras. a) Tablero utilizado para *baseline* de 6cm, b) Tablero utilizado para *baseline* de 4cm. Fuente propia.**

El algoritmo de rectificación necesita varias vistas del tablero, para cuyo propósito se capturaron 50 imágenes para cada cámara. El algoritmo carga las imágenes y calcula la matriz fundamental y la homografía del sistema. Con esta información se re-calculan las coordenadas que cada pixel de las imágenes debe ocupar, con las nuevas posiciones de los pixeles se debe obtener una configuración estéreo estándar. Algunas de las imágenes utilizadas para la rectificación se muestran en la Fig. 3.8.



a)

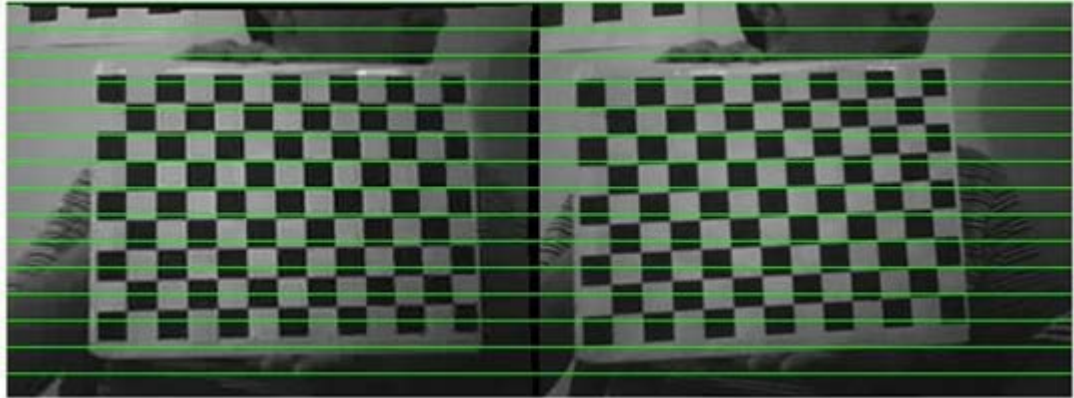


b)

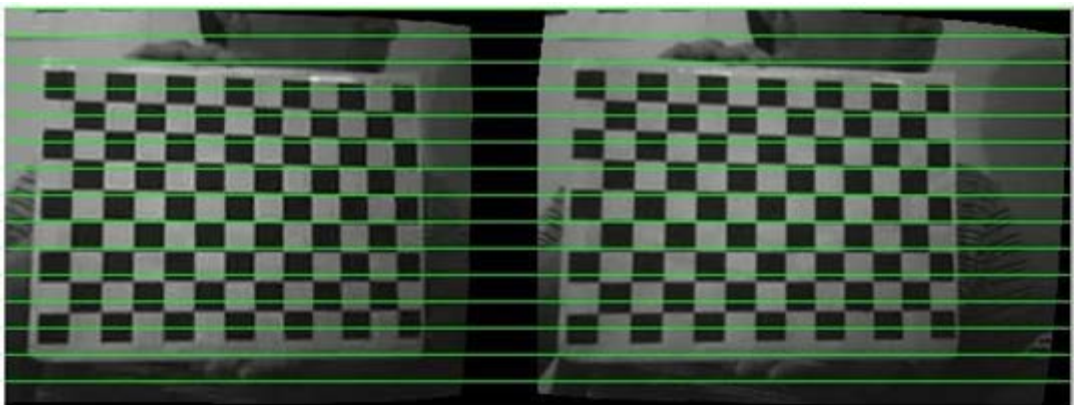
**Fig. 3.8 Pares de imágenes usadas en el proceso de rectificación, a) cámara izquierda, b) cámara derecha.**

**Fuente propia.**

En el proceso de rectificación se obtienen generalmente errores menores a un 1 pixel. El menor error obtenido en las pruebas de rectificación fue 0,243165. En la Fig. 3.9 se muestra el par de imágenes antes y después del proceso de rectificación.



a)



b)

Fig. 3.9 Proceso de rectificación, a) Imágenes solamente con alineación hardware, b) Imágenes rectificadas. Fuente propia.

### 3.3.4 Disparidad densa

Una vez conocida la geometría epipolar y rectificado el par estéreo, es posible considerar dos opciones para encontrar el mapa de disparidad [33]. Si se elige buscar la disparidad para cada pixel de la imagen, se aplica algún método de búsqueda de correspondencia para toda la imagen y se calcula la disparidad para cada uno de los pixeles. Si se elige buscar la disparidad por características, se debe decidir que características utilizar, y se calcula la disparidad para aquellos pixeles que tengan dicha característica.

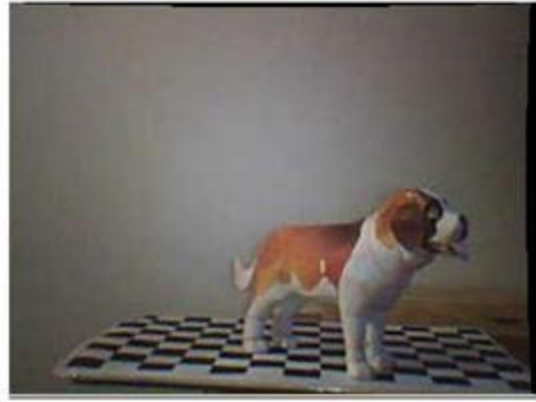
En el presente trabajo se exploraron los dos caminos antes mencionados. Para el primero de ellos se utilizó el algoritmo de búsqueda de correspondencia basado en suma de diferencias absolutas SAD, descrito en la sección 2.4.1., de esta forma se obtiene un mapa de disparidad denso. Este procedimiento se realizó para los dos *baseline* que permite el dispositivo. En la Fig. 3.10 se muestran los resultados obtenidos.



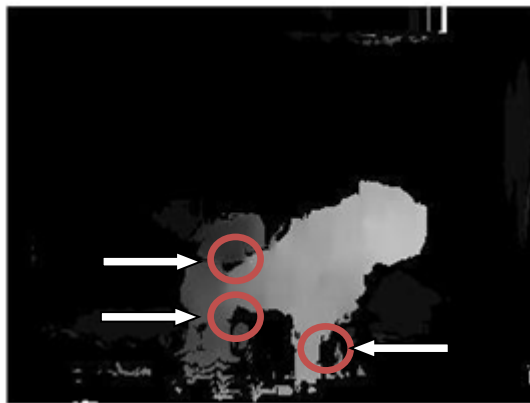
**Fig. 3.10** Disparidad densa con *baseline* de 4cm, a) Imagen izquierda, b) Mapa de disparidad. Fuente propia.

En los mapas de disparidad se establecen zonas en las que el algoritmo de correlación no tiene éxito, esto se debe principalmente a errores en la rectificación y principalmente a la falta de textura en la escena. Se presentan en las imágenes discontinuidades o saltos bruscos de la disparidad. Para dar una solución parcial a este problema se aplicó filtros de media en los mapas de disparidad. En la Fig. 3.11 se muestran algunos resultados que dejan ver el trabajo del filtro sobre la imagen.

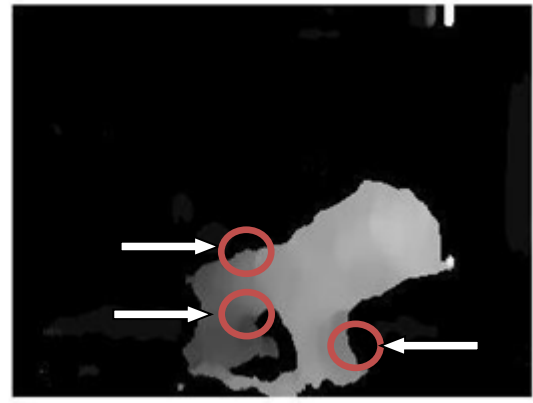




a)



b)



c)

**Fig. 3.11 Efecto del filtro de media en los mapas de disparidad., a) Imagen izquierda, b) Mapa de disparidad sin filtrar, c) Mapa de disparidad filtrado. Fuente propia.**

La Fig. 3.11 muestra que el tratamiento hecho al mapa de disparidad trabaja sobre zonas en las que la correspondencia no fue encontrada, marcadas con círculos, e interpola algunos valores de disparidad para esos lugares.

El mapa de disparidad representa la profundidad de la escena en niveles de gris, por lo general con valores de intensidad de 0 a 255. Sin embargo, en ocasiones es más favorable representar la profundidad usando imágenes de 3 canales RGB. Las imágenes en pseudocolor son muy utilizadas para representar una escena tridimensional sobre una imagen

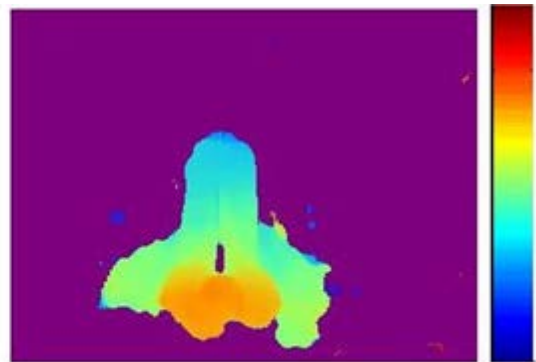
plana [34]. En esta representación, la tendencia al color rojo indica aumento de disparidad y disminución de profundidad, y la cercanía al azul es equivalente a disparidad baja y mayor profundidad. En la Fig. 3.12 se muestra una escena, su mapa de disparidad y su representación en pseudo-color.



a) Imagen izquierda



b) Mapa de disparidad



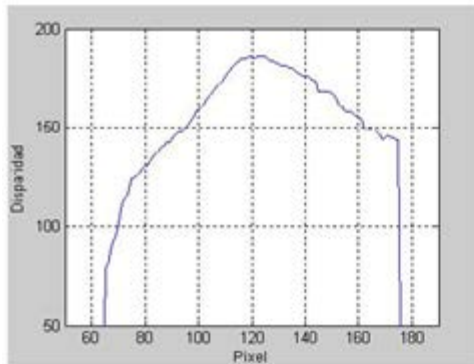
c) Mapa en pseudocolor

**Fig. 3.12 Representación del mapa de disparidad en pseudo-color. Fuente propia.**

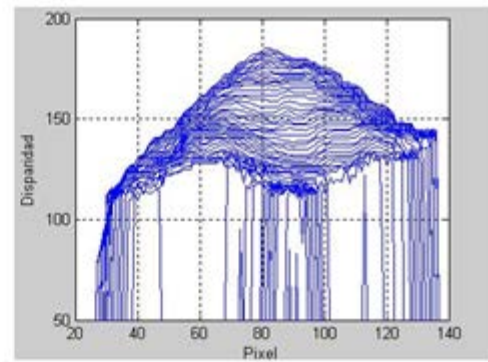
En ocasiones es conveniente representar la profundidad usando menos información. El perfil de profundidad es una opción que muestra la disparidad o profundidad de algunos lugares de la escena [35]. Por ejemplo de la Fig. 3.13a se puede hacer el perfil de profundidad de la fila central, tal como se observa en la Fig. 3.13b, o una cantidad reducida de perfiles de línea como se presenta en la Fig.3.13c.



a)



b)



c)

Fig. 3.13 Perfil de disparidad, a) Un solo perfil de disparidad, b) varios perfiles de disparidad. Fuente propia.

### 3.3.5 Disparidad por extracción de características

Una alternativa a la solución del problema de visión estéreo es buscar la disparidad de algunas partes representativas de la escena. Se obtuvieron estas características particulares aplicando la derivada en la dirección de las filas y en la dirección de las columnas, obteniendo principalmente bordes y esquinas. En la Fig. 3.14 se presenta la operación realizada para un marco en particular, el recuadro superior muestra una sección aumentada de la imagen derivada, marcada con una flecha roja.



**Fig. 3.14 Extracción de características con filtro Sobel. Fuente propia.**

El proceso de extracción de características permite tomar de la imagen aquellos píxeles que son más representativos. Una imagen de tamaño 320x240 tiene 76800 píxeles, en una imagen como la mostrada en la Fig. 3.14 se extraen alrededor de 13000 píxeles. Aunque esta muestra representativa filtra alrededor de 64.000 píxeles, sigue siendo un número grande de datos para procesar. Por esta razón se decidió tomar uno de cada tres píxeles que pasan el filtro de características, reduciendo el volumen de datos a tratar para disminuir el consumo de procesador. En la Fig. 3.15 se observa el resultado de este procedimiento. Se resalta la misma sección de la Fig. 3.14 y se encuentra que existen lugares en los que se han eliminado puntos pertenecientes a los bordes.



Fig. 3.15 Eliminación uniforme de puntos característicos. Fuente propia.

El siguiente paso es encontrar los puntos correspondientes de aquellos que han pasado el filtrado. Para el emparejamiento de puntos se utilizó el concepto de flujo óptico descrito en la sección 2.4.2, en particular el método de *Lucas-Kanade* descrito en 2.4.2.1.

El algoritmo de *Lucas-Kanade* toma el conjunto de puntos extraídos en el filtrado de la imagen derecha y busca un conjunto de puntos correspondiente en la imagen izquierda. En la Fig. 3.16 se muestra el algoritmo aplicado a una imagen. Los puntos rojos muestran puntos de interés, los puntos azules son los puntos emparejados en la segunda imagen, y las líneas verdes son las líneas de flujo óptico que nacen en los puntos rojos y terminan en los verdes.

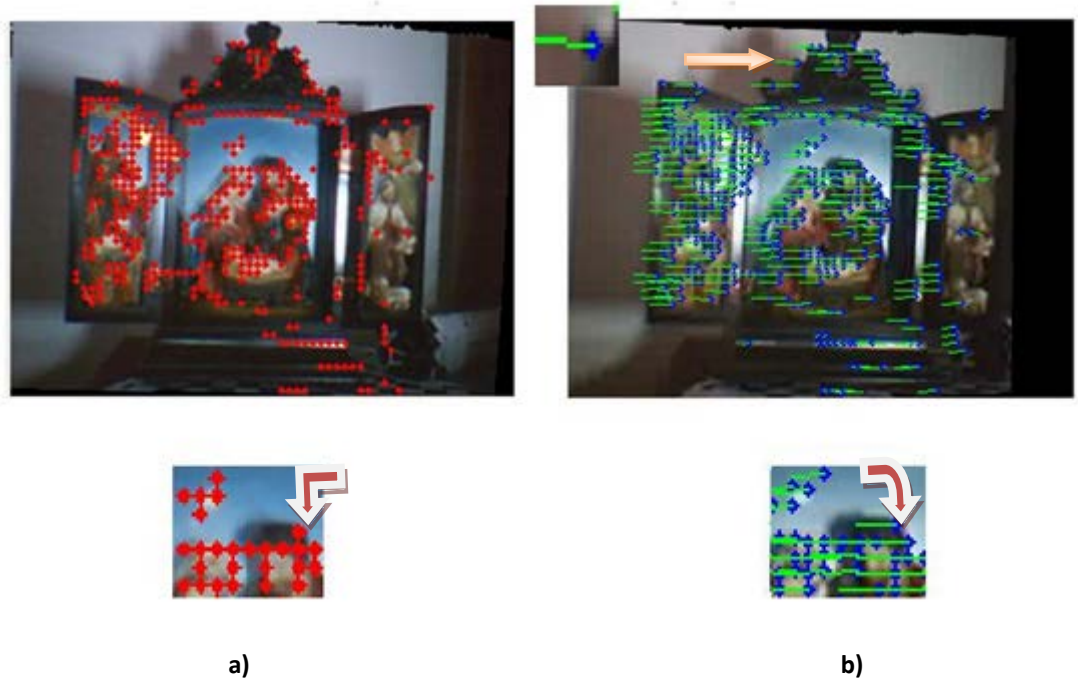
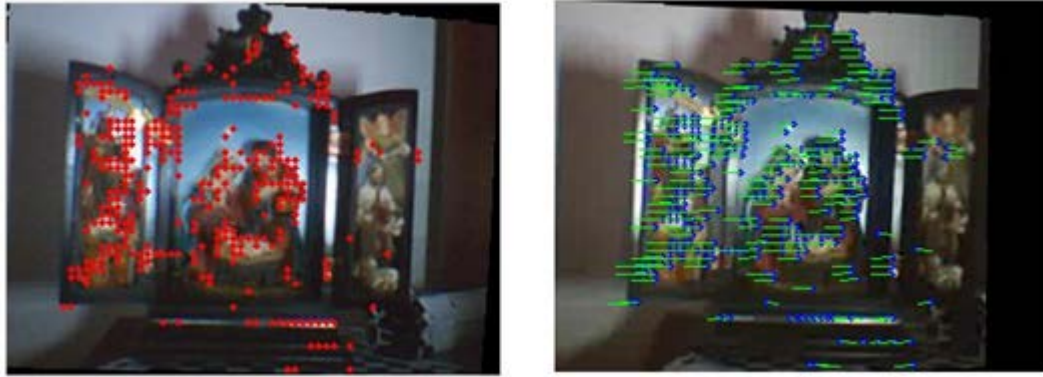


Fig. 3.16 Búsqueda de correspondencia basada en flujo óptico, a) Características, b) Emparejamiento.

Fuente propia.

### 3.3.6 Restricción epipolar

Si las imágenes están rectificadas, es de esperarse que las líneas de flujo óptico vayan en dirección de las filas. En la Fig. 3.16 existen algunos puntos de correspondencia con líneas de flujo óptico no paralelas a las filas de la imagen, como la que se muestra en el acercamiento. Los emparejamientos que no cumplan con la geometría epipolar conducirán a errores en el cálculo de la profundidad en esos puntos, sin importar que la correspondencia sea exitosa o no. Para evitar este tipo de errores se debe filtrar aplicando lo que se denomina la restricción epipolar, es decir conservar los emparejamientos que estén sobre las líneas epipolares. En la Fig. 3.17 se muestra el resultado de aplicar la restricción.



a)

b)

**Fig. 3.17 Correspondencia usando flujo óptico aplicando restricción epipolar, a) Características, b) Emparejamiento. Fuente propia. Fuente propia.**

En la Fig. 3.17 las líneas de flujo óptico corresponden con las filas, de esta forma todos los emparejamientos cumplen con la restricción epipolar. Las líneas verdes, que representan el flujo óptico en la imagen nacen en un punto rojo y terminan en uno verde, pero lo hacen desplazándose sobre una única fila.

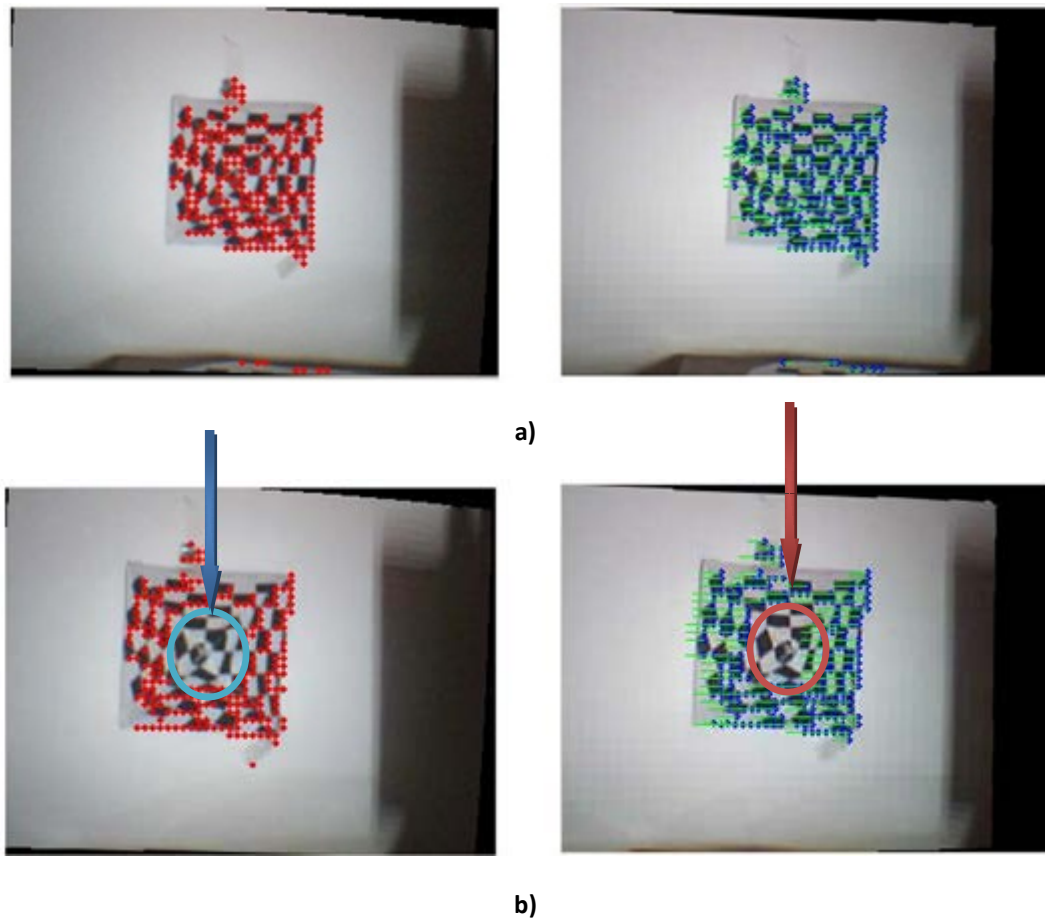
### **3.3.7 Restricción de máxima disparidad**

Una condición importante es la disparidad máxima que se permite en el sistema, para determinar el máximo valor de disparidad se debe conocer el rango de funcionamiento del sistema. Este procedimiento se verá en el capítulo 4. Por ahora es conveniente revisar el proceso de filtrado de puntos emparejados con la restricción de la máxima disparidad.

En este filtrado se toma como emparejamientos válidos sólo aquellos que estén relacionados con un valor de disparidad menor a un determinado umbral. De esta forma se está delimitando la mínima profundidad que medirá el sistema estéreo. En la Fig. 3.18a se muestra el emparejamiento sin la restricción de disparidad máxima, y en la Fig. 3.18b se presenta el resultado de esta restricción, donde la mínima profundidad permitida se



encuentra en un lugar sobre la pirámide. En este caso particular se ha considerado 16 el valor de máxima disparidad, por esta razón los valores más cercanos al sistema de visión no aparecen como emparejamientos válidos. En la Fig. 3.18, los puntos en rojo son características extraídas de la imagen, los puntos verdes son los emparejamientos en la segunda imagen, las líneas verdes representan el flujo óptico. Los círculos indican los lugares para los cuales la disparidad es mayor a la disparidad máxima permitida, y son ignorados por el algoritmo.



**Fig. 3.18 Correspondencia estéreo (pirámide con textura), a) Sin restricción de máxima disparidad, b) con restricción de máxima disparidad. Fuente propia.**

La restricción epipolar y la de máxima disparidad evitan que existan apareamientos indeseados, terminado el proceso de búsqueda de correspondencia es posible encontrar el mapa de disparidad no denso para las características relevantes de la escena. En la Fig.



3.19a se muestra el proceso de búsqueda de características, en la Fig. 3.19b se presenta el mapa de disparidad, y en Fig. 3.19c la representación en pseudo-color.

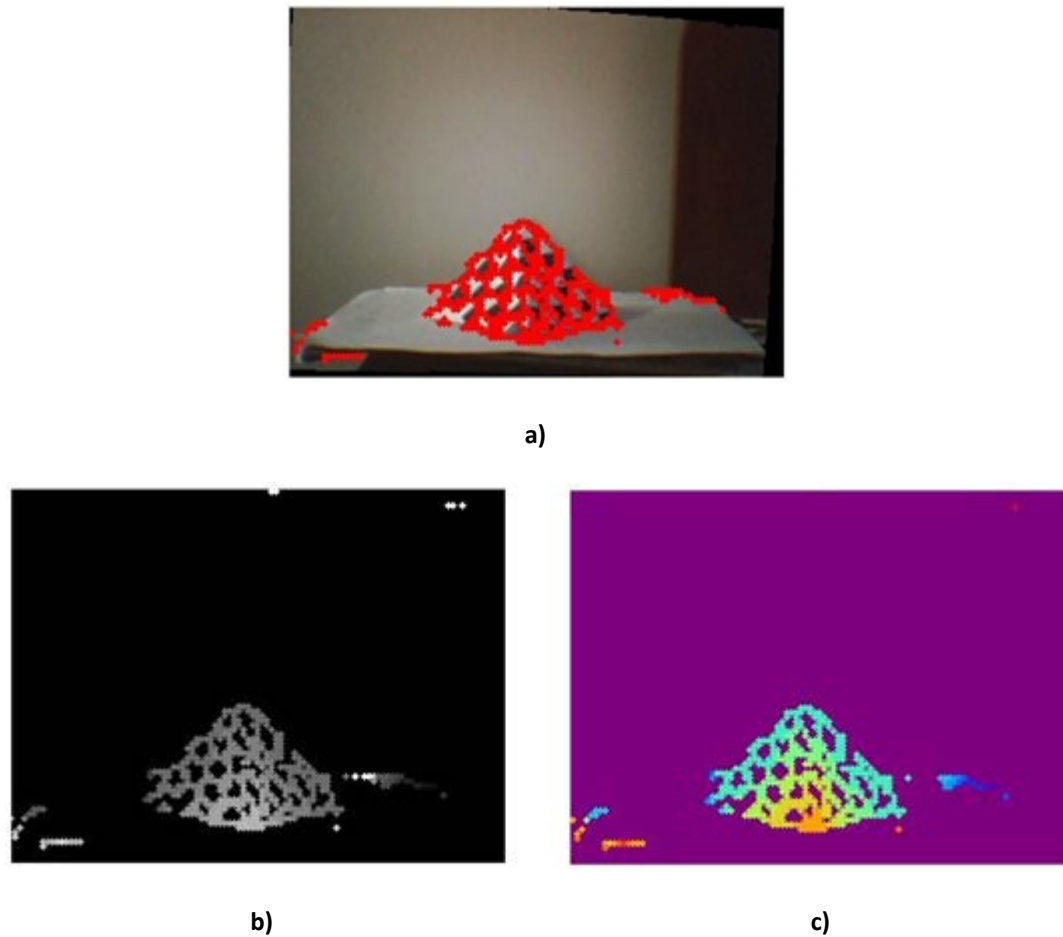


Fig. 3.19 Correspondencia de características relevantes de la escena, a) Características seleccionadas, b) Mapa de disparidad, c) Representación en pseudo-color. Fuente propia.

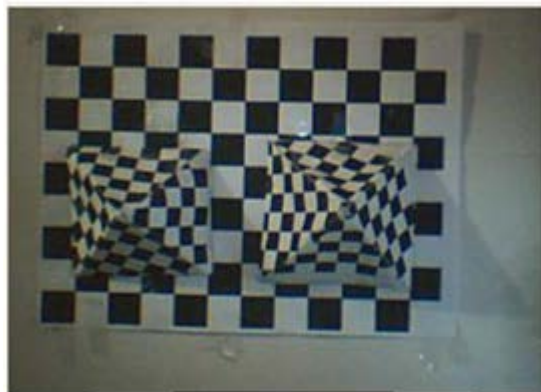
### 3.3.8 Representación 3D

Partiendo del mapa de disparidad es posible obtener las tres coordenadas espaciales (X,Y,Z), según la expresión la ecuación 2.26. Hasta ahora en el trabajo se ha representado la profundidad usando mapas de disparidad por regiones o de características, y usando el

perfil de línea del mapa de disparidad. Ahora se presentarán los resultados en un escenario tridimensional.

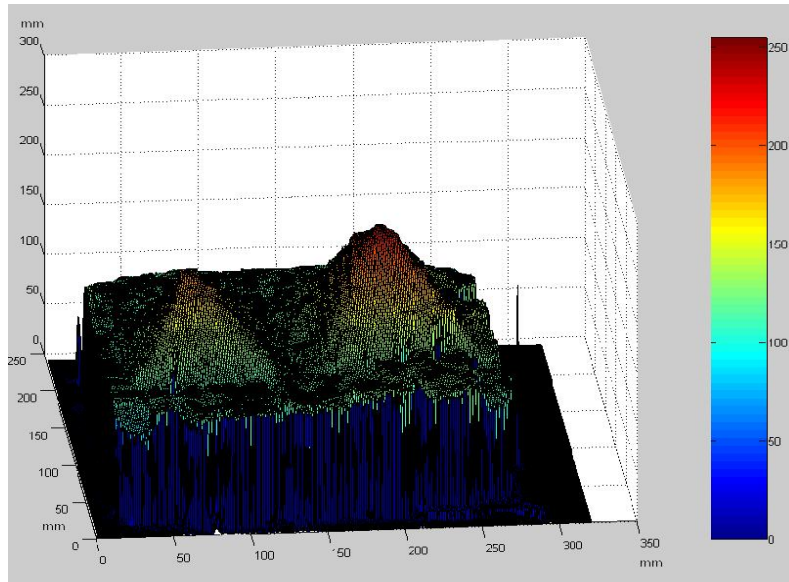
Se realiza la representación tridimensional de los datos en dos programas software diferentes: Matlab y Glut. En Glut se grafican los puntos en un escenario 3D con su respectivo nivel de gris en los canales RGB. Este método es rápido y puede realizarse *on-line*. La segunda representación de los datos se realiza con Matlab usando mallas tridimensionales. El tiempo que demora la ejecución del algoritmo es grande y el proceso debe realizarse *off-line*. Se deben distinguir dos tipos de aplicaciones del sistema de visión: la primera permite detectar objetos u obstáculos *on-line*, la segunda hacer reconstrucción tridimensional de una escena, que puede utilizarse para realizar medidas sobre ella.

En la Fig. 3.20 se encuentran dos pirámides de diferentes dimensiones, sobre un tablero de ajedrez.

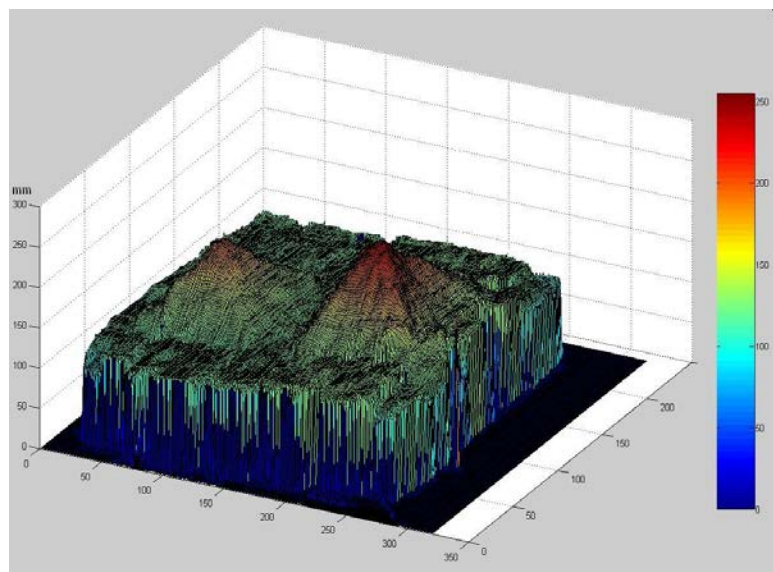


**Fig. 3.20 Escena a reconstruir (con textura). Fuente propia.**

Se utiliza a lo largo de todo el trabajo de investigación software libre y principalmente Linux. En particular la representación tridimensional de las escenas se realiza con Glut y OpenCV, sin embargo se realizaron reconstrucciones 3D usando mallas en el entorno de Matlab. En la Fig. 3.21 se muestra la reconstrucción de la escena de la Fig. 3.20 usando mallas.



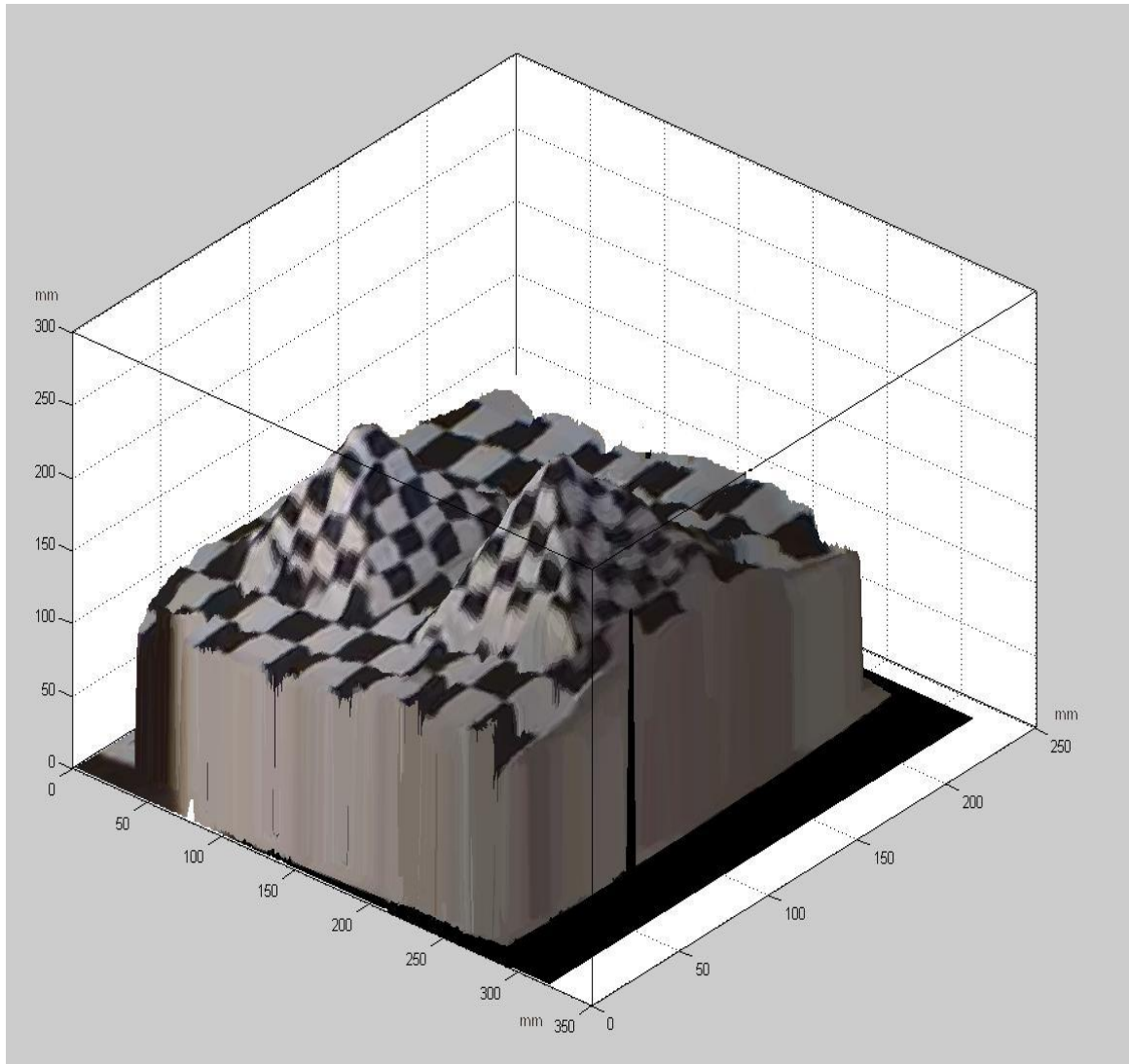
a)



b)

**Fig. 3.21 Reconstrucción tridimensional en Matlab. Fuente propia.**

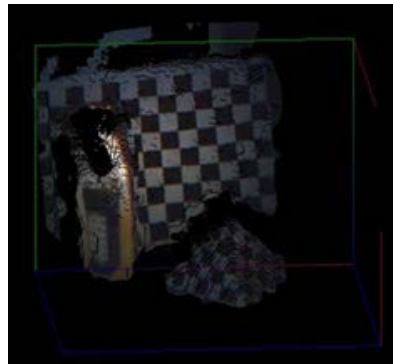
Se realiza el *renderizado* utilizando la información de la imagen original mostrada en la Fig. 3.20 con el fin de obtener mayor realismo en la reconstrucción. En la Fig. 3.22 se muestra la reconstrucción tridimensional con la textura original de la escena.



**Fig. 3.22 Reconstrucción tridimensional con texturas en Matlab. Fuente propia.**

Las reconstrucciones mostradas se ejecutan *off-line*, debido al tiempo que toma el procesador en generar la malla y *renderizar* la escena. Sin embargo es de interés en el trabajo de investigación lograr recrear la escena 3D con el dispositivo en funcionamiento. El sistema debe ser capaz de realizar la reconstrucción tridimensional de cada marco capturado por las cámaras. Se presentan a continuación la reconstrucción de escenas en video, a partir de disparidad de características y disparidad densa.

En la Fig. 3.23 se presenta la reconstrucción de un escenario en secuencia de video. La escena se recrea a partir de un mapa de disparidad denso.



a)

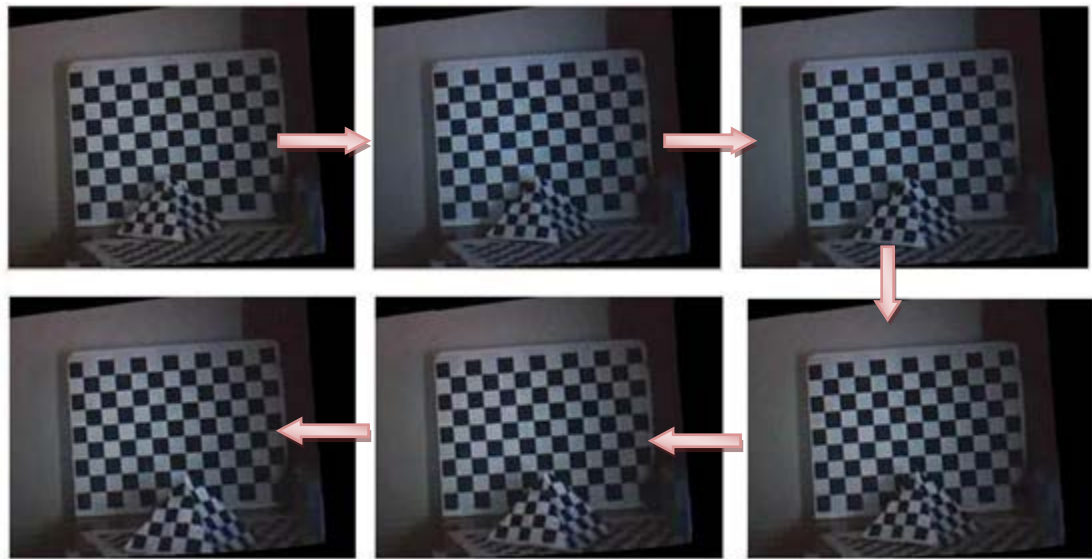


b)

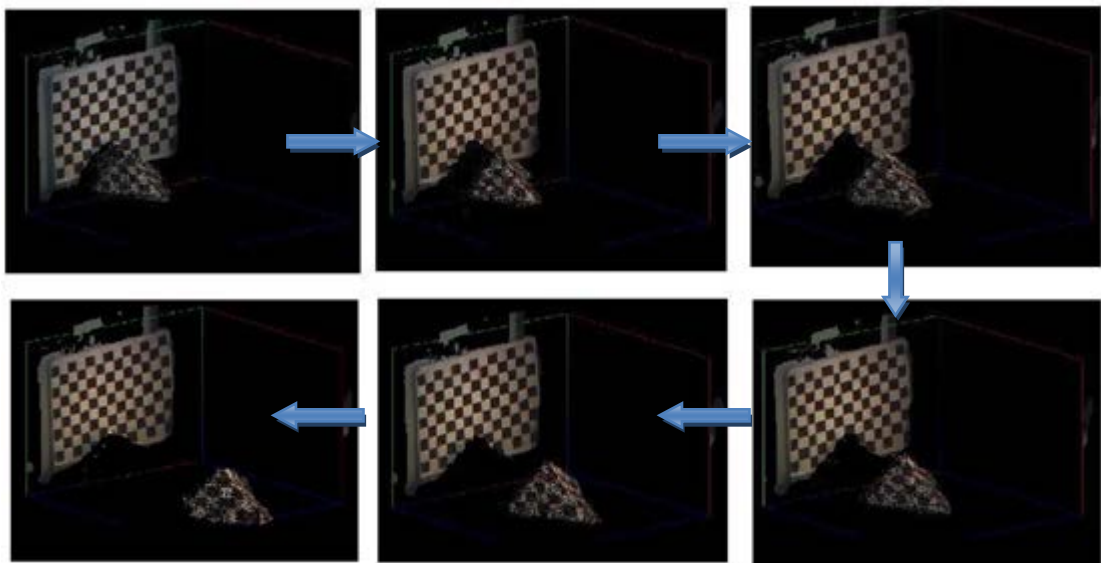
c)

**Fig. 3.23 Reconstrucción tridimensional con texturas con Glut, a) Representación 3D, b) Escenario, c) Mapa de disparidad denso. Fuente propia.**

Para ilustrar mejor la representación 3D se presenta en la Fig. 3.24a una secuencia de imágenes correspondientes a seis marcos de un video, y se realiza la reconstrucción de la escena para cada marco. La representación tridimensional se muestra en la Fig.3.24b. Las flecha rojas indican el orden de la secuencia de video capturada, mientras que las flechas azules indican el orden de la secuencia de reconstrucciones realizadas por el algoritmo. La pirámide se separa físicamente del tablero, y esto se evidencia en la reconstrucción.



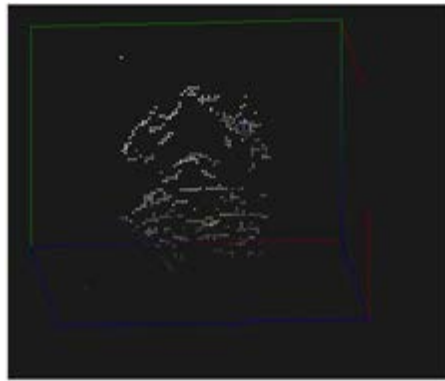
a)



b)

**Fig. 3.24** Reconstrucción tridimensional con texturas con Glut, a) Secuencia de video, b) Reconstrucción 3D de la secuencia de video. Fuente propia.

La reconstrucción de la escena usando disparidad de características consume menos recursos de procesador y produce menos errores que la reconstrucción con disparidad densa. En la Fig. 3.25 se presenta una escena recreada con la técnica estereoscópica partiendo de características relevantes de la escena.



a)



b)



c)



d)

Fig. 3.25 Reconstrucción tridimensional partiendo de disparidad de características con Glut, a) Representación 3D, b) Características relevantes, c) Restricción de unicidad, epipolaridad y de máxima disparidad, d) mapa de disparidad no denso. Fuente propia.

### 3.4 CARACTERIZACIÓN DEL SISTEMA ESTÉREO

Existen varios parámetros que se necesitan determinar en un sistema de visión estereoscópico. Estos parámetros son, entre otros, la relación real entre la disparidad y la profundidad, cuya ecuación matemática puede no ser necesariamente como la descrita en 2.2. Por lo tanto es necesario hacer un ajuste entre datos de calibración y datos teóricos. Las distancias mínima y máxima de profundidad acotan el rango del sistema estéreo. Este rango dependerá del número de disparidades permitidas y del *baseline*. Además, se debe determinar la resolución del instrumento, para determinar el error en las medidas. Finalmente es necesario evaluar el coste computacional de los algoritmos de visión estereoscópica.



### 3.4.1 Profundidad vs. disparidad

A pesar de la alineación *off-line* y la rectificación de imágenes, es difícil obtener un sistema de visión estéreo perfectamente alineado. Por esta razón, en la práctica, los ejes ópticos no son paralelos, y por lo tanto la expresión 2.2 no aplica. Sin embargo, el sistema estéreo puede modelarse como un sistema de cámaras no paralelas, en particular afectado por una rotación en el eje Y. El análisis de este efecto en el sistema estéreo se mostró en la sección 2.2.1.

El soporte del par estéreo permite ubicar las cámaras a distancia *baseline* de 4cm y 6cm. Para cada distancia *baseline* se obtuvo la relación experimental entre la profundidad y la disparidad. En la Fig. 3.26 se muestra la curva con *baseline* de 40mm y en la Fig. 3.27 se presenta la curva con *baseline* de 60mm.

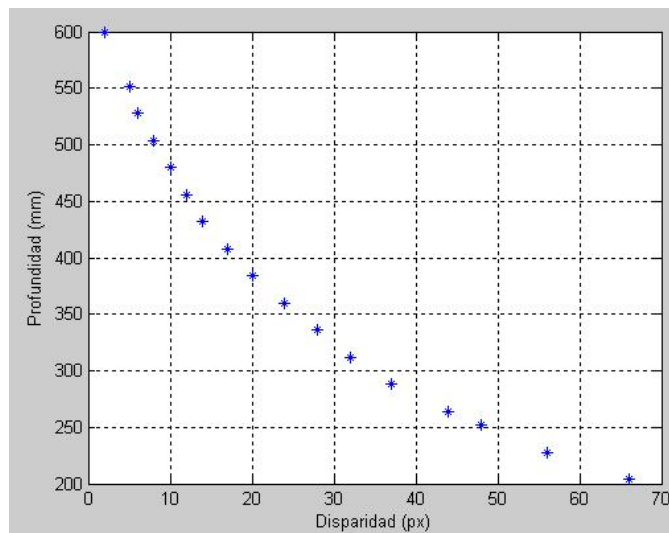


Fig. 3.26 Profundidad vs Disparidad, baseline=40mm. Fuente propia.



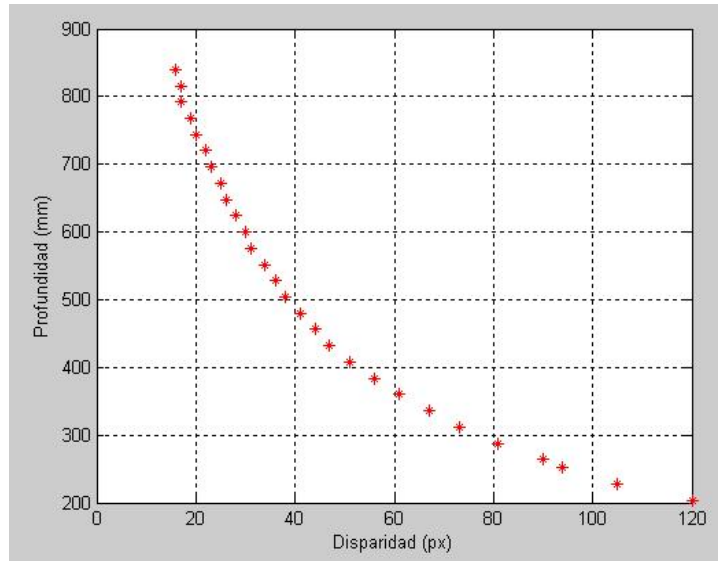


Fig. 3.27 Profundidad vs Disparidad, *baseline*=60mm. Fuente propia.

Al examinar la Fig. 3.26 y Fig.3.27 se puede notar que existe un corrimiento de la curva hacia la derecha conforme aumenta la distancia *baseline*, lo que se corrobora con la literatura consultada [29]. Además se puede confirmar que para una profundidad determinada, el valor de disparidad es mayor para el sistema con mayor *baseline*. En la Fig. 3.28 se puede ver el comportamiento descrito.

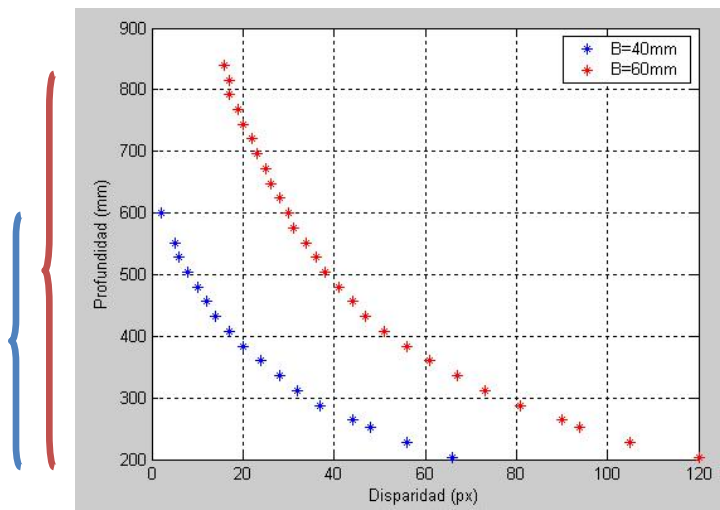
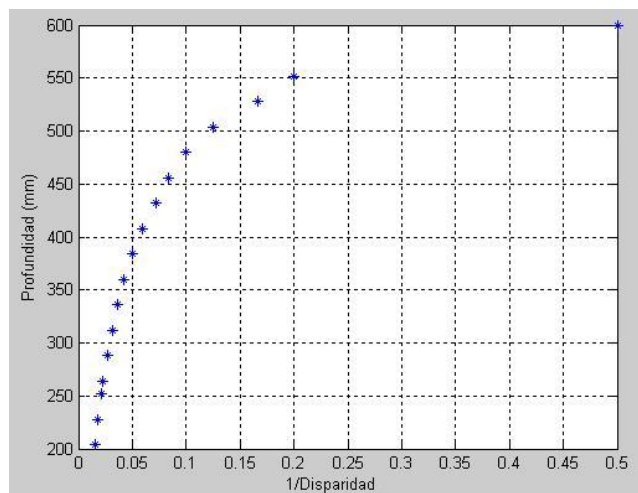
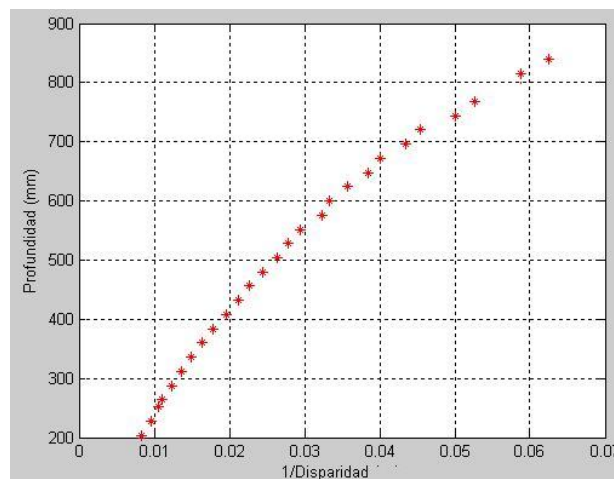


Fig. 3.28 Profundidad vs Disparidad. Fuente propia.

Ahora se pretende determinar cómo está alineado el sistema estéreo. Si sus ejes ópticos están perfectamente alineados, graficar la profundidad contra el inverso de la disparidad debería producir una línea recta con pendiente  $f \times B$ . Si este no es el caso, como se ve en las Fig. 3.29 y Fig.3.30 con *baselines* de 4 cm y 6 cm respectivamente, se debe determinar qué tipo de movimiento en las cámaras está afectando la alineación. Como se esperaba, los ejes ópticos no están perfectamente alineados. El efecto más probable es el producido por una rotación sobre el eje Y de las cámaras [17]. Si esto se cumple, entonces la disparidad tendrá una relación lineal con  $\frac{(Z_0 - Z)}{(Z \times Z_0)}$ .



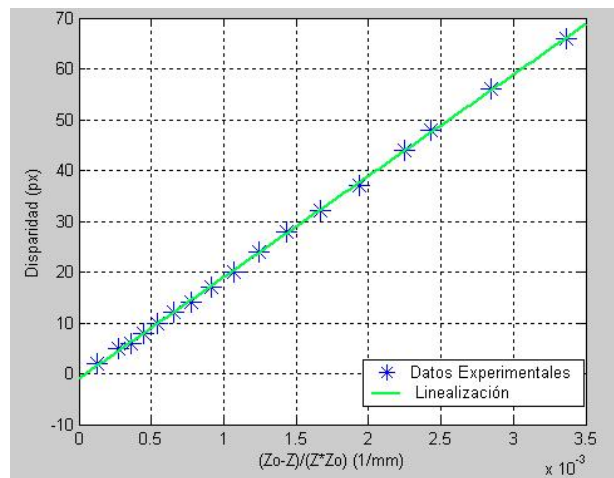
**Fig. 3.29 Profundidad vs 1/Disparidad, baseline=40mm. Fuente propia.**



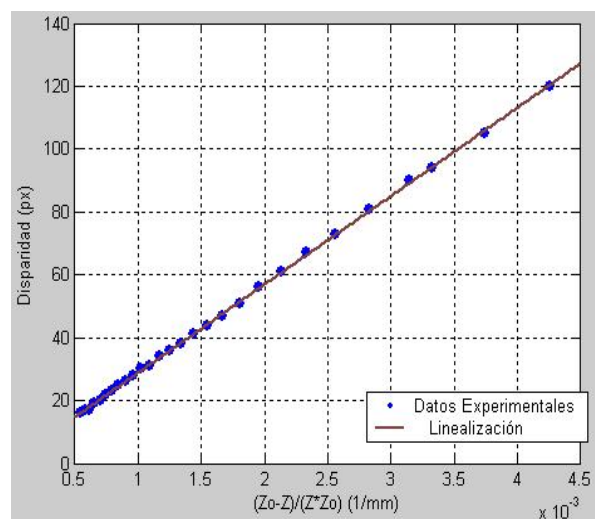
**Fig. 3.30 Profundidad vs 1/Disparidad, baseline=60mm. Fuente propia.**

En la Fig. 3.29 y Fig. 3.30 se presenta la alinealidad en las curvas debida a la no alineación de los ejes ópticos. Se observa que la alineación es más crítica en el sistema con menor *baseline*. Sin embargo al realizar el ajuste descrito por la ecuación 2.4 se obtiene la relación lineal. Se presenta este resultado en las Fig. 3.31 y Fig. 3.32.

El efecto de la rotación de las cámaras en el eje Y solamente afecta la ecuación para la profundidad, los valores de X e Y siguen siendo como se muestran en la ecuación 3.25.



**Fig. 3.31** Linealización, baseline=40mm. Fuente propia.



**Fig. 3.32** Linealización, baseline=60mm. Fuente propia.

La pendiente de las curvas anteriores es igual al producto  $fB$ . En la Tabla 3.1 se muestran los valores calculados de la curva de calibración y los valores arrojados en la calibración de cámara.

**Tabla 3.2** Valores relacionados con la lineación de las curvas en la Fig. 3.28

<b>Baseline (mm)</b>	<b>Pendiente calculada (px.mm)</b>	<b>Pendiente medida (px.mm)</b>	<b>Offset (px)</b>	<b>Error (%)</b>
60	29994.49	29978.63	0.71	0.053
40	19985.75	20000	-0.93	0.071

La pendiente de la curva de linealización aumenta conforme crece la distancia de separación entre las cámaras, como se muestra en Tabla 3.2.

Existe la posibilidad de que al terminar el proceso de calibración y alineación, el valor de  $Z_0$  en la expresión 2.4 sea lo suficientemente grande para considerar que el comportamiento del sistema es ideal, como el que se presenta en 2.2 y la linealización se obtiene graficando profundidad contra el inverso de disparidad.

Las curvas que relacionan las variables profundidad y disparidad cambian si existe un ángulo diferente de cero entre los ejes ópticos de las cámaras. En la Fig. 3.33 se muestra el comportamiento de las curvas al variar el ángulo entre los ejes. Los puntos representan los datos experimentales, y las líneas punteadas representan las curvas ideales según muestra la ecuación 2.4.

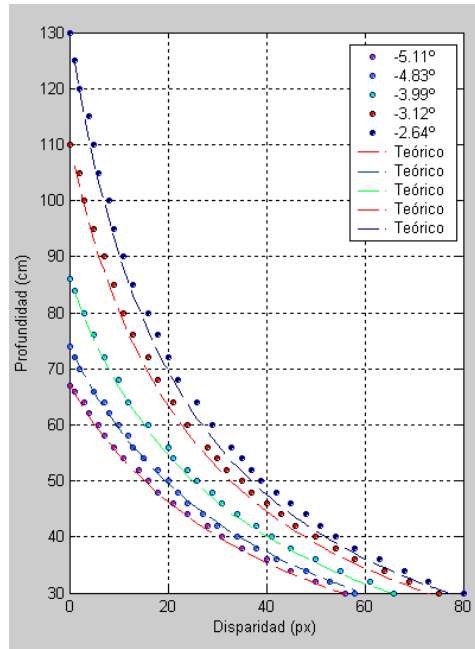


Fig. 3.33 Variación del ángulo de calibración, *baseline*=60mm. Fuente propia.

En la Fig.3.33 se muestra un corrimiento en las curvas hacia la derecha al aumentar el ángulo entre ellas. Este resultado coincide con el encontrado en la literatura consultada [38].

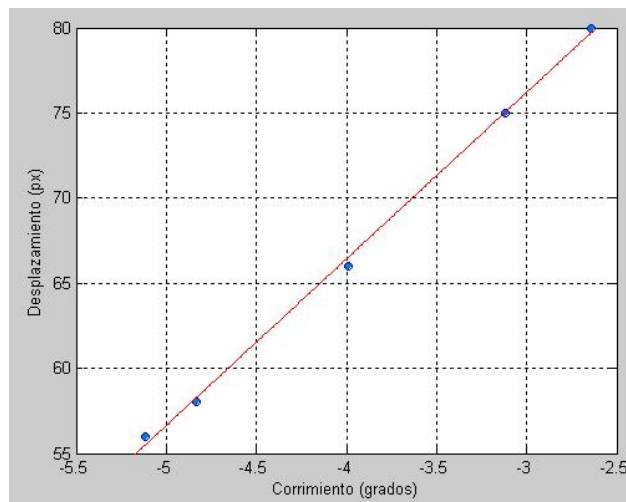


Fig. 3.34 Variación del ángulo entre ejes ópticos, *baseline*=60mm. Fuente propia.

Para determinar si el corrimiento mostrado en la Fig. 3.33 es lineal, se gráfica el desplazamiento de las curvas en relación al ángulo de alineación. Se muestra efectivamente que la relación para ángulos pequeños es lineal, como se puede ver Fig. 3.34, gracias a que las curvas de calibración son paralelas entre sí.

### 3.4.2 Rango

El rango del sistema determina el espacio de trabajo del dispositivo. Se desea determinar en qué intervalo de profundidades trabaja el sistema estéreo. En la Fig. 3.28 los corchetes indican el rango para cada *baseline*. Se puede afirmar que el rango aumenta conforme aumenta la distancia *baseline* entre las cámaras.

### 3.4.3 Sensibilidad

La función sensibilidad para el sistema de visión se define como la pendiente de la curva que relaciona profundidad y disparidad [36]. Si se considera la expresión 2.4 la función sensibilidad estará representada por la ecuación 3.6.

$$S = \frac{\partial Z}{\partial d} = \left| -\frac{Z^2}{f \times B} \right| \quad (3.6)$$

En la Fig. 3.35 se representa la función sensibilidad para las dos distancias *baseline*. La sensibilidad para los dos casos aumenta para lugares lejanos al sistema de visión y disminuye conforme acercamos a él. Esta expresión es equivalente al error absoluto, en la profundidad, del sistema de visión estéreo.

Si por ejemplo, se toma como punto de referencia  $Z=300\text{mm}$ , La sensibilidad para el sistema con 40mm y 60mm es  $S=4,5 \text{ mm/px}$  y  $S=3 \text{ mm/px}$ , respectivamente. Esto significa que una variación de un pixel representa 4,5 mm y 3mm de variación en la profundidad.

De esta forma, para una medida de profundidad determinada se obtiene mayor resolución para el sistema con mayor *baseline* ( $B=60\text{mm}$ ). Esta ventaja se ve compensada por el aumento en la dificultad en el proceso de búsqueda de correspondencia con distancias de *baseline* grandes.

- b) *Baseline* grande: Menor error de profundidad, mayor rango, dificultad para encontrar correspondencias.
- c) *Baseline* pequeño: Fácil búsqueda de correspondencia, menor rango, mayor error de profundidad.

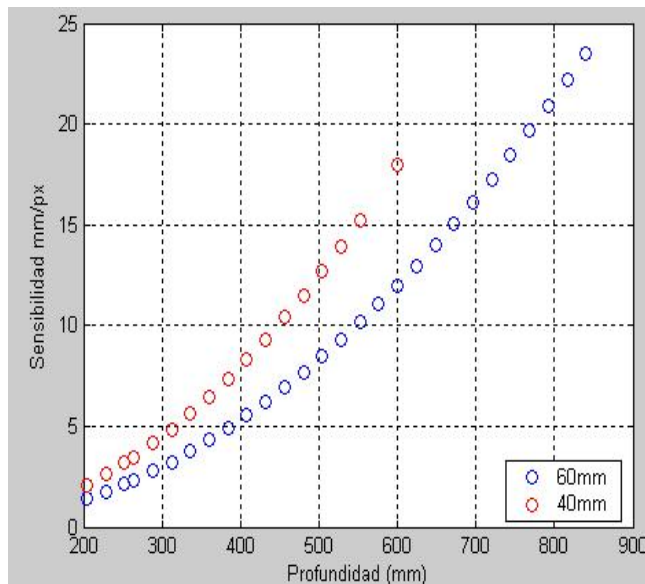


Fig. 3.35 Función sensibilidad. Fuente propia.

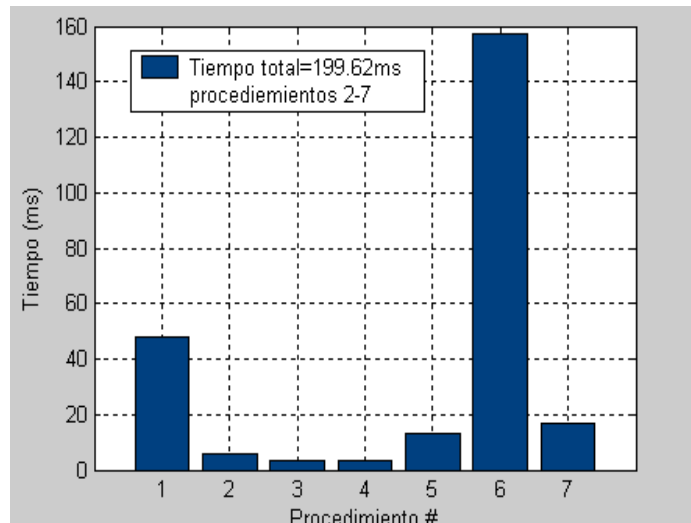
### 3.4.4 Coste Computacional

En un sistema de visión en video es de gran importancia conocer los tiempos que toman los algoritmos en ejecutarse, ya que retardos muy grandes pueden producir respuestas inútiles. Más aún cuando se trata de un sistema binocular que supone que el retardo entre las dos capturas es despreciable. A continuación se detalla los tiempos de respuesta de cada etapa de los algoritmos estéreo, ejecutados sobre un computador Intel Core 2 Duo 2.1GHz y memoria RAM 3GB.

Se exploraron dos algoritmos básicos: el basado en disparidad densa y el basado en disparidad por características. Se detalla el coste computacional de cada uno de ellos.

El algoritmo implementado usando la idea de disparidad por características está conformado por siete procedimientos generales. Estos procedimientos son: 1. Declaración e iniciación de variables, 2. Captura de imágenes, 3. Corrección de distorsiones, 4. Rectificación de imágenes, 5. Extracción de características, 6. Búsqueda de correspondencia 7, Representación 3D.

En la Fig. 3.36 se muestra un diagrama que muestra los tiempos, en milisegundos, del consumo computacional para los procedimientos antes descritos.



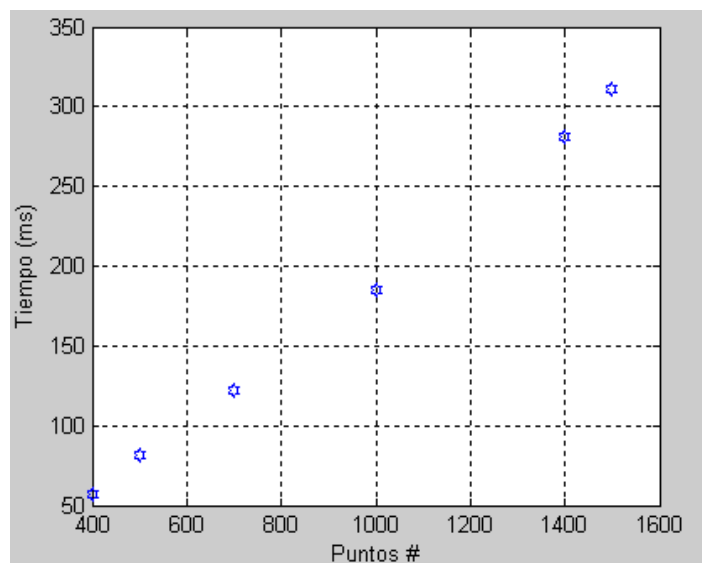
**Fig. 3.36 Coste computacional, método disparidad por extracción de características. Fuente propia.**

El primer procedimiento en la Fig. 3.36 se ejecuta una sola vez cuando se ejecuta el algoritmo. El procedimiento seis hace referencia a la búsqueda de correspondencia, el



tiempo que tarda el algoritmo en realizar esta tarea depende del número de puntos obtenidos en el procedimiento cinco (extracción de características), el valor tomado en el diagrama de barras es un valor promedio.

Se midieron los tiempos del algoritmo en el procedimiento seis con diferentes cantidades de puntos característicos. En un rango entre 400 y 1500 puntos extraídos se obtuvo el comportamiento que muestra la Fig. 3.37.



**Fig. 3.37 Variación del tiempo de procesamiento respecto al número de puntos de búsqueda.**

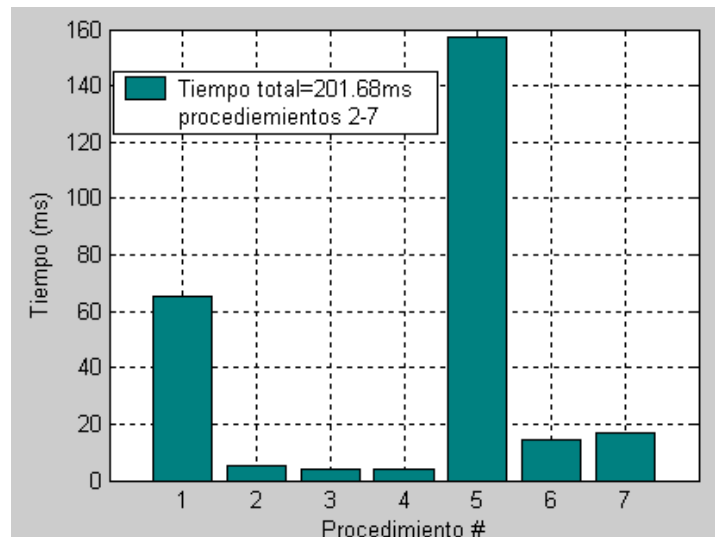
**Fuente propia.**

Evidentemente el tiempo de procesamiento aumenta si el número de puntos de búsqueda crece.

El algoritmo implementado usando la idea de disparidad por regiones está conformado por siete procedimientos generales: 1. Declaración e inicialización de variables, 2. Captura

de imágenes, 3. Corrección de distorsiones, 4. Rectificación de Imágenes, 5. Obtención del mapa de disparidad por SAD, 6. Filtrado, 7. Representación 3D.

En la Fig. 3.38 se muestra un diagrama que muestra los tiempos, en milisegundos, del consumo computacional para los procedimientos antes descritos.



**Fig. 3.38 Coste computacional, método disparidad por regiones. Fuente propia.**

En este caso como en el anterior, el primer procedimiento se ejecuta solamente una vez al ejecutar el algoritmo, y el procedimiento que consume más recursos de procesador es el de búsqueda de correspondencia.

Se debe garantizar además la simultaneidad entre las capturas de las cámaras, es decir que se debe certificar que las imágenes izquierda y derecha del par estéreo corresponden a escenas tomadas en el mismo instante de tiempo. El retardo que existe entre la captura de una cámara y la otra es de 0.006849ms para el primer algoritmo descrito, y 0.006912ms para el segundo. Estos tiempos son despreciables comparados con los del

movimiento de un robot o de cualquier vehículo que soporte el sistema. De esta forma se asegura que las imágenes tomadas son aceptables para los algoritmos de visión estéreo.

En la tabla 3.3 se muestra el tiempo de ejecución total de ejecución de los dos algoritmos y los detalles del computador utilizado.

**Tabla 3.3** Características del computador y tiempos de ejecución. Fuente propia.

<b>Tipo de algoritmo</b>	<b>Procesador</b>	<b>Memoria RAM</b>	<b>Tiempo total (ms)</b>
Reconstrucción 3D usando disparidad por características.	Intel Core 2 Duo 2.1GHz	3GB	199.62
Reconstrucción 3D usando disparidad por regiones.	Intel Core 2 Duo 2.1GHz	3GB	201.68

---

## Capítulo 4

# APLICACIÓN DE LA TÉCNICA DE VISIÓN ESTÉREO A LA CONSTRUCCIÓN DE MAPAS

---

### 4.1 INTRODUCCIÓN

El principal problema de la robótica móvil es la navegación. El campo de la robótica móvil se caracteriza por la aleatoriedad de los entornos reales. La autonomía de navegación es un agente que requiere de capacidades complejas [37].

Para que la navegación sea exitosa, el robot móvil debe localizarse y desplazarse sobre un mapa. De esta forma la localización es un problema estrechamente relacionado con la construcción de mapas. Si un robot está explorando un entorno desconocido y desea mantener información actualizada sobre su posición, entonces es imprescindible que vaya construyendo un mapa de su entorno al mismo tiempo que descubre nuevas zonas. Este problema se conoce comúnmente como *SLAM (Simultaneous Localization and Mapping)* [38].

De lo anterior se puede concluir que para que un robot móvil pueda localizarse y navegar, es necesario que construya un modelo o mapa del entorno, o disponer de uno a *priori*. Uno de los métodos más habituales es la extracción de características. Se puede considerar que dentro de un determinado entorno, una característica es un objeto físico estático, perceptible, y que puede tener una descripción abstracta (borde, esquina, recta etc.). En la mayoría de entornos del mundo real existen características distinguibles que son muy útiles para la navegación, las características que tienen una geometría simple suelen ser una buena elección ya que son relativamente fáciles de detectar y existen en gran cantidad en la mayoría de los entornos.

El experimento de navegación se presenta soportado sobre la plataforma Player/Stage [31], este es un entorno de simulación de aplicaciones robóticas. Una herramienta de código libre que ayuda y simplifica el desarrollo de algoritmos de control y navegación de robots. El entorno proporciona un servidor Player, que ofrece a clientes el acceso a sensores y actuadores y un simulador de robot llamado Stage.

En este capítulo se desarrolla la construcción de mapas usando la técnica de visión estéreo con extracción de características. Debido a que no se cuenta con un robot real, se simulan sus accionamientos en la plataforma Stage mientras se desplaza el sistema de visión de forma manual. En la sección 4.1 se presenta la integración del sistema de visión estéreo con la plataforma robótica Player/Stage. En la sección 4.2 se muestra los resultados de la construcción de mapas en entornos estructurados, simulando los accionamientos del robot en Stage, y realizando la reconstrucción simultánea de las escenas 3D. En la sección 4.3 se presenta el comportamiento del algoritmo para exteriores.

## **4.2 Sistema de visión estéreo y Player/Stage**

Player es un interfaz para dispositivos robóticos. Actúa como una capa de abstracción de Hardware (*Hardware Abstraction Layer HAL*) para dispositivos robóticos. Su filosofía es similar a la de un sistema operativo (Linux, Mac OS X, etc.), el cual oculta los detalles de hardware del ordenador definiendo unos conceptos genéricos como son el "ratón", y la "impresora", cada uno con un interfaz propio. Player tiene el mismo objetivo para dispositivos robóticos, se puede considerar un sistema operativo para robots. Player define una serie de interfaces estándar, cada uno de los cuales es una especificación de las formas en las que se puede interactuar con alguna clase de dispositivos. Por ejemplo la interfaz *interface\_position2d* es usado por los robots móviles, permitiéndoles aceptar comandos que les hacen moverse (con consignas en velocidad o en posición) y devuelve

su estado (la velocidad y la posición actual). El trabajo del controlador es hacer que el robot soporte a una interfaz definida como estándar. Así el código de control escrito para Player que es usado en un robot podrá funcionar (dentro de unos límites razonables) en otro robot.

Player también pone a disposición de los usuarios unos mecanismos de transporte que permiten a los datos ser intercambiados entre los controladores y los programas de control que están siendo ejecutados en máquinas distintas. Con gran diferencia el método de transporte más común en uso hoy en día es un transporte cliente/servidor basado en sockets TCP.

Stage es un simulador para aplicaciones robóticas. Proporciona un mundo virtual para robots móviles, sensores y varios objetos que los robots pueden detectar y manipular. Este programa es normalmente usado junto con el servidor Player para servir como mundo simulado a los programas de control. Aunque puede ser usado también como biblioteca de simulación.

Stage es capaz de simular una población de robots móviles, sensores y objetos del entorno. Todos los sensores y actuadores son accesibles a través de la interfaz estándar de Player. Los clientes no notan ninguna diferencia entre los dispositivos reales y su equivalente simulado en Stage.

En el presente trabajo de investigación se integró el sistema de visión estéreo como un dispositivo de Player, esto quiere decir que los datos arrojados por el sistema de visión estéreo pueden ser usados por cualquier cliente de Player, además el sistema de visión queda sincronizado con todos los dispositivos que se conecten a la plataforma robótica, y puede comunicarse de forma remota por medio del protocolo TCP/IP.

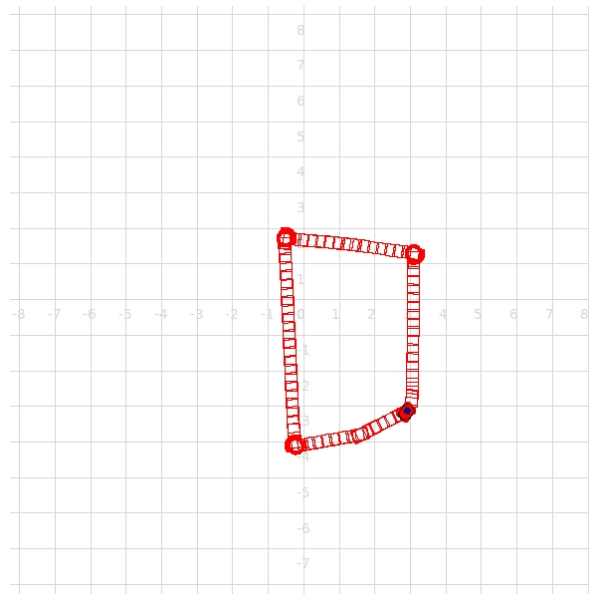
## **4.2 Construcción de mapas en interiores usando visión estereoscópica**

La navegación de vehículos autónomos en interiores es una aplicación de gran interés en el estudio de la robótica móvil. En este trabajo de investigación se diseñó un experimento que permite usar el sistema de visión estéreo en la construcción de mapas, sin embargo, el desarrollo del trabajo se enfoca en la elaboración del sistema sensor de visión, y no en la construcción de un robot móvil. Por esta razón los accionamientos y actuadores del robot se simularon en la plataforma robótica Player/Stage.

Para desarrollar el experimento se efectuaron los pasos de calibración y rectificación de imágenes, además de la pre alineación hardware. Posteriormente se realizó la extracción de características usando la idea de derivada discreta. A los puntos extraídos se les aplicó los criterios de la geometría epipolar, de unicidad y de máxima disparidad. Para los puntos emparejados se encontró la disparidad y usando las ecuaciones de las curvas de calibración se transformaron los valores de disparidad en valores de coordenadas reales en unidades físicas.

El objetivo que se busca en esta aplicación en particular es que el sistema sea capaz de decidir cuándo enviar órdenes a los actuadores del robot para evitar un obstáculo. De todas las profundidades de los puntos importantes de la imagen se debe escoger la más representativa. El criterio utilizado fue el del pico del histograma de los puntos emparejados, que consiste en contar cuantos puntos existen con cada profundidad posible, y tomar el valor de profundidad que mayor número de puntos involucre. Este valor representativo de profundidad se compara con un umbral, por ejemplo 100 cm. Entonces, el algoritmo debe activar los accionadores del motor si el valor de profundidad más representativo es menor que dicho umbral, de esta forma el robot gira hasta esquivar el obstáculo.

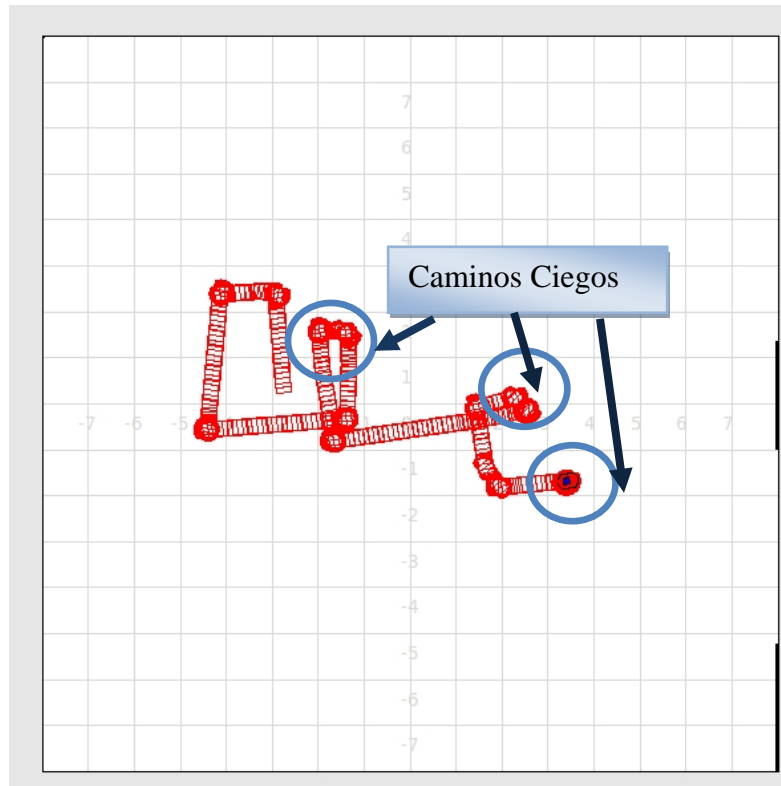
En la Fig. 4.1 se muestra una trayectoria cerrada simple. En este procedimiento se mueve el sistema de visión en línea recta hasta que se encuentra un obstáculo a menor distancia que 100cm, en ese momento el sistema le envía la orden a los accionamientos del motor (simulados en Player), para que el robot gire hacia la izquierda hasta que esquive el obstáculo, cuando esto ocurre, el sistema de visión le da la orden al robot de seguir en línea recta, cuando encuentra un nuevo obstáculo el robot recibe la orden de girar nuevamente hacia la izquierda. Este procedimiento se repite hasta que se termina la trayectoria y se llega al punto de partida. En la Fig. 4.1 la grilla representa el plano XZ del escenario real y los rectángulos rojos representan la trayectoria que ha seguido el sistema de visión.



**Fig. 4.1 Trayectoria cerrada generada en Stage, usando la visión estéreo como sensor detector de obstáculos. Fuente propia.**

El algoritmo propuesto debe estar en la capacidad de detectar obstáculos y buscar una salida, y si no la encuentra de regresarse por el camino de llegada para buscar una nueva trayectoria. El comportamiento del sistema en este caso se muestra en la Fig. 4.2. En esta ocasión se llevó el sistema de visión por lugares en los que no existía otra salida que no fuera el mismo camino de llegada, y se construyó un mapa con bifurcaciones en un entorno estructurado.



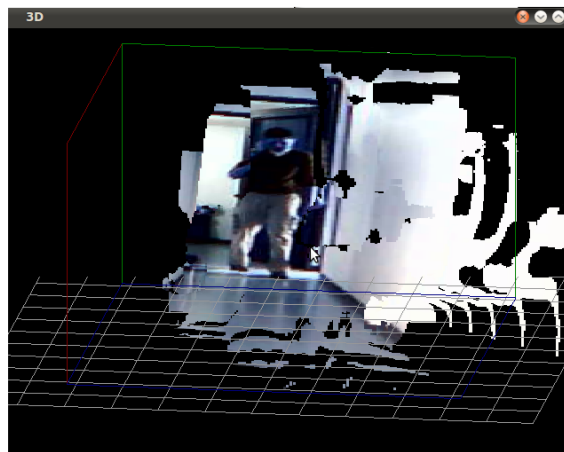


**Fig. 4.2. Trayectoria con bifurcaciones generada en Stage, usando el sistema de visión estéreo como elemento sensor. Fuente propia.**

Debido a que no se conoce con exactitud el desplazamiento de las cámaras, no se puede dar una medida cuantitativa de las dimensiones del mapa reconstruido. Sin embargo, con un sistema de odometría ordinario o visual, se obtendría la construcción del mapa con medidas reales, lo que se presenta es entonces la geometría del lugar navegado por el sistema de visión.

Con la construcción del mapa, se realiza simultáneamente la reconstrucción tridimensional de las imágenes estéreo capturadas en cada marco procesado. En la Fig.4.3 se muestra la reconstrucción tridimensional de un pasillo recorrido por el sistema de visión realizada mediante disparidad por regiones. Cabe resaltar que tanto la construcción

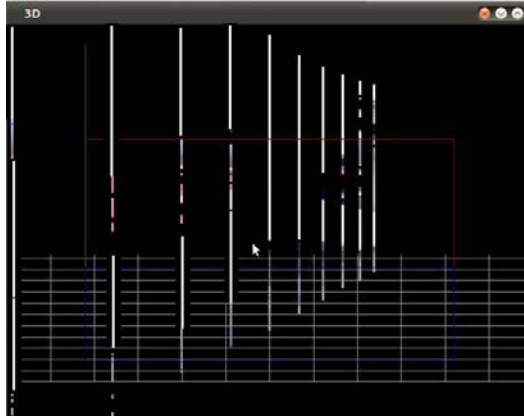
del mapa como la reconstrucción del entorno se realizan simultáneamente en video sobre la plataforma robótica Player en el sistema operativo Linux.



**Fig. 4.3 Reconstrucción tridimensional con disparidad densa. Fuente propia.**

Existe una limitación en las aplicaciones de el sistema de visión debido a la resolución. Como se mostró en el capítulo 3, la resolución aumenta con la separación de las cámaras, el dispositivo desarrollado en este trabajo de investigación permite ubicar las cámaras a distancias de separación pequeñas, lo cual ocasiona que solamente se obtenga buena resolución para lugares en la cercanía del sistema de visión. De esta forma los valores de profundidad para distancias grandes quedan discretizados en valores separados unos de otros. Esto significa que para varios valores de distancias reales existe un único valor de distancia en la reconstrucción. En la Fig. 4.4 se muestra los planos para profundidades alejadas del sistema de visión, donde se puede ver claramente que para mayor profundidad los planos presentan mayor separación. Este hecho es indeseable si la

aplicación requiere precisión en la reconstrucción o en la obtención de medidas tridimensionales reales.



**Fig. 4.4 Planos de profundidad, poca resolución para lugares con mayor profundidad. Fuente propia.**

Sin embargo se mostró en el capítulo 3, que para objetos cercanos al sistema de visión, la reconstrucción es de alta resolución y se obtiene buen resultado en la discretización de la profundidad. Si se desea, entonces, realizar una aplicación en la que se requiera mayor resolución para distancias más lejanas al par estéreo, se debe separar los centros ópticos de las cámaras hasta lograr los requisitos de resolución deseados para la aplicación.

## **4.2 Respuesta del sistema de visión estéreo en exteriores**

El mecanismo de funcionamiento del sistema para exteriores es básicamente el mismo expuesto en la sección anterior, sin embargo existen algunas diferencias en cuanto a las condiciones en las que se desarrolla el experimento.

En primer lugar, la iluminación en exteriores es mucho más intensa que en interiores, las cámaras utilizadas están diseñadas para dejar pasar a través de sus filtros un poco de radiación infrarroja, esto hecho hace que las cámaras se saturen muy fácilmente y se pierda información representativa para el procesamiento. Sin embargo existe una ventaja en la extracción de características en ambientes no estructurados debido a la gran

cantidad de texturas que se presentan. Se puede apreciar que los algoritmos de extracción de características representativas de la escena obtienen muchos más puntos que en el experimento realizado en interiores.

Si el valor representativo de las características obtenidas es mayor al umbral establecido, el sistema da la orden al robot de continuar con su trayectoria (flecha verde), como se muestra en Fig. 4.5.



**Fig. 4.5** Respuesta del algoritmo ante objetos lejanos. Fuente propia.

Si por el contrario el valor representativo en la obtención de los valores de profundidad es menor al umbral establecido, el sistema le indica a los actuadores que deben hacer girar al robot hasta que el obstáculo este fuera de la trayectoria del dispositivo, en la Fig. 4.6 se muestra un marco en el cual el sistema detecta un obstáculo cercano.



**Fig. 4.6** Respuesta del algoritmo un obstáculo dentro del umbral establecido. Fuente propia.

En exteriores, los algoritmos estéreo pueden tardar más tiempo en ejecutarse. Como se mostró en la Fig. 3.37 el tiempo de ejecución está fuertemente influenciado a la cantidad de puntos característicos de la escena.

## CONCLUSIONES

- Se construyó un sistema hardware de visión estereoscópica que permite alinear físicamente los ejes ópticos de dos cámaras. Este sistema cuenta con la posibilidad de separar los centros ópticos en 4cm y 6cm, además permite ajustar manualmente rotaciones en el eje X y Z, y la rotación en el eje Y se realiza con resolución de 0.0191 grados/paso, para ángulos pequeños.
- Se desarrolló una interfaz práctica de consola que permite: calibrar las dos cámaras izquierda y derecha, corregir las distorsiones geométricas de las cámaras, tomar pares de imágenes para rectificación, rectificar las imágenes y evaluar los resultados de rectificación. La aplicación software convierte un par de cámaras tipo webcam en un sistema estéreo de visión.
- Se logró la reconstrucción de escenarios reales partiendo de mapas de disparidad encontrados por dos métodos: disparidad por regiones y disparidad por características. Usando dos técnicas de búsqueda de correspondencia: SAD y flujo óptico. Se obtuvo tiempos de ejecución aceptables, los cuales permitieron la reconstrucción en línea.
- Se confirmó que el aumento de la distancia base en las cámaras provoca el incremento del rango del sistema estéreo, y la disminución de la distancia entre los centros ópticos de las cámaras favorece la etapa de correspondencia. Además el error absoluto medido en la profundidad aumenta al disminuir la línea base que une las cámaras.

- Se confirmó que el error en la profundidad del sistema de visión estéreo aumenta proporcionalmente al cuadrado de la profundidad. La resolución es mayor para lugares cercanos a las cámaras y aumenta a medida que crece la distancia de separación entre los centros ópticos de las cámaras.
- Se analizó el comportamiento de la curva de calibración con la variación del ángulo de alineación, producto de la rotación de las cámaras sobre el eje  $y$ . Se encontró que la curva sufre un corrimiento hacia mayores valores de disparidad cuando el ángulo de alineación separa los ejes ópticos, además el corrimiento de las curvas tiene un comportamiento lineal, para ángulos pequeños, como consecuencia de la linealidad las curvas generadas son paralelas entre sí.
- El sistema estéreo fue implementado completamente en software libre. Además el sistema fue integrado con la plataforma robótica Player, y con el simulador en 2D Stage. Se configuró el sistema estéreo como un dispositivo de Player permitiéndole a uno varios clientes o robots utilizar la información estéreo mediante el protocolo TCP/IP.
- Los tiempos de ejecución para la reconstrucción, utilizando extracción de características están condicionados a la cantidad de puntos que resulten del proceso de búsqueda de característica. En general se encontró que los algoritmos tardan más tiempo en ejecutarse en aplicaciones en exteriores, debido a que la presencia de más textura genera, generalmente, más puntos de interés en la imagen.
- Se encontró que integrar las cámaras estéreo como un dispositivo de Player provoca un aumento en el tiempo de captura de las cámaras, el cual puede

impedir la implementación del sistema en una aplicación real. Esta afirmación está sujeta a los recursos computacionales usados.

- Se construyeron mapas en 2D de entornos reales, usando como único elemento sensor el sistema de visión estereoscópico, utilizando como herramienta de simulación de robots Player/Stage para construir la trayectoria recorrida por el sistema estéreo.



---

## TRABAJOS FUTUROS

---

1. Añadir una pieza extra al soporte de las cámaras que permita ampliar la distancia de *baseline*, para que el sistema de visión pueda tener aplicación en objetivos lejanos, conservando una resolución aceptable.
2. Desarrollar algoritmos que permitan abordar el problema de oclusión, con el fin de evitar falsas correspondencias y mapas de disparidad erróneos. Además, desarrollar algoritmos que permitan reducir los efectos de la variación de la iluminación en la escena sobre superficies especulares.
3. Verificar el funcionamiento del sistema de visión estereoscópico como un dispositivo de la plataforma robótica Player sobre un robot real.
4. Integrar un sistema de iluminación artificial al dispositivo de visión estéreo, para generar texturas artificiales usando luz estructurada, con el propósito de mejorar los resultados en los algoritmos de emparejamiento sobre escenarios de poca textura.
5. Desarrollar algoritmos que permitan encontrar la posición real del par estereoscópico en un escenario real, usando la técnica de odometría visual, con el fin de construir mapas en 3D.

---

## REFERENCIAS

---

- [1] E. Krotkov, M. Hebert, R. Simmon. "Stereo Perception and Dead Reckoning for a Prototype Lunar Rover", Robotics Institute, Carnegie Mellon University.
- [2] J. L. Lotti, G. Giraudon, "Correlation algorithm with adaptive window for aerial image in stereo vision". Institut National de Recherche en Informatique et Automatique.
- [3] M. Groeger, K. Arbter, G. Hirzinger. "Motion Tracking for Minimally Invasive Robotic Surgery". Institut of Robotic and Mechatronics, German Aerospace Center.
- [4] L. Bartolini, A. Bordone, R. Fantoni, M. Ferri de Collibus, G. Fornetti, C. Moriconi, C. Poggi. "Development of a Laser Range Finder for the Antarctic Plateau". Division of Applied Physics, ENEA, C.R. Frascati. Division of Robotics and Informatics, ENEA, C.R. Casaccia, Rome, Italy (2000).
- [5] Tsai, Y. Roger. "Synopsis Recent Progress on Camera Calibration for 3D Machine Vision", The Robotics Review 1, Oussama Khatib, John J. Craig and Tomas Lozano-Perez. (1987).
- [6] F. Remondino, C. Fraser. "Digital Camera Calibration Methods Considerations and Comparisons". Institute of Geodesy and Photogrammetry, ETH Zurich, Switzerland. Department of Geomatics, University of Melbourne, Australia.
- [7] D. Pizarro, P. Campos, C. L. Tozzi. "Comparación de técnicas de calibración de cámaras digitales". Universidad Tarapacá (2005).
- [8] A. Y. Yang. "Image Formation and Camera Models". Berkeley 225b (2007).
- [9] J. Heikkilä, O. Silvén, "A Four-step Camera Calibration Procedure with Implicit Image Correction". Infotech Oulu and Department of Electrical Engineering. University Oulu. Fin 90570 Oulu, Finland.
- [10] J. V. Francés. "Design of the Filtering and Sensing Part of the SmartSpectra Camera". Boca Raon, Florida, USA (2005).

- [11] Z. Zhang. "A flexible new technique for camera calibration", IEEE Transactions on Pattern Analysis and Machine Intelligence (2008).
- [12] D. C. Brown, "Close-range camera calibration," Photogrammetric Engineering 37 (1971).
- [13] G. A. Idrobo, C. A. Ramírez. "Diseño e Implementación de un Sistema de Adquisición de Datos para la Representación Virtual Tridimensional del Rostro Humano (TRI\_IDENT)". Universidad del Cagua (2005).
- [14] D. B. Guerrero, L. B. Villaluenga. "Técnicas y Algoritmos para la Adquisición, Transmisión y Visualización de Escenas 3D", IOC- Division de Robótica (2008).
- [15] Gonzales R, Woods R. E, "Tratamiento Digital de Imágenes", Addison Wesley Iberoamericana, páginas 74-76, (1996).
- [16] G. Bradski, A. Kaebler. "Learning OpenCV Computer Vision with the OpenCV Library". O'Reilly (2008).
- [17] J. L Yubero. "Técnicas de Tracking y Autocalibración" (2001).
- [18] D. Mery, "Visión por computador". Departamento de Ciencia de la Computación. Universidad Católica de Chile. Santiago de Chile (2004).
- [19] R. I. Hartley, "Theory and practice of projective rectification". International Journal of Computer Vision 35 (1998).
- [20] D. Sharstein. "View Synthesis Using Stereo Vision". Computer Science. Berlin Heidelberg (1999).
- [21] J. Canny. "A computational approach to edge detection". IEEE Transactions on Pattern Analysis and Machine Intelligence 8 (1986).
- [22] H. K. Nishihara. "Practical imaging real-time stereo matcher". Optical Engineering. (1998).
- [23] S. T. Barnard, M.A Fischler. "Computational Stereo". SRI International, Menlo Park, California 94025 (1982).
- [24] R. Lane, N. A Thacker, N. L. Seed, "Stretch-correlation as a real-time alternative to feature-based stereo matching algorithms". Image and Vision Computing (1994).

- [25] O. Faugeras, "Real-time correlation-based stereo: algorithm, implementations and applications". INRIA research raport 2013. (1993).
- [26] C. A. Morillas. "Modelos y Herramientas para Sistemas de Rehabilitación Visual". Tesis Doctoral. Dpto Arquitectura y Tecnología de Computadores. Universidad de Granada (2006).
- [27] B. D. Lucas and T. Kanade, An iterative image registration technique with an application to stereo vision, (1981).
- [28] K. P. Horn, B. G. Shunck. " Determining Optical Flow". Artificial Ingelligence Laboratory, Massachusetts Institute of Technology, Cambridge. MA 02139. USA.
- [29] M. A. T. Torriti. "Reconstrucción Confiable de Superficies Usando Rango de Disparidad Adaptativo". Pontificia Universidad Católica de Chile. Escuela de Ingeniería Santiago de Chile (1998).
- [30] S. Mattocchia. "Stereo Vision: Algorithms anda Aplicacions. DEIS. University of Bologna. Italy (2010).
- [31] <http://playerstage.sourceforge.net/doc/Player-2.1.0/player/>
- [32] <http://sourceforge.net/projects/opencvlibrary/files/>
- [33] K. M. Biggs. " Real-Time Mapping Using Stereoscopic Vision Optimization". Departament of Air Force. Air University. Air Force Institute of Technology (2005).
- [34] J. Wang, Z. Hu, H. Lu, K. Uchimura. "Motion Detection and Driving Environment Using U-V Disparity". National Laboratory of Pattern Recognition, Institute of Automation China. Department of Computer Science, Kumamoto University Japan.
- [35] M. Modesti, M. Pedrazzini, L. Canali, E. Destefanis, "Calibration de un Sistema de Visión Estéreo para Navegación de un AVG". Universidad Nacional de Cordoba C. E. F. y N. Dto de Electrónica. Universidad Tecnológica Nacional Cordoba Argentina.
- [36] T. T. Lu, T. H. Chao. "A High Resolution and High Speed 3D Imaging System and Its Aplicacions on ATR". Jet Propulsion Laboratory. California Insitute of Technology.
- [37] J. M. Cañas. "Programación de robots móviles". Universidad Rey Juan Carlos. (2004).

- [38] Phd T. Bailey. "Mobile Robot Localisation and Mapping in Extensive Outdoor Environments". Australian Centre for Field Robotics Department of Aerospace, Mechanical and Mechatronic Engineering The University of Sidney, (2002).