PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ

FACULTAD DE CIENCIAS E INGENIERÍA

DISEÑO DE UN SISTEMA DE EVASIÓN DE OBSTÁCULOS PARA UNA AERONAVE NO TRIPULADA USANDO VISIÓN ESTEREOSCÓPICA

Tesis para optar el Título de **Ingeniero Electrónico**, que presenta el bachiller:

Alvaro Guido Layme Huaquisto

ASESORES: Andrés Flores Espinoza
Jorge Benavides Aspiazu

Lima, Junio de 2015

RESUMEN

En los últimos años, la utilización de unidades aéreas no tripuladas se ha incrementado enormemente debido a la gran cantidad de usos que se les puede dar. El uso de aeronaves autónomas es importante, por ejemplo, para la agricultura, arqueología, seguridad ciudadana, construcciones a gran escala y minería. Para garantizar que el vuelo autónomo de la aeronave se realice sin problemas es necesario que cuente con los medios necesarios para detectar los objetos que se encuentran en su plan de vuelo y evitar una posible colisión. Las aeronaves no tripuladas pueden ser utilizadas en distintos rubros por lo que es necesario proveer la tecnología necesaria para adaptarse a las funciones requeridas.

A mi madre por su apoyo, amor incondicional y enseñanzas durante todos estos años.

A mi padre por sus consejos y comprensión.

A mis asesores Andrés Flores y Jorge Benavides por su confianza, tiempo y enseñanzas brindadas.

A mis hermanos Briam y Omar por ser las personas con quienes podía contar siempre.

A mis abuelos Augusto y Nemesia y mis tíos por sus palabras de aliento y la confianza plena que depositaron en mí.

A Grecia por ser una gran amiga con quien compartí momentos únicos e inolvidables.

A mis compañeros Jorge Rojas, Sammy Cerida, Guillermo Garayar, Hector Flores, Eduardo Bejar, Niels Prieto, Jorge Carpena, Arturo Purizaga, Frank Salvador.

A mis amigos Ángel Pizarro, Diego Salas, Sergio Atavillos, Daniel Vidal, Joel Fernández, Julissa Chávez, Christian Felices, María Luisa Aparcana y Carol Quispe por su amistad incondicional durante todo el transcurso de mi vida universitaria.

¡Muchas gracias a todos!

Índice general

CAPÍTULO 1: Importancia del desarrollo de tecnología para vehículos aéreos no tripulados 1

1.1 Origen e historia de los UAV 1
1.2 Aplicaciones de los UAV 2

CAPÍTULO 2: Marco teórico y estado del arte 5

2.1 Modelo de la cámara 5
2.2 Relación entre distancia y posición de puntos 6
2.3 Geometría epipolar 7
2.4 Traslapo de las cámaras 8
2.5 Estimación de profundidad 10
2.6 Emparejamiento estéreo 11
2.7 Rectificación 13
2.8 Matriz esencial y fundamental 14
2.9 Estado del arte en algoritmos de emparejamiento estéreo 17

CAPÍTULO 3: Diseño del sistema de visión estéreo 23

3.1 OpenCV 23
3.2 Procesamiento de las imágenes 23
3.3 Implementación en OpenCV 25
3.3.1 Encontrar patrón de tablero de ajedrez 26
3.3.2 Calibración estéreo 27
3.3.3 Rectificación estéreo 27
3.3.4 Detección de puntos de interés 28
3.3.5 Emparejamiento de características 29
3.3.6 Matriz Fundamental 30
3.3.7 Reubicación final de los puntos en las imágenes 30
3.3.8 Mapa de disparidad 32

CAPÍTULO 4: Pruebas y resultados 36

4.1 Calibración del sistema estéreo 36
4.1.1 Parámetros intrínsecos 38
4.1.2 Parámetros extrínsecos 39
4.2 Emparejamiento estéreo sobre par rectificado 40
4.3 Rectificación 41
4.4 Emparejamiento estéreo 48
4.5 Post-procesamiento del mapa de disparidad 51

5 Conclusiones 55

6 Recomendaciones 57

7 Bibliografía 58

Índice de ilustraciones

Ilustración 1: Primera aeronave controlada remotamente en la historia (modelo Curtiss F-5L) [4] 2
Ilustración 2: Robot con sistema de visión estéreo 4
Ilustración 3: Modelo del sistema estéreo. 5
Ilustración 4: Proyección de puntos en las cámaras. 6
Ilustración 5: Restricción epipolar. 7
Ilustración 6: Campo de visión de las cámaras 9
Ilustración 7: Modelo geométrico del sistema estéreo. 10
Ilustración 8: Configuración canónica del sistema estéreo. [1] 13
Ilustración 9: Proceso de rectificación [1] 14
Ilustración 10: Traslación y rotación de las cámaras [1] 15
Ilustración 11: Par estéreo Tsukuba. 19
Ilustración 12: *Ground Truth* del par estéreo Tsukuba. 19
Ilustración 13: Algoritmo de infección. 20
Ilustración 14: Algoritmo de suma de diferencias absolutas (SAD). 20
Ilustración 15: Algoritmo de optimización global. 21
Ilustración 16: Algoritmo de filtros adaptativos guiados. 21
Ilustración 17: Diagrama de flujo del procesamiento de las imágenes. 25
Ilustración 18: Reconocimiento de vértices en tablero de ajedrez. 26
Ilustración 19: Reconocimiento de características con SURF. 28
Ilustración 20: Emparejamiento de características con FLANN. 29
Ilustración 21: Par de imágenes no rectificado [6]. 31
Ilustración 22: Par de imágenes rectificado [6]. 32
Ilustración 23: Par de imágenes (izquierda y derecha). 32
Ilustración 24: Ventana de la imagen estéreo para el emparejamiento. 33
Ilustración 25: Disimilitud del algoritmo de Birchfield [2]. 34
Ilustración 26: Imágenes utilizadas para la calibración. 36
Ilustración 27: Reconocimiento de patrones en las imágenes de calibración. 37
Ilustración 28: Mapa de disparidad con algoritmo SGBM. 41
Ilustración 29: Par de imágenes 1. 42
Ilustración 30: Par de imágenes 2. 42
Ilustración 31: Emparejamiento de características en el par de imágenes 1. 43
Ilustración 32: Emparejamiento de características en el par de imágenes 2. 44
Ilustración 33: Emparejamiento de características en un par de imágenes rectificado. 45
Ilustración 34: Resultado de rectificación del par de imágenes 1. 47
Ilustración 35: Resultado de rectificación del par de imágenes 2. 47
Ilustración 36: Resultado del mapa de disparidad para un par de imágenes no rectificado. 48
Ilustración 37: Resultado del mapa de disparidad para el par de imágenes rectificadas 1. 49
Ilustración 38: Resultado del mapa de disparidad para el par de imágenes rectificadas 2. 50
Ilustración 39: Filtrado y umbralización del par de imágenes 1. 51
Ilustración 40: Filtrado y umbralización del par de imágenes 2. 52
Ilustración 41: Pruebas adicionales de la implementación desarrollada. 53

INTRODUCCIÓN

Por sus siglas en inglés, UAV se traduce como vehículo aéreo no tripulado. Por este motivo, UAV hace referencia a cualquier vehículo que pueda volar sin un piloto humano a bordo. Actualmente, los UAV han ganado popularidad y son comúnmente conocidos como drones; estos pueden ser aviones, helicópteros, hexacópteros, etc.

Por su definición, un UAV puede ser controlado de dos maneras: piloto remoto a través de un control inalámbrico y autopiloto electrónico incorporado. La primera opción es bastante simple de implementar; sin embargo, tiene desventajas como las limitaciones en el alcance del manejo remoto y la dificultad para maniobrar estas naves manualmente; es de importancia señalar que en muchos casos como aviones y helicópteros se requiere gran habilidad del piloto para mantener estable el vehículo. Por otro lado, un autopiloto electrónico cuenta con todos los algoritmos de control basados en el modelo aerodinámico del vehículo para garantizar la estabilidad del vuelo; además de su autonomía para completar el plan de vuelvo sin supervisión continua.

Una de las tantas aplicaciones de los UAV es la toma de imágenes o video a una altura considerable en espacios abiertos. La ubicación espacial del UAV se logra con un sistema de navegación inercial (INS) incorporado y un receptor GPS. Este sistema trabaja con un error entre 5 y 10 metros aproximadamente que no resulta significante debido a la amplitud del área cubierta. Sin embargo, para aplicaciones en espacios más limitados y con una alta probabilidad de encontrar obstáculos, el sistema GPS no resulta una opción viable. Por este motivo, el UAV requiere una forma de determinar la posición en que está así como evitar alguna posible colisión que pueda producirse en su plan de vuelo. La visión estereoscópica es una técnica muy útil para dotar al UAV la facultad de estimar la distancia a la que se encuentran los objetos de su entorno. Esta técnica está basada en el mismo principio de la visión humana porque utiliza dos cámaras paralelas dispuestas del mismo modo que los ojos. A partir del desfase generado entre las imágenes y un modelo matemático de las cámaras, se puede determinar la distancia de los objetos que se encuentran frente al UAV. En algunos casos también se utilizan sensores de proximidad para una medición más exacta.

CAPÍTULO 1: Importancia del desarrollo de tecnología para vehículos aéreos no tripulados

1.1 Origen e historia de los UAV

Un UAV puede ser entendido como un vehículo aéreo motorizado reusable y sin personas a bordo que es capaz de transportar cargamentos de distinta naturaleza y, de este modo, poder cumplir con una tarea específica [8]. La carga puede variar desde sensores que serán utilizados para recopilar datos de algún entorno en particular a objetos cuyo transporte sea requerido.

Recientemente los UAV han incrementado su nivel de popularidad debido a las diversas aplicaciones que se están proyectando; principalmente, esto se debe a la tecnología que ha sido mejorada en estos vehículos. Actualmente, el término UAV o dron es reconocido por la mayoría de personas; sin embargo, esto no fue siempre así debido a que el área a la que se restringía el conocimiento de los UAV era a la investigación científica y fuerzas militares. Aunque muchas personas no son conscientes de ello, el desarrollo de los UAV comenzó hace muchísimo tiempo desde las primeras décadas del siglo XX [4].

Del mismo modo que la red de internet, el desarrollo de tecnología para los UAV contó con sus primeras iniciativas en el área militar. Los primeros experimentos datan de la Primera Guerra Mundial con el desarrollo de bombas y torpedos aéreos. En ese momento, dos fueron las principales dificultades que se presentaron. En primer lugar, las personas encargadas tenían dificultad con el despegue y aterrizaje de los UAV; por otro lado, no se lograba estabilizar el UAV mientras se encontraba en pleno vuelo. Con el paso de los años, el desarrollo de los UAV fue mejorando con la llegada de las distintas guerras que se dieron. Entre estas tenemos la Segunda Guerra Mundial, la Guerra Fría y la Guerra del Golfo [4]. Actualmente, existen UAV muy potentes en cuanto a capacidad de carga y estabilidad.

Ilustración 1: Primera aeronave controlada remotamente en la historia (modelo Curtiss F-5L) [4]

1.2 Aplicaciones de los UAV

Los UAV ganaron popularidad últimamente debido a que se ha buscado desarrollar tecnología que pueda resolver problemas más cercanos a las personas, sean estos de tipo social o industrial. Las aplicaciones para los UAV son muy diversas y aún siguen en expansión; entre las más conocidas se puede citar a la agricultura de precisión, fotogrametría, seguridad ciudadana, topografía, etc. Por otro lado, existen también proyectos futuros como la entrega de correspondencia con unidades aéreas no tripuladas.

Entre las aplicaciones de los UAV está la agricultura de precisión. La idea es utilizar un UAV que lleve a bordo una o más cámaras que adquieren información gráfica de los campos de agricultura. Por ejemplo, en Perú, el Centro Internacional de la Papa lleva varios años utilizando tecnología relacionada a los UAV para poder realizar sensado remoto de los campos de cultivo de papa; su principal objetivo es diferenciar dos tipos de papa con imágenes tomadas a gran altura [http://cipotato.org]. En otros casos las imágenes son utilizadas para determinar las zonas de un campo de cultivo donde la calidad de las plantaciones no es la deseada.

Otras aplicaciones más recientes son la arqueología y minería. En el primer caso, los UAV son utilizados para la fotogrametría, la cual consiste en determinar las dimensiones geométricas de un objeto a partir de imágenes tomadas en distintas

perspectivas. Esta aplicación es importante para lograr obtener modelos en tres dimensiones de estructuras valoradas por la arqueología. Por otro lado, en el área de la minería, la corporación minera Barrick muestra un ejemplo del uso de los UAV. Recientemente, esta corporación ha implementado el uso de unidades aéreas no tripuladas en su operación minera Lagunas Norte ubicada en Perú. Los UAV trazan su vuelo sobre los tajos de la mina para tomar imágenes y, a partir de estas, elaborar mapas, planos y otros documentos útiles para el área de Operaciones de Mina [http://barricklatam.com].

Las aplicaciones mencionadas, se realizan con las unidades aéreas ubicadas a gran altura y pueden navegar gracias al uso de su sistema inercial y la señal GPS. Sin embargo, esta tecnología no puede ser utilizada si el UAV se encuentra en el interior de alguna estructura o en zonas urbanas debido a que la señal GPS puede presentar ruido o simplemente no estar disponible [7]. Por este motivo, es necesario que se cuente con un sistema de navegación que prescinda de la localización GPS. Por otro lado también existen aplicaciones que, aunque son realizadas en exteriores, requieren un sistema de reconocimiento del entorno para evitar colisiones con objetos que se encuentren cercanos al plan de vuelo.

Los UAV son utilizados también para inspección de estructuras como construcciones a gran escala, puentes, molinos de viento, antenas o gasoductos. El uso de los UAV simplifica el trabajo que antes se realizaba manualmente y evita posibles accidentes de los operarios puesto que algunas inspecciones son de difícil acceso como en el caso de las antenas o molinos de viento. En estos casos, el vuelo del UAV no se realiza a gran altura por lo que es muy probable que pueda encontrar obstáculos en su camino. Es por este motivo que se requiere que la aeronave, dentro de su plan de vuelo autónomo, pueda detectar obstáculos y modificar el plan de vuelo inicial. Aplicaciones como la seguridad ciudadana o control de tráfico requieren este sistema debido a que en las zonas urbanas las edificaciones o estructuras que estén a la altura del UAV podrían causar una colisión.

Una solución para evitar dichas colisiones es la instalación de un sensor de proximidad que permita detectar algún cuerpo sólido que se encuentre frente al UAV. Sin embargo, estos sensores no dan gran información respecto al entorno en que se encuentra el

UAV; por este motivo, se comenzó a utilizar el procesamiento digital de imágenes para obtener la información que el UAV necesita para evadir los obstáculos. La técnica más común para este problema es el uso de la visión estereoscópica. La visión estereoscópica se basa en el principio bajo el cual funciona la visión humana y permite diferenciar la profundidad a la que se encuentran los objetos. Los detalles de este principio, que se basa en la disparidad, serán explicados en el siguiente capítulo. Es posible también que se combine la visión estéreo con sensores de proximidad para hacer más robusto el sistema de reconocimiento de obstáculos. A continuación se puede observar un ejemplo de sistema estéreo que consta de dos cámaras; estas cumplen la función de los ojos humanos para obtener dos perspectivas ligeramente desfasadas del entorno del robot.

Ilustración 2: Robot con sistema de visión estéreo

Con el paso del tiempo y las nuevas utilidades que se le da a los UAV se presentan nuevas dificultades para ser resueltas. En sus inicios, los grandes problemas eran el despegue, aterrizaje y estabilidad de vuelo que actualmente ya no representan ninguna dificultad. Sin embargo, ahora se tienen nuevas problemáticas para los UAV; la principal es la autonomía de estos además de los sistemas de navegación. Actualmente los UAV son provistos de energía por baterías que usualmente cuentan con un tiempo de autonomía muy limitado. Por este motivo, es necesario desarrollar tecnología y algoritmos que ayuden a resolver los problemas actuales y futuros que se presenten en el uso de los UAV.

Como ya se ha mencionado anteriormente, el objetivo de la tesis es estimar la distancia de los objetos próximos al UAV. Para este fin se usará visión estereoscópica, motivo por el cual es necesario conocer el modo de operación de una cámara, la influencia de los parámetros intrínsecos a la cámara que afectan al propósito buscado y la teoría necesaria para optimizar la búsqueda de las distancias.

2.1 Modelo de la cámara

El modelamiento de la cámara es muy importante para poder establecer más adelante una relación geométrica entre los puntos del mundo real y los puntos en las imágenes tomadas por dicha cámara. En la ilustración 3 se aprecia el modelo de un arreglo de dos cámaras. El lente de la cámara permite la entrada de luz, la cual llega hasta un sensor de luz; actualmente son comúnmente usados los sensores CMOS y CCD. Estos sensores contienen células fotoeléctricas muy pequeñas que registran la imagen a partir de la luz percibida, después esta es almacenada digitalmente en alguna memoria interna. Para este ejemplo se ha tomado un punto cualquiera en el espacio, este punto es proyectado directamente hacia las imágenes a través de la recta formada por el centro de proyección de cada cámara y dicho punto. Finalmente se obtienen dos puntos cada uno en su imagen respectiva. Dependiendo de la forma en que estén dispuestas las cámaras, los puntos X1 y X2 cambiarán de posición. Esta es una de las primeras pistas para lograr la estimación del entorno frente a las cámaras. Un modelo más detallado será presentado en un apartado posterior.

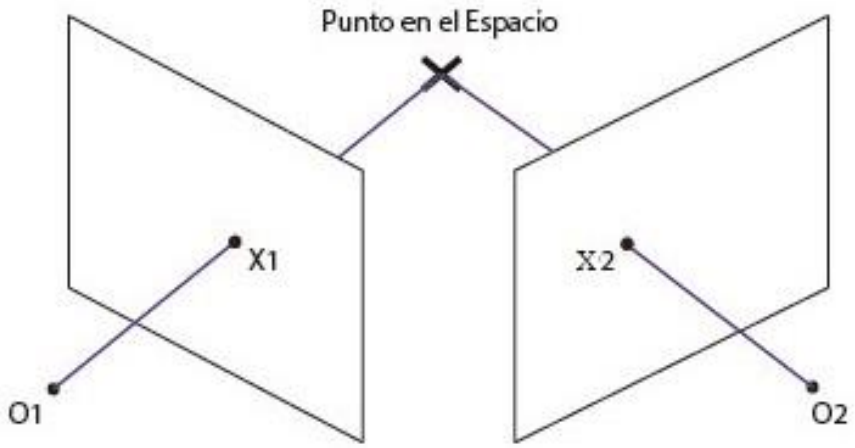


Ilustración 3: Modelo del sistema estéreo.

2.2 Relación entre distancia y posición de puntos

Antes de analizar la geometría del modelo de las cámaras se presenta un ejemplo cualitativo del problema a analizar en la tesis. De este modo, las ecuaciones presentadas más adelante servirán para corroborar el análisis cualitativo de esta sección. El objetivo principal es poder determinar qué tan lejos está un objeto de la cámara. En la ilustración 4 se muestra un arreglo de cámaras en paralelo y dos puntos en el espacio. El punto A es relativamente cercano al arreglo de cámaras, por otro lado el punto B se encuentra bastante alejado. Al ver las proyecciones del punto A, es claramente visible la distinta posición que ocupan en sus imágenes respectivas. El punto A1 se encuentra pegado al borde derecho de la cámara izquierda mientras que el punto A2 es más cercano al borde izquierdo de la imagen derecha. Para el caso del punto B se presenta un caso similar; sin embargo, la diferencia en la posición de los puntos B1 y B2 respecto al borde izquierdo de sus respectivas imágenes es mucho menor al caso de los puntos A1 y A2. Por este motivo, a partir de un análisis simple de dos puntos a distancias distintas, se puede concluir que la distancia es inversamente proporcional a la diferencia en posición de los puntos proyectados.

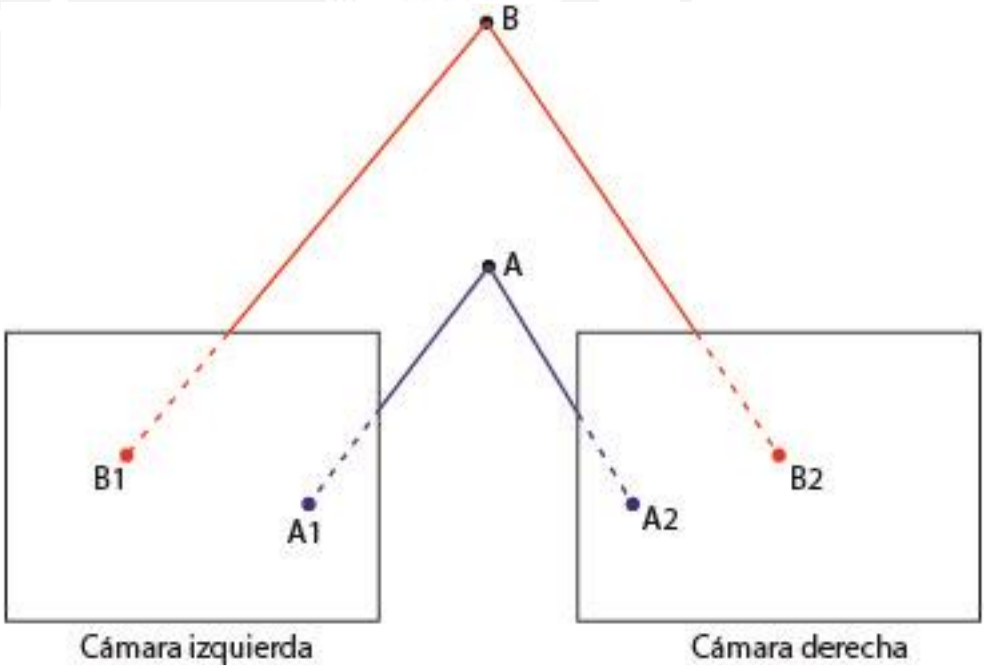


Ilustración 4: Proyección de puntos en las cámaras.

2.3 Geometría epipolar

Por lo desarrollado en la sección anterior, es necesario saber que tan distantes son los puntos proyectados a la imagen respecto a cada una de sus imágenes. Entonces, el primer objetivo es poder ubicar los puntos X1 y X2 que son las proyecciones de un punto X en el espacio. La solución más simple es hacerlo manualmente al determinar un punto en la imagen izquierda y su correspondiente en la imagen derecha, la distancia entre estos es conocida como disparidad. La disparidad entonces nos dice que tan lejos está un objeto, si la disparidad es pequeña el objeto está alejado, en el caso contrario el objeto es más cercano. Sin embargo, este es un proceso que no se puede hacer manualmente porque tomaría demasiado tiempo y se requiere que el proceso sea automático para implementarlo en algún tipo de hardware. Para ubicar un pixel en especial en una imagen de 10x15 se requeriría compararlo con cada pixel de la imagen y ver cuál es el más parecido. Este proceso constaría de 150 comparaciones que, para el tamaño de la imagen, representa una carga de procesamiento excesiva. Por esta razón, es necesario hacer un análisis más exhaustivo del sistema y poder determinar una manera de reducir el rango de búsqueda en la imagen.

La geometría de un sistema estéreo es conocida como Geometría Epipolar. A continuación, en la ilustración 5, se aprecia el modelo de la geometría epipolar y un ejemplo para determinar la Restricción Epipolar. Más adelante se verá que las restricciones son muy importantes para minimizar la cantidad de procesamiento y mejorar la eficiencia del sistema.

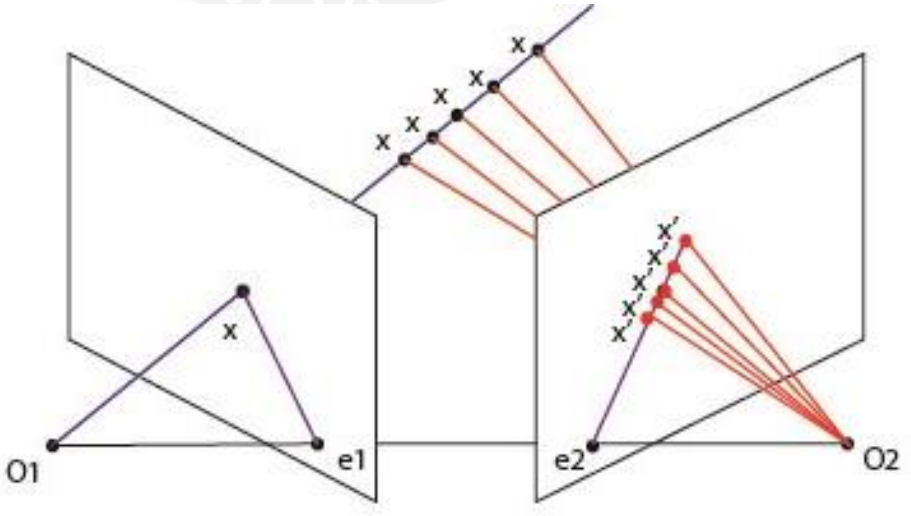


Ilustración 5: Restricción epipolar.

Esta imagen es similar a la presentada en el modelo de la cámara. Sin embargo, el objetivo de esta última es poder determinar donde se encuentra un punto en la cámara derecha si se fija un punto en la cámara izquierda. La línea que une los centros de proyección O1 y O2 intersecta a los planos en los puntos e1 y e2 respectivamente. Estos puntos son llamados los epipolos o puntos epipolares. Estos puntos tienen la particularidad de ser las proyecciones de los puntos O1 y O2 en la imagen correspondiente a la otra cámara. Es decir, e1 es la proyección del punto O2 en la dirección del centro de proyección O1. Una vez fijado el punto x, se traza una línea que pasa por O1 y x la cual contiene también al punto en el espacio correspondiente a x. Sin embargo, solo con una imagen no se puede determinar qué tan alejado está dicho punto debido a que, como se aprecia en la figura, x puede tomar distintas posiciones dentro de la recta que pasa por O1 y x [18].

En este punto es importante notar que existe un plano, llamado plano epipolar, que contiene a O1, O2, x, e1, e2 y todos los posibles puntos en el espacio que corresponden a x. La intersección del plano epipolar con el plano de proyección de la cámara derecha genera la recta que se muestra. Esta recta, como se aprecia en la figura, contiene la proyección de cada posible punto en el espacio de x. Esto significa que, sin importar a qué punto en el espacio corresponde x exactamente, la proyección de dicho punto en la otra imagen debe estar sobre la recta mostrada necesariamente. Este alcance es conocido como la restricción epipolar.

Anteriormente se había mencionado que para encontrar un pixel en una imagen debía buscarse en toda la imagen. Sin embargo, gracias a la restricción epipolar ya no es necesario realizar dicho proceso sino solo buscar en la recta generada por la intersección del plano epipolar y el plano de proyección de la imagen.

2.4 Traslapo de las cámaras

Para la búsqueda de puntos correspondiente ya se pudo establecer una restricción que minimiza notablemente el rango de búsqueda y en consecuencia la cantidad de procesamiento. También es necesaria una restricción sobre los puntos que una cámara puede ver y la otra no, esto con el fin de evitar que la búsqueda genere posibles

errores en los rangos donde no es necesario realizar ninguna búsqueda. En la siguiente figura se aprecia un arreglo de dos cámaras paralelas y sus rangos de visión.

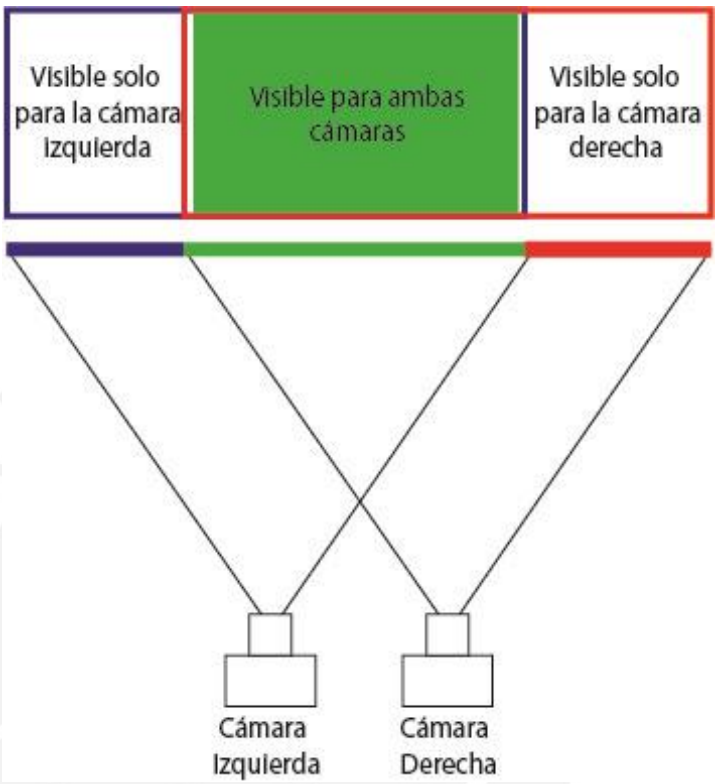


Ilustración 6: Campo de visión de las cámaras

La zona de color azul solo es visible por la cámara izquierda y la zona de color rojo solo visible por la cámara derecha. Por este motivo, cualquier punto en estas dos zonas es imposible de encontrarse en la otra cámara. Todas las correspondencias posibles entre las dos imágenes deben encontrarse necesariamente en la zona verde. Por este motivo se puede afirmar que para la primera correspondencia se encuentra en la primera columna de la imagen derecha y la última correspondencia se encuentra en la última columna de la imagen izquierda.

2.5 Estimación de profundidad

Ya se ha visto que, de algún modo, la disparidad está relacionada con la distancia a la que se encuentra un punto. Sin embargo, este solo fue un análisis cualitativo; por este motivo, ahora se procederá a plantear una ecuación que relacione estos dos parámetros. En la ilustración 7 se aprecia el modelo de dos cámaras paralelas alineadas.

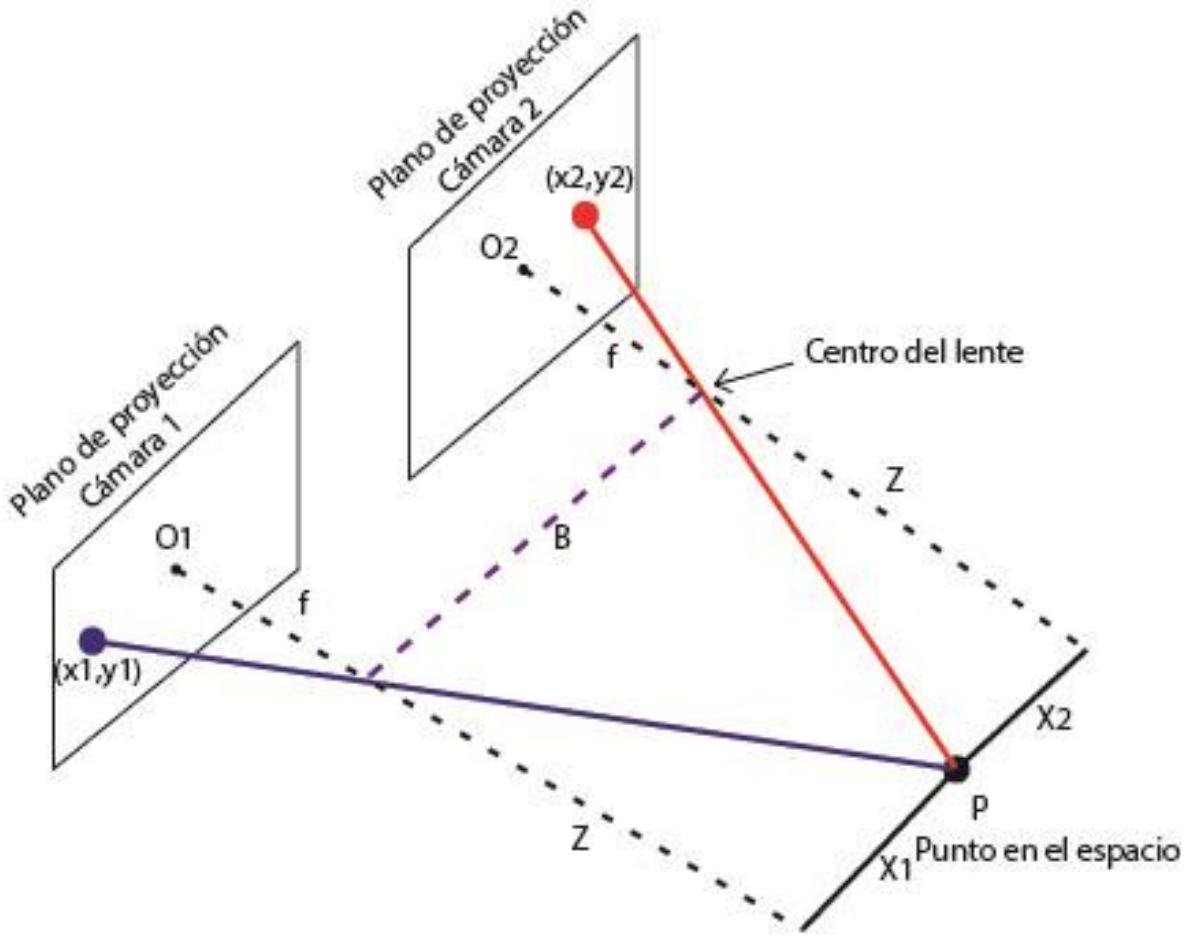


Ilustración 7: Modelo geométrico del sistema estéreo.

En primer lugar, se asume que ambas cámaras están correctamente alineadas; es decir, un mismo punto se encuentra en la misma línea horizontal para ambas imágenes. El centro del lente es el punto de proyección mientras que f se refiere a la

distancia focal y Z a la profundidad del punto P. A continuación se muestran las fórmulas aplicadas al modelo anterior:

Por semejanza:

$$X1 = \frac{x1*Z}{f} \qquad X2 = \frac{x2*Z}{f} \tag{1}$$

Se sabe que:

$$X2 = X1 + B \qquad x2 - x1 = d \tag{2}$$

Entonces:

$$B = \frac{(x2 - x1) * Z}{f} \tag{3}$$

$$Z = \frac{f*B}{d} \tag{4}$$

A partir de un análisis geométrico del sistema se concluye, a través de la última ecuación, que es posible estimar la distancia Z de cualquier punto en el espacio. Debido a que f y B son parámetros intrínsecos al sistema, solo quedaría hallar la disparidad de un punto para el par de imágenes. La disparidad, como se ha detallado anteriormente, es simplemente la distancia que existe entre las proyecciones de un mismo punto para ambas imágenes. Cabe resaltar que el modelo es útil si las cámaras están alineadas paralelamente. Entonces, el procedimiento a realizarse consiste en ubicar ambos puntos de las imágenes y restar sus coordenadas.

2.6 Emparejamiento estéreo

El principal problema del emparejamiento de pixeles en un par de imágenes estéreo es garantizar una alta probabilidad de que ambos puntos correspondan al mismo punto en

el espacio. Sin embargo, esta no es una tarea fácil debido a que las imágenes reales presentan muchos patrones que, de algún modo, dificultan la labor de emparejamiento. Por ejemplo, si tomamos en cuenta una caja totalmente blanca, el primer pixel de la imagen de la caja corresponde al primer pixel de la caja en la otra imagen. Sin embargo, ese mismo primer pixel en la caja blanca es idéntico a todos los demás píxeles que pertenecen a la caja ya que esta es de color blanco enteramente. Este es un problema con porciones de imágenes no texturizadas. Por otro lado, un problema relacionado con la disposición de las cámaras es que, si no están calibradas, no se garantiza que estén exactamente alineadas. Por este motivo, como se vio en la sección de geometría epipolar, habría que ubicar la recta donde se buscará el pixel correspondiente. El trabajo de emparejamiento ya es de por si complejo, por ello se prefiere que una línea horizontal en una imagen corresponda a la misma línea horizontal en la otra imagen. Para garantizar que esto sea exactamente así se requiere transformar las imágenes bajo ciertos parámetros. Este proceso es conocido como rectificación de imágenes estéreo.

Configuración canónica estéreo

Antes de definir el proceso de rectificación y describir los detalles de dicho proceso, se procederá a dar una breve explicación sobre la configuración canónica estéreo. Esta configuración se refiere al modo en que están dispuestas físicamente las dos cámaras que forman parte del sistema estéreo. Un sistema estéreo en configuración canónica consiste, de manera resumida, en dos cámaras que están perfectamente alineadas. En consecuencia, existen dos detalles que merecen gran consideración debido a que estos determinan si se requiere finalmente el proceso de rectificación. Primero, cada línea horizontal en ambas imágenes deben corresponder a los mismos puntos del espacio; es decir, las líneas epipolares deben estar alineadas. Finalmente, los ejes de proyección deben ser paralelos y perpendiculares a la línea base que une los centros de proyección de ambas cámaras; esta condición implica que los puntos epipolares se hallan en el infinito. En la siguiente figura se muestra un sistema estéreo en configuración canónica.

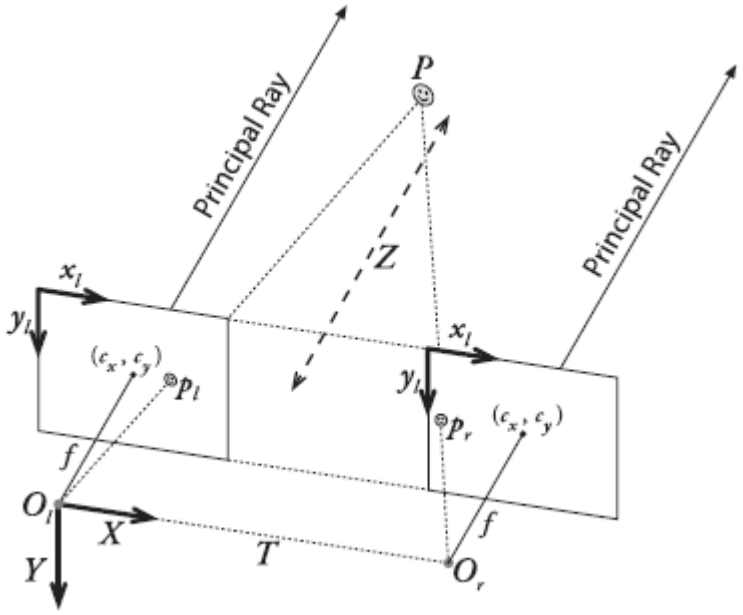


Ilustración 8: Configuración canónica del sistema estéreo. [1]

En esta figura se observa que los ejes formados por los centros de proyección y los centros de los planos de proyección (C_x, C_y) son completamente paralelos y perpendiculares a la línea base formada por los puntos O_l y O_r. Esta observación es muy importante para llegar a la definición de la matriz fundamental, la cual se tratará en detalle más adelante. Asimismo los puntos p_l y $p_r$se hallan en la misma coordenada vertical ypor lo que solo difieren en la coordenada x_l; esta es la base del concepto de disparidad. Esta configuración es muy importante debido a que los algoritmos de emparejamiento, en su gran mayoría, asumen que ambas imágenes estéreo son completamente paralelas. La razón de asumir tal condición es que para efectos de minimización de tiempo y simplificar el emparejamiento se restringe el área de búsqueda a las líneas epipolares horizontales.

2.7 Rectificación

Es muy complicado que un sistema estéreo real pueda estar en configuración canónica debido a que basta una pequeña desviación para que los ejes de proyección no sean paralelos o los centros de proyección no estén a la misma altura. Sin embargo, los algoritmos de emparejamiento consideran que ambas imágenes están rectificadas; es decir, que se han obtenido de un arreglo de cámaras en configuración canónica. Por

este motivo, se requiere de un proceso de rectificación previo al emparejamiento estéreo pues, de lo contrario, no se obtendría un resultado satisfactorio con el algoritmo de emparejamiento.

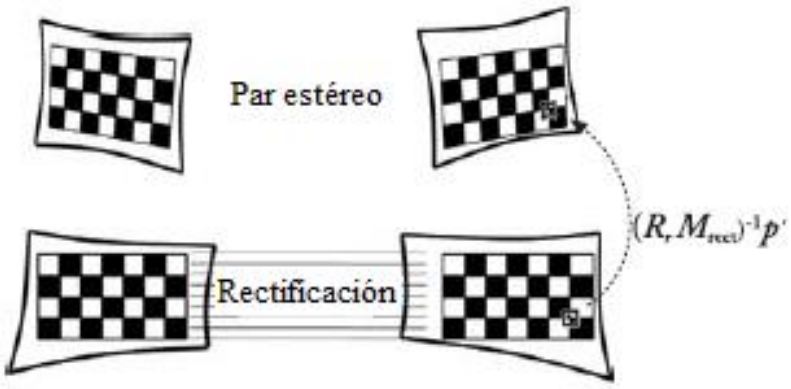


Ilustración 9: Proceso de rectificación [1]

En el primer par estéreo se aprecia que ambas imágenes no han sido proyectadas paralelamente y, probablemente, los centros de proyección de ambas cámaras no estén a la misma altura. Sin embargo este último detalle no es tan relevante porque el proceso de rectificación se encarga de ambas condiciones. En el siguiente par de imágenes estéreo ya rectificadas se puede observar que se encuentran perfectamente alineadas y cada punto característico en la imagen derecha tiene su correspondiente en la imagen izquierda en la misma línea horizontal.

Antes de entrar en detalle sobre los algoritmos de emparejamiento es necesario definir la matriz fundamental, la cual será de gran importancia más adelante en la implementación de la rectificación.

2.8 Matriz esencial y fundamental

Ambas matrices de 3x3 nos dan información sobre la ubicación espacial del arreglo de cámaras. En el caso de la matriz esencial, esta contiene información de traslación y rotación de ambos planos de proyección de las cámaras en coordenadas físicas; es decir, es una relación puramente geométrica. Por otro lado, la matriz fundamental contiene la misma información geométrica de la matriz esencial pero además incluye información de los parámetros intrínsecos del arreglo estéreo [1]. Mayor información acerca de los parámetros intrínsecos será dada en la sección de implementación

donde se hará uso de ellos. A continuación se muestran dos planos no alineados donde se aprecia una rotación denotada por R y una traslación denotada por T. En la implementación R y T son matrices de las cuales se puede servir para implementar una relocalización de puntos (*remap)* de las imágenes y lograr la rectificación.

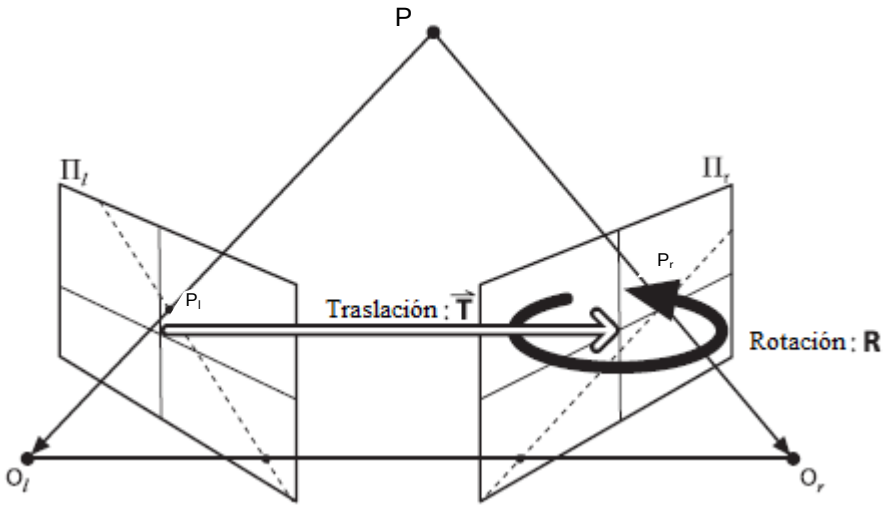


Ilustración 10: Traslación y rotación de las cámaras [1]

Ahora se verá una breve explicación matemática de la matriz esencial y matriz fundamental para entender cómo se relacionan los puntos en ambas imágenes a través de dichas matrices. Dado un arreglo de cámaras estéreo con centros de proyección O_ly O_r, el punto P en el espacio y los puntos observados P_l y P_r que se encuentran en los planos, se procede a tomar como origen al punto O_l. De este modo se tiene queO_r se encuentra en T pues esta es la traslación que las relaciona.

Ahora nos ubicaremos en el plano epipolar formado por los puntos O_l, O_r, P_r, P_l y P. En este punto cabe recordar que el producto de dos vectores perpendiculares es cero. Los vectores T y P_l pertenecen al plano epipolar en cuestión y son formados por la resta de puntos T – O_l, y como este último fue tomado como punto de origen se considera cero. Entonces, el producto vectorial TxP_l es perpendicular al plano. Por otro lado, el plano contiene a los vectores P_l y T. En consecuencia se puede afirmar lo siguiente:

$$(P_l - T)^T (T x P_l) = 0 \qquad (1)$$

El punto P_r está relacionado de la siguiente manera: Pr = $R(P_l - T)$. Entonces se tiene que $(P_l - T) = R^T P_r$. Finalmente la ecuación anterior queda del siguiente modo:

$$(R^T P_r)^T (T \times P_l) = 0 \qquad (2)$$

El producto vectorial $T \times P_l$ se expresará en función de un producto de matrices por lo que es necesario definir una matriz S:

$$T \times P_l = S P_l \quad \Rightarrow \quad S = \begin{bmatrix} 0 & -T_z & T_y \\ T_z & 0 & -T_x \\ -T_y & T_x & 0 \end{bmatrix}$$

De este modo la ecuación resultante es: $(P_r)^T R\, S\, P_l = 0$

Finalmente llegamos a la definición de la matriz esencial como E=RS. Como resultado se obtiene la relación entre un punto en el plano derecho, la matriz esencial y un punto en el plano izquierdo:

$$(P_r)^T E P_l = 0 \qquad (3)$$

Sin embargo, aun esta matriz esencial no es suficiente porque solo contiene información sobre la disposición geométrica de ambas cámaras pero no contiene información alguna sobre las cámaras. Para nuestro caso la información geométrica con coordenadas físicas resulta insuficiente; por este motivo, se requiere hallar la matriz fundamental que opera con coordenadas en pixeles.

Los puntos P_l y P_r son substituidos por los puntos Q_l y Q_r. La relación entre estos puntos está dada por la matriz intrínseca de la cámara (M). Entonces se tiene que:

$$Q = MP \quad \text{o} \quad P = M^{-1}Q \qquad (4)$$

La ecuación de la matriz esencial es ahora escrita en función de los puntos Q en coordenadas de pixeles.

$$Q_r^T (M_r^{-1})^T E M_l^{-1} Q_l = 0 \qquad (5)$$

Finalmente se llega a la definición de la matriz fundamental la cual relaciona la matriz esencial del arreglo estéreo y las matrices intrínsecas de ambas cámaras. Es por este

motivo que se afirma que además de la información contenida en la matriz esencial se obtiene la información de los parámetros intrínsecos de las cámaras:

$$F = (M_r^{-1})^T E M_l^{-1} \quad \text{entonces:} \quad Q_r^T F Q_l = 0 \qquad (6)$$

2.9 Estado del arte en algoritmos de emparejamiento estéreo

El emparejamiento estéreo es el paso fundamental en los sistemas de visión estereoscópica; sin embargo, aunque ha sido vastamente estudiado, aun es un problema sin haber sido resuelto completamente. El emparejamiento estéreo tiene grandes dificultades que debe afrontar y, a pesar de que hay algoritmos capaces de obtener una tasa de error baja, estos consumen muchos recursos de procesamiento. El tiempo requerido por dicho procesamiento no es viable para aplicaciones en tiempo real. Por otro lado, algoritmos clásicos como SAD (suma de diferencias absolutas) o SSD (suma de diferencias al cuadrado), que son lo suficientemente livianos en cuanto a carga de procesamiento pero son muy susceptibles a presentar error en el emparejamiento, son complementados con métricas adicionales para optimizarlos.

El principal motivo por el cual se presentan errores en los últimos algoritmos mencionados es que estos operan solamente sobre el valor numérico de los píxeles. En consecuencia, cualquier clase de ruido, cambio de brillo o contraste ocasionará que el algoritmo no reconozca dos patrones como correspondientes.

Por otro lado, tenemos los algoritmos que, para efectos de estudio, son importantes pero como ya se ha mencionado no son útiles para aplicaciones en tiempo real. En este grupo tenemos, por ejemplo, algoritmos basados en optimización global y *graph cuts* o también basados en probabilidades como lo son los algoritmos basados en propagación de creencia (*belief propagation*) [16].

A pesar de las limitaciones de los algoritmos SAD o SSD, actualmente se prefiere usar estos por las mejoras que han sido implementadas usando información mutua u optimización semiglobal y la velocidad en que pueden ser procesados. En el tercer capítulo se verá que se hace uso del algoritmo modificado de H. Hirschmuller. Este algoritmo además del método basado en SAD aplica información mutua y ecuaciones de energía global para el emparejamiento semiglobal [9]. Sin embargo, el algoritmo de

OpenCV prescinde de la información mutua y utiliza una métrica más simple a nivel subpixel definida en el trabajo de Stan Birchfield y Carlo Tomasi [2]. El trabajo realizado por Birchfield y Tomasi es importante debido a la definición de disimilitud que viene a ser un costo o métrica de emparejamiento y la programación dinámica [2]. Como se puede intuir, para realizar una búsqueda se debe hacer un barrido sobre todas las posibilidades; en consecuencia, es necesario realizar operaciones consecutivas que con el método de recursión no resultaría eficiente. Por este motivo, la programación dinámica es fundamental para la optimización del tiempo de procesamiento. Este tipo de programación se basa en guardar en memoria los resultados que serán utilizados en operaciones posteriores; de este modo, se evita realizar una gran sucesión de operaciones cada vez que se desee hallar un determinado cálculo.

Se ha visto que existen varios métodos para el emparejamiento de puntos en imágenes estéreo; sin embargo, no se han cubierto todos debido a la gran cantidad que existen. El trabajo realizado en el año 2002 por Scharstein y Szelisky en su publicación *A taxonomy and evaluation of dense two-frame stereo correspondence algorithms* presenta un vasto estudio de los algoritmos desarrollados hasta el momento de su publicación [16]. Aunque se han publicado trabajos importantes en los últimos años, es un buen referente para ver los distintos métodos que se pueden utilizar para lograr la correspondencia estéreo. Los autores de este trabajo tienen una base de datos disponible en la red de internet con todos los algoritmos presentados para ser evaluados, ahí se aprecia el resultado de los algoritmos como mapas de disparidad así como el porcentaje de error. A continuación se presentarán algunos de los últimos algoritmos desarrollados en los años 2013 y 2014 para realizar una comparación entre ellos.

Esta página web cuenta con cuatro pares de imágenes estéreo perfectamente rectificadas, los cuales son sometidos al algoritmo desarrollado por algún investigador y a continuación los resultados son publicados. En nuestro caso, solo mostraremos un par de imágenes para la comparación de los algoritmos. Así como la imagen Lena es comúnmente utilizada en algoritmos de procesamiento de imágenes, el par de imágenes estéreo Tsukuba es comúnmente utilizado para ilustrar el resultado de mapas de disparidad. A continuación se muestran ambas imágenes de las cámaras izquierda y derecha respectivamente que fueron obtenidas de la página web de *Middlebury Stereo Evaluation*:

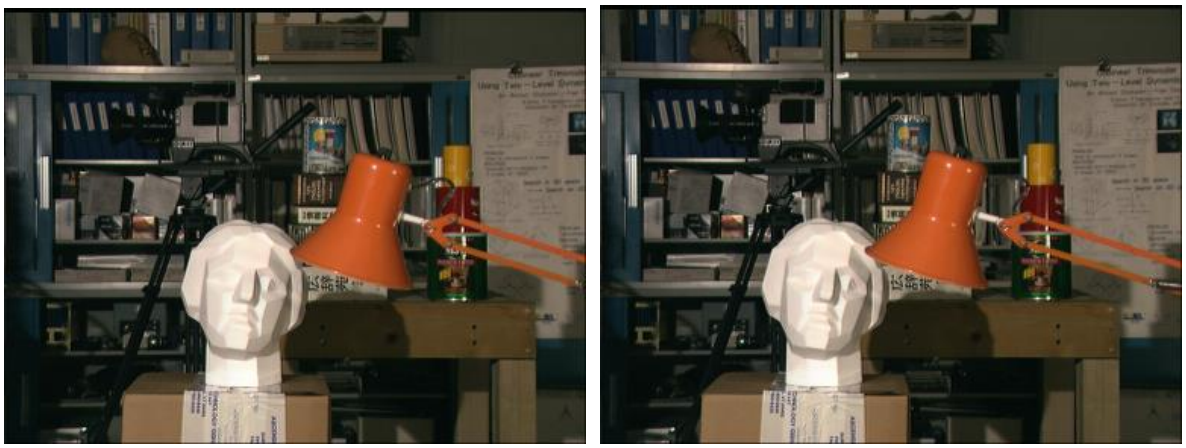


Ilustración 11: Par estéreo Tsukuba.

A continuación se muestra el mapa de disparidad *ground truth* el cual es hallado por métodos especiales o manualmente y su función es determinar el resultado que se espera de los algoritmos evaluados.

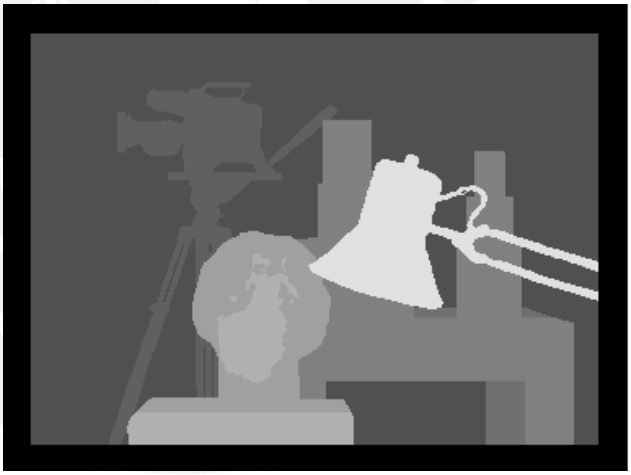


Ilustración 12: *Ground Truth* del par estéreo Tsukuba.

En la anterior figura se aprecia que los principales componentes de la imagen que son la escultura, la lámpara, la mesa, la cámara y el fondo se distinguen completamente por los distintos niveles de gris. Este contraste es precisamente lo que logra la disparidad, el objeto más cercano presenta mayor disparidad por lo que el nivel de gris que lo representa es el más claro mientras que el objeto más alejado presenta menor disparidad y es representado por un gris más oscuro. A partir de esta imagen es posible determinar cuál es el objeto más cercano que se encuentra frente al arreglo de cámaras estéreo.

Ahora pasaremos a revisar cuatro algoritmos almacenados en la base de datos mencionada anteriormente. Primero tenemos un algoritmo publicado en el año 2006 por G. Olague, F. Fernandez, C. Pérez y E. Lutton llamado algoritmo de infección [19]. El resultado obtenido no es claro respecto a los que se revisará más adelante; sin embargo, es útil para ver la diferencia entre algoritmos de gran exactitud respecto a los algoritmos de gran velocidad.

Ilustración 13: Algoritmo de infección.

A continuación se presenta un trabajo basado en el algoritmo SAD presentado en el año 2010 por K. Ambrosch y W. Kubinger [20]. En este caso se puede apreciar mayor nitidez que el algoritmo anterior.

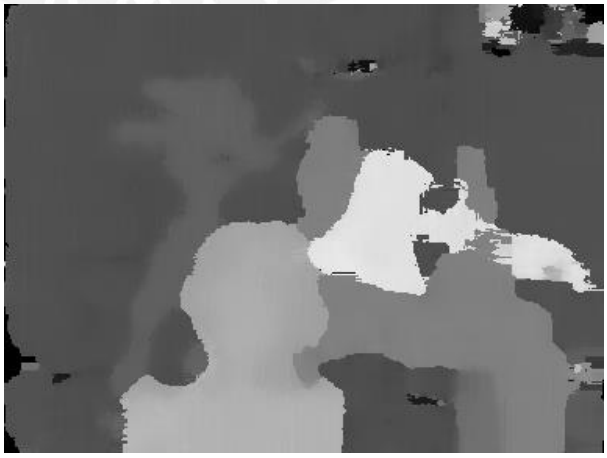


Ilustración 14: Algoritmo de suma de diferencias absolutas (SAD).

El tercer algoritmo fue presentado en el año 2014 por M. Mozerov J. van Weijer basado en optimización global [21]. Este método consiste en considerar todo el conjunto de correspondencias y determinar la posibilidad más óptima en base a algún criterio en específico que plantea el algoritmo.

Ilustración 15: Algoritmo de optimización global.

Finalmente se presenta el trabajo realizado por Q. Yang, P. Ji, D. Li, S. Yao y M. Zhang en el año 2013 [22]. Este algoritmo de emparejamiento estéreo está basado en filtros adaptativos guiados y presenta gran velocidad en el procesamiento.

Ilustración 16: Algoritmo de filtros adaptativos guiados.

De estos cuatro algoritmos presentados observamos que el de infección provee un mapa de disparidad con poca nitidez respecto a los demás. Por otro lado, el algoritmo

basado en emparejamiento de bloques (SAD) es suficiente para identificar las distancias de los objetos de la imagen. Finalmente, los dos últimos algoritmos presentan una claridad mucho mayor en cuanto a la forma de los objetos en las imágenes. Estos son trabajos realizados en los dos últimos años por lo cual forman parte del conjunto de algoritmos más modernos y optimizados hasta la fecha.

Hasta este punto se han visto los conceptos básicos sobre la disparidad así como los mejores algoritmos de correspondencia estéreo respecto a exactitud que hay en el presente. En el siguiente capítulo se verán las funciones en OpenCV que serán necesarias para la implementación del procesamiento de las imágenes. Algunos conceptos que no han sido descritos en el presente capítulo como la calibración de cámaras serán explicados cuando sea necesario.

CAPÍTULO 3: Diseño del sistema de visión estéreo

En el capítulo anterior se desarrolló la teoría necesaria para poder crear el mapa de disparidad así como para entender los conceptos básicos relacionados a un sistema de adquisición de imágenes estéreo. Si bien cada paso del proceso total podría implementarse desde cero, se ha optado por hacer uso de librerías especializadas en procesamiento de imágenes. A continuación se verá una breve descripción de este conjunto de librerías.

3.1 OpenCV

OpenCV es un conjunto de librerías de código abierto especializado en visión por computadora. Cada función implementada en OpenCV ha sido optimizada con el objetivo de cumplir los requerimientos de aplicaciones en tiempo real. OpenCV cuenta con múltiples funciones para imágenes médicas, seguridad, reconocimiento de patrones, visión estéreo, calibración de cámaras, interfaz de usuario, etc. Además puede ser implementado sobre distintos sistemas operativos como Windows, Linux y Mac OS X. Por otro lado, respecto a los lenguajes de programación que soporta están C, C++, Python e inclusive Java. En la presente tesis se hará uso de OpenCV en C++ [1].

3.2 Procesamiento de las imágenes

En esta sección se describirá brevemente el proceso que seguirán las imágenes para obtener el mapa de disparidad final que nos brindará la información sobre la distancia de los objetos visto en las imágenes.

1. Las imágenes serán adquiridas de un arreglo estéreo de cámaras dispuestas de la forma más cercana a la configuración canónica.

2. El siguiente paso consiste en garantizar que las imágenes estén alineadas; por este motivo, se procederá a realizar el proceso de rectificación. Para lograr rectificar las imágenes es necesario realizar cálculo de parámetros a través de la calibración estéreo y la matriz fundamental.

2.1 Calibración de las cámaras: se logra utilizando objetos con patrones distintivos y uniformes. Usualmente se utiliza un tablero de ajedrez no simétrico.

2.2 Ubicación y emparejamiento de puntos característicos en las imágenes con los algoritmos SURF y FLANN respectivamente.

2.3 Hallar los parámetros necesarios de rotación sin distorsión necesarios para la reubicación de los puntos.

2.4 Reubicación de los puntos para obtener las imágenes rectificadas (*remap)*.

La calibración permite obtener parámetros intrínsecos y extrínsecos del arreglo de cámaras. Estos parámetros son útiles para una función implementada en OpenCV que permite generar las matrices que contienen la información sobre la rotación y traslación entre las cámaras. Por otro lado, es necesario obtener la matriz fundamental; esta es hallada con una función especial de OpenCV que requiere conocer como mínimo 8 parejas de puntos característicos ubicados en ambas imágenes. Como resultado se obtienen las matrices necesarias para el *remap* o reubicación de puntos que generará las imágenes rectificadas.

3. Finalmente se procede a someter las imágenes rectificadas al algoritmo de correspondencia estéreo para obtener el mapa de disparidad. En esta etapa se incluyen dos procesos de filtrado realizados antes y después del algoritmo de correspondencia.

En la siguiente ilustración se tiene el diagrama de flujo que resume todo el proceso que seguirán tanto las imágenes de calibración como los pares de imágenes para la estimación de profundidad.

Ilustración 17: Diagrama de flujo del procesamiento de las imágenes.

3.3 Implementación en OpenCV

En esta sección se desarrollarán las funciones que serán utilizadas en el proceso completo de la tesis. Se mostrará la sintaxis, explicará el funcionamiento y detallará la utilidad que tiene en el proceso completo.

3.3.1 Encontrar patrón de tablero de ajedrez.

La función en OpenCV que analiza las imágenes y determina si se encuentran los vértices del tablero de ajedrez es ***findChessboardCorners*** y su sintaxis es la siguiente:

bool **findChessboardCorners** (Imagen de entrada, Tamaño del patrón, Arreglo de salida, banderas)

Como se puede apreciar, la función devuelve un valor booleano que indica verdadero en caso se encuentre el patrón del tablero de ajedrez determinado por el tamaño de patrón. Como salida se obtiene un arreglo que contiene los vértices del tablero de ajedrez detectados. Cabe señalar que la ubicación de estos vértices es aproximado, por esta razón se utiliza una función adicional para determinar con mayor exactitud su ubicación:

Void **cornerSubPix (**Imagen de entrada, Arreglo de vértices, tamaño de ventana, tamaño de zona muerta, criterio de proceso iterativo)

Esta función recibe como entrada el arreglo de vértices, lo modifica en función al proceso iterativo para refinamiento de vértices respectivo y lo entrega como salida. El resultado esperado del uso de ambas funciones es el siguiente:

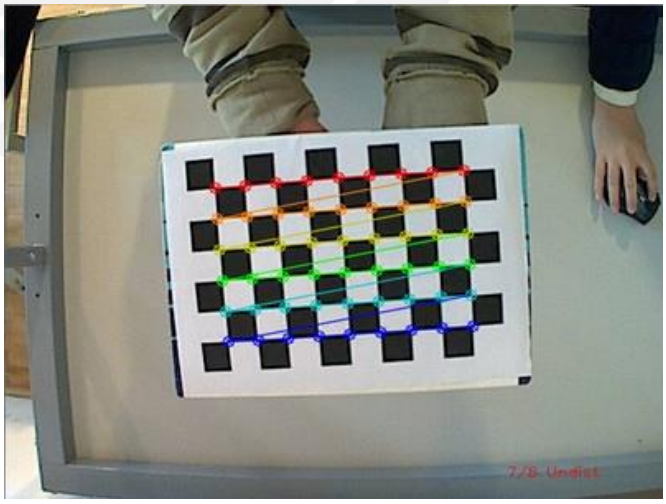


Ilustración 18: Reconocimiento de vértices en tablero de ajedrez.

En dicha figura se aprecia que se ubican todos los vértices internos del tablero de ajedrez. Más adelante la información obtenida de distintas imágenes del tablero de ajedrez en distintas perspectivas es útil para realizar la calibración estéreo.

3.3.2 Calibración estéreo

A continuación son almacenados en un vector todos los puntos de objeto obtenidos en la fase previa; es decir, lo vértices hallados. La sintaxis de la función a utilizarse es la siguiente:

> Double**StereoCalibrate** (Arreglo de puntos de objeto, Arreglo de puntos de imagen 1, Arreglo de puntos de imagen 2, Matriz de la cámara 1, Coeficientes de distorsión 1, Matriz de la cámara 2, Coeficientes de distorsión 2, tamaño de la imagen, Matriz de rotación, Matriz de traslación, Matriz Esencial, Matriz fundamental, Criterio de optimización iterativa, banderas)

Esta función nos permite obtener matrices importantes en el proceso de rectificación; estas son las matrices de las cámaras y los coeficientes de distorsión correspondientes a cada cámara. Además se obtienen como salidas las matrices R, T, E y F que son las matrices de rotación, traslación, esencial y fundamental respectivamente. La función devuelve el valor del error de re-proyección.

3.3.3 Rectificación estéreo

Aunque el nombre de la función puede sugerir que la rectificación se realiza en este paso, la función de OpenCV *StereoRectify*solo provee matrices útiles para la rectificación final. A continuación se presenta la sintaxis de dicha función:

> void **StereoRectify** (Matriz de la cámara 1, Coeficientes de distorsión 1, Matriz de la cámara 2, Coeficientes de distorsión 2, Tamaño de imagen, Matriz de rotación, Matriz de traslación, Matriz R1, Matriz R2, Matriz P1, Matriz P2, Matriz Q, Banderas, parámetro de escalamiento, Nuevo tamaño de imagen)

Esta función recibe como entrada las matrices halladas por la función *StereoCalibrate* y genera las matrices de interés R1, R2, P1 y P2. En primer lugar, se tienen las matrices R1 y R2 de 3x3 de transformación para la rectificación

relacionadas con la rotación. Por otro lado las matrices P1 y P2 de 3x4 son matrices de proyección en el nuevo sistema coordenado generado para la rectificación en cada cámara.

Hasta este punto se han revisado las principales funciones para obtener las matrices esenciales para la rectificación. A continuación se revisarán las funciones para obtener la matriz fundamental a partir del emparejamiento de puntos de interés y la reubicación de los puntos de cada imagen.

3.3.4 Detección de puntos de interés

Para le detección de puntos característicos en ambas imágenes se hará uso del algoritmo SURF (*speeded up robust features*) [10]. Este algoritmo es una mejora del algoritmo SIFT (*scale-invariant feature transform*) y nos permite obtener puntos especiales de una imagen como esquinas, bordes, cambios de color, etc. Además de ello, también se puede obtener un descriptor por cada punto encontrado. Un descriptor es lo que hace único a un punto pues constituye el espacio que rodea a cada punto encontrado. En la siguiente figura se puede observar el funcionamiento del algoritmo SURF:

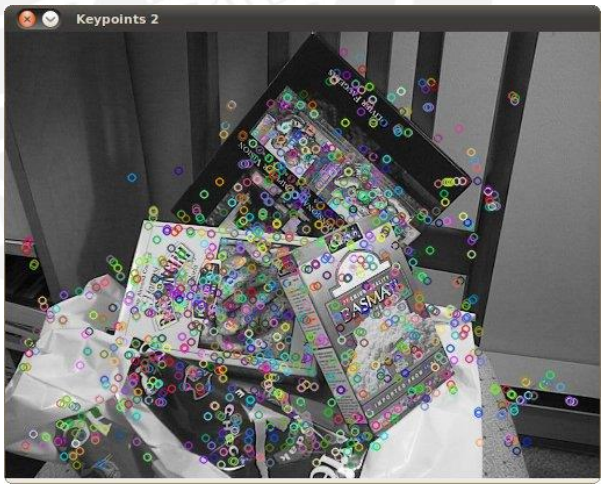


Ilustración 19: Reconocimiento de características con SURF.

Esta función, como se observa, permite ubicar ciertos puntos especiales dentro de una imagen. Sin embargo, en la presente tesis se utilizan imágenes por pares; por

este motivo, es necesario ubicar los puntos en ambas imágenes y determinar cuáles son correspondientes entre sí.

3.3.5 Emparejamiento de características

Una vez encontrados los puntos característicos en ambas imágenes es necesario saber que pares de puntos corresponden a la misma característica; por este motivo, de un modo parecido al emparejamiento estéreo, se procede a realizar el emparejamiento de características. Para lograr el emparejamiento se hace uso de la función implementa en OpenCV denominada *FLANN Matcher* que,por sus siglas en inglés, significa búsqueda rápida aproximada del vecino más cercano [http://opencv.org]. Esta función requiere los descriptores de los puntos hallados con el algoritmo SURF. Una vez hallado todos las posibles parejas de puntos se filtran las parejas de puntos cuya distancia entre puntos se encuentra bajo cierto umbral definido. La figura a continuación muestra un ejemplo del emparejamiento de característica realizado con FLANN.

Ilustración 20: Emparejamiento de características con FLANN.

3.3.6 Matriz Fundamental

Finalmente una vez hallados los puntos correspondientes de las imágenes estéreo es posible realizar el cálculo de la matriz fundamental. OpenCV cuenta con una implementación dedicada a esta labor; dicha función es *findFundamentalMat*. La sintaxis de esta función se presenta a continuación:

> mat findFundamentalMat (Arreglo de puntos en la imagen 1, Arreglo de puntos en la imagen 2, Método, Parámetro1,Parámetro2, Máscara)

La función recibe como entradas los arreglos de los puntos hallados y emparejados previamente; además, se debe especificar el método a usarse para hallar la matriz fundamental así como parámetros numéricos para el método escogido.

3.3.7 Reubicación final de los puntos en las imágenes

La reubicación de puntos en cada imagen se realiza con tres funciones de OpenCV las cuales serán descritas a continuación. La primera función es *stereoRectifyUncalibrated*; la cual, si bien trabaja para un sistema estéreo no calibrado, servirá para hallar las matrices de homografía H1 y H2 que contienen información sobre la transformación de perspectiva entre ambas imágenes. A continuación se presenta su sintaxis:

> bool**stereoRectifyUncalibrated** (Arreglo de puntos en la imagen 1, Arreglo de puntos en la imagen 2, Matriz fundamental, Tamaño de imagen, Matriz H1, Matriz H2)

Las dos matrices H1 y H2 son utilizadas para actualizar el valor de las matrices R1 y R2 las cuales son entradas para la siguiente función que se describirá.

La función *initUndistortRectifyMap* calcula la rectificación no distorsionada y la expresa como mapas para ser pasados a la última función del proceso de rectificación. La sintaxis de esta función es presentada a continuación:

> void**initUndistortRectifyMap** (Matriz de la cámara, Coeficientes de distorsión, Matriz R, Nueva matriz de la cámara, Tamaño de la imagen sin distorsión, tipo del mapa de salida, primer mapa, segundo mapa)

Esta función debe ejecutarse por cada cámara; de este modo se obtendrán cuatro mapas de los cuales dos son para la reubicación de puntos de cada imagen.

Finalmente se han hallado todos los parámetros necesarios para la rectificación de las imágenes. El último paso del proceso de rectificación se realiza con la función *remap* de OpenCV; esta función transforma la imagen de entrada en función a los mapas que recibe como entradas también. La sintaxis de la función es presentada a continuación:

> void**remap** (imagen de entrada, imagen de salida, primer mapa, segundo mapa, método)

La imagen de entrada es modificada a partir de los dos mapas calculados previamente y se obtiene la imagen de salida ya rectificada respecto a la otra imagen del par estéreo. Finalmente esa imagen de salida es almacenada como una imagen nueva para pasar al proceso de emparejamiento estéreo y obtener el mapa de disparidad.

A continuación se presentandos pares de imágenes estéreo que ponen en evidencia el cambio que se espera en el resultado después de aplicar el proceso de rectificación alprimer par de imágenes.

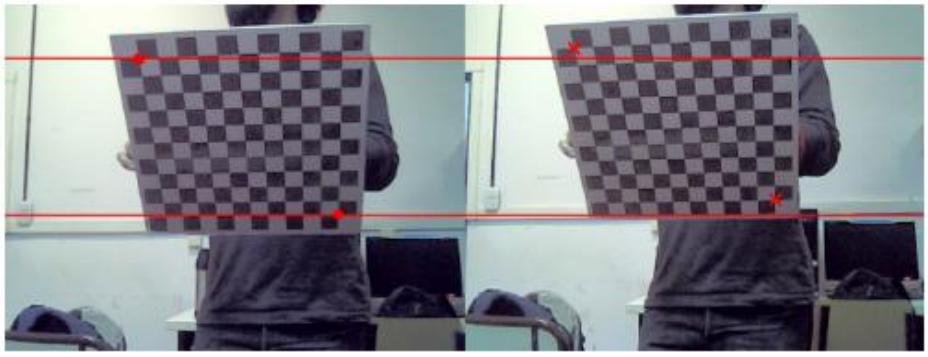


Ilustración 21: Par de imágenes no rectificado [6].

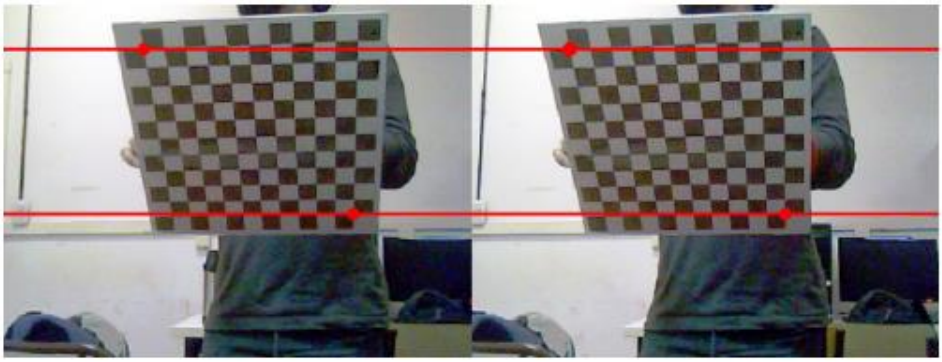


Ilustración 22: Par de imágenes rectificado [6].

En el primer par se aprecia que la linea roja no cruza por los mismos puntos del tablero de ajedrez. Sin embargo, después de aplicar la rectificación la línea atraviesa los mismos puntos en ambas imágenes.

3.3.8 Mapa de disparidad

En el capítulo anterior han sido descritos los conceptos básicos del emparejamiento estéreo y mapa de disparidad. Para el caso particular de la función implementada en OpenCV, el procesamiento está basado en el algoritmo de suma de diferencias absolutas (*Sum of Absolute Differences SAD)*. Sin embargo, en el capítulo anterior no fue descrito en detalle el proceso para el algoritmo SAD; por este motivo, se realizará una breve explicación del principio que opera en este algoritmo. La siguiente figura muestra un ejemplo de las imágenes proyectadas en dos cámaras de un arreglo estéreo canónico.

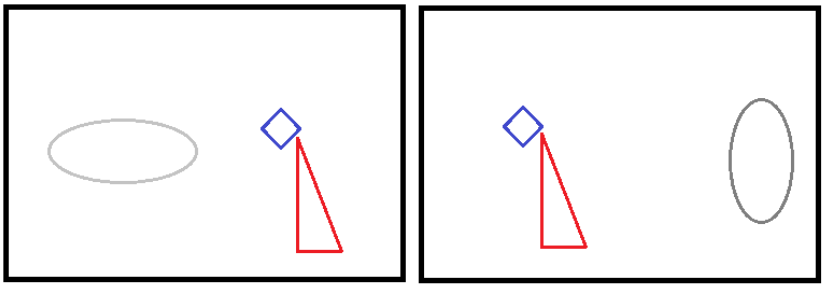


Ilustración 23: Par de imágenes (izquierda y derecha).

La imagen anterior sirve para ilustrar dos conceptos básicos en el emparejamiento estéreo. En primer lugar hay zonas de las imágenes que son imposibles de emparejar debido a que solo son visibles por una cámara a la vez. En la imagen, ambas figuras de forma ovalada son solo visibles para una cámara simultáneamente. Por otro lado, se explicará la búsqueda por ventanas utilizada en el algoritmo SAD.

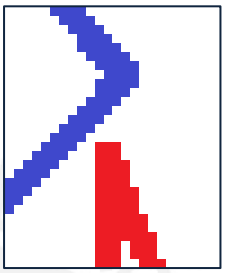


Ilustración 24: Ventana de la imagen estéreo para el emparejamiento.

El algoritmo SAD consiste en tomar una porción de imagen; es decir una ventana, y buscarla a lo largo de la línea epipolar que se asume está en la misma línea horizontal [13]. Es necesario realizar un barrido por todos los posibles puntos correspondientes y encontrar el más parecido posible. Tanto los métodos SAD y SSD son métodos basados en correlación; sin embargo se prefiere usar SAD debido a que los resultados son muy parecidos y la carga de procesamiento es menor. Cada pixel es restado de su correspondiente en la ventana y se toma el valor absoluto; finalmente, se suman los valores hallados para cada pixel de la ventana y ese valor final constituye el costo de ese emparejamiento. Una vez realizado el barrido completo, se escoge como pareja al punto cuyo costo sea el menor.

Además, cabe mencionar la métrica usada como costo en el algoritmo de Birchfield debido a que este es implementado en la función de OpenCV que se utilizará. Birchfield introduce el concepto de disimilitud que utiliza como una función de costo para determinar el camino en que las correspondencias sonmás probables. Si bien no se entrará en el detalle de su algoritmo, basta mencionar que se basa no solo en una correspondencia sino en una cadena de correspondencias; por este motivo,

el costo utilizado para la decisión de correspondencias está en función a una serie de posibles pixeles emparejados y no a una sola pareja. La disimilitud relaciona dos pixeles en su nivel de gris, sin embargo, no solo resta los valores numéricos sino que considera un rango a partirdel valor promedio de dos píxeles vecinos; por este motivo se considera una métrica a nivel subpixel. Gracias a esta métrica se evita que el algoritmo sea susceptible al muestreo de las imágenes y se descarten nodos incorrectos para acelerar la programación dinámica [2]. La siguiente imagen muestra el concepto de disimilitud de Birchfield donde se toma en consideración los niveles de gris de los píxeles anterior y posterior a y_i.

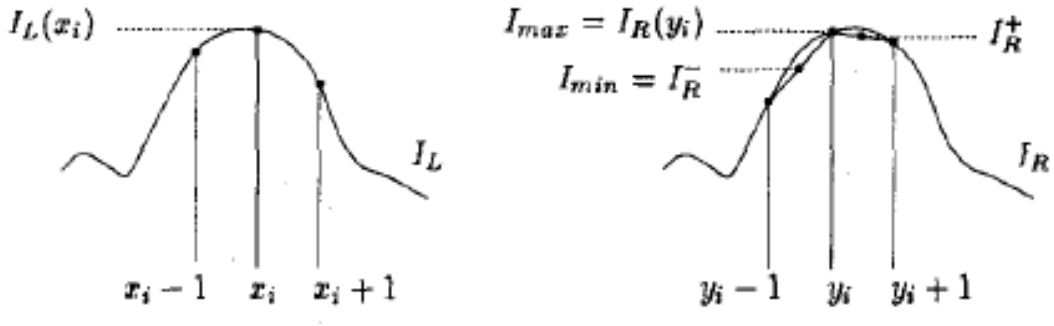


Ilustración 25: Disimilitud del algoritmo de Birchfield [2].

Finalmente, se describirá la función de OpenCV que implementa el algoritmo de emparejamiento estéreo. Esta función es llamada SGBM (*Semiglobal block matching)*; es decir, algoritmo de emparejamiento semiglobal. Como se mencionó en el anterior capítulo, este algoritmo está basado en el trabajo realizado por H. Hirschmuller pero prescinde de la información mutua y en su lugar usa la métrica de Birchfield para el emparejamiento estéreo. Además, esta función presenta dos niveles de filtrado; antes del procesamiento se somete las imágenes a un filtro Sobel mientras que al finalizar se realiza un filtrado para el control de unicidad e interpolación cuadrática. La sintaxis de esta función se presenta a continuación:

StereoSGBM(Disparidad mínima, Disparidad máxima, tamaño de ventana SAD, parámetros del algoritmo)

OperadorSGBM(Imagen izquierda, Imagen derecha, arreglo de salida de disparidad)

La primera sentencia corresponde a la inicialización de los parámetros del algoritmo. Es importante utilizar valores que garanticen un buen funcionamiento del algoritmo. Por otro lado, el operador hace uso del algoritmo con sus parámetros definidos y actúa sobre ambas imágenes para crear el mapa de disparidad.

El proceso completo al que son sometidas las dos imágenes adquiridas puede ser dividido en dos etapas. Estas dos etapas difieren entre sí porque la primera solo se requiere ejecutar una vez mientras que la segunda debe ser ejecutada por cada par estéreo a analizarse. La primera etapa consiste en analizar las imágenes de calibración y obtener los parámetros intrínsecos y extrínsecos del sistema para ser almacenados en un archivo XML o YAML. Este archivo será útil para el procesamiento de cada par estéreo más adelante debido a que los parámetros del sistema son únicos.

En el siguiente capítulo se mostrarán los resultados de cada etapa del procesamiento descrita en el presente capítulo.

CAPÍTULO 4: Pruebas y resultados

En este capítulo serán presentadas las imágenes que serán utilizadas para las pruebas de la implementación. Además, se mostrarán algunos detalles del almacenamiento de los parámetros internos en archivos XML/YAML. Finalmente, se apreciará el mapa de disparidad obtenido a partir de imágenes propias de la tesis; es decir, que no están rectificadas y cuyos parámetros no se conocen a priori.

4.1 Calibración del sistema estéreo

En esta etapa se han utilizado dieciséis pares de imágenes para la calibración. El patrón utilizado es un tablero de ajedrez asimétrico de 9x6 vértices internos en distintas poses o perspectivas. A continuación se muestran cuatro pares estéreo utilizados en la calibración.

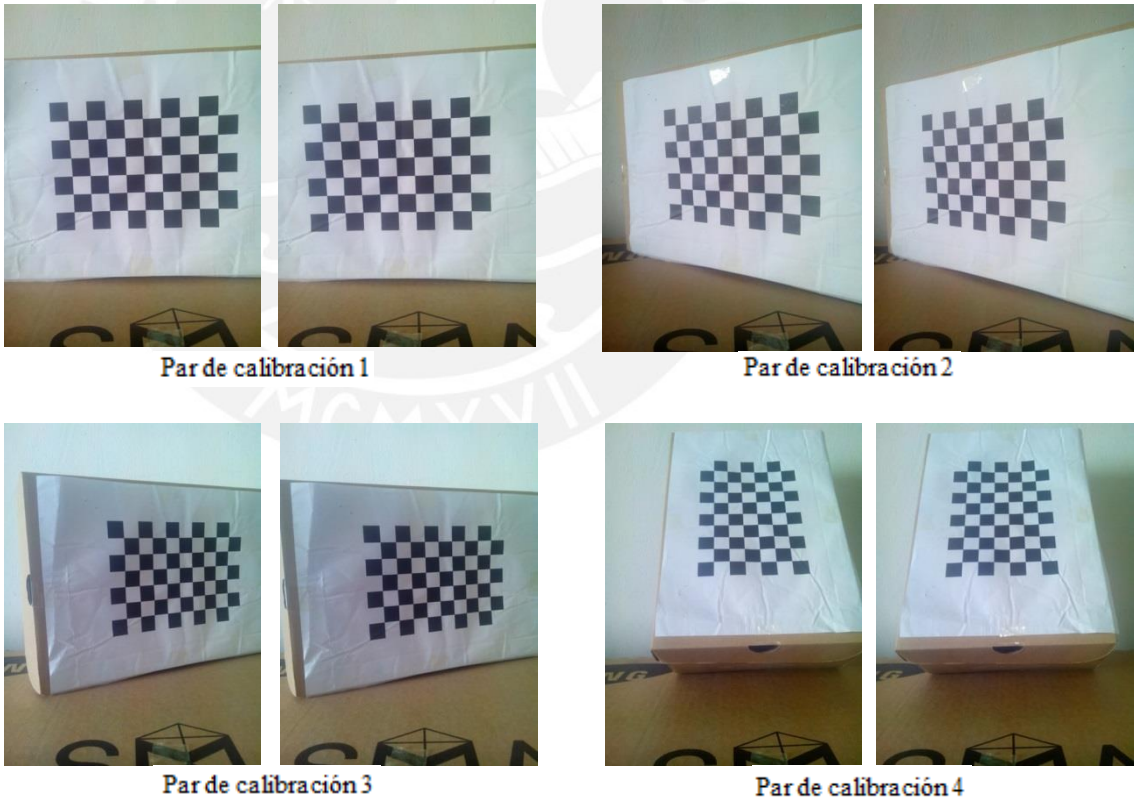


Ilustración 26: Imágenes utilizadas para la calibración.

El primer paso es realizar la detección de los vértices con la función **findChessboardCorners** y determinar si la imagen corresponde al patrón señalado ya que dicha función recibe como entrada el tamaño del tablero de ajedrez. El total de 16 pares de imágenes de calibración fueron detectados por la función y, para efectos de visualización se utiliza la función **drawChessboardCorners.** Esta función muestra la imagen con el patrón de calibración y señala con círculos de colores los vértices encontrados. A continuación se muestra el resultado obtenido a partir de los cuatro pares de imágenes provistos anteriormente:

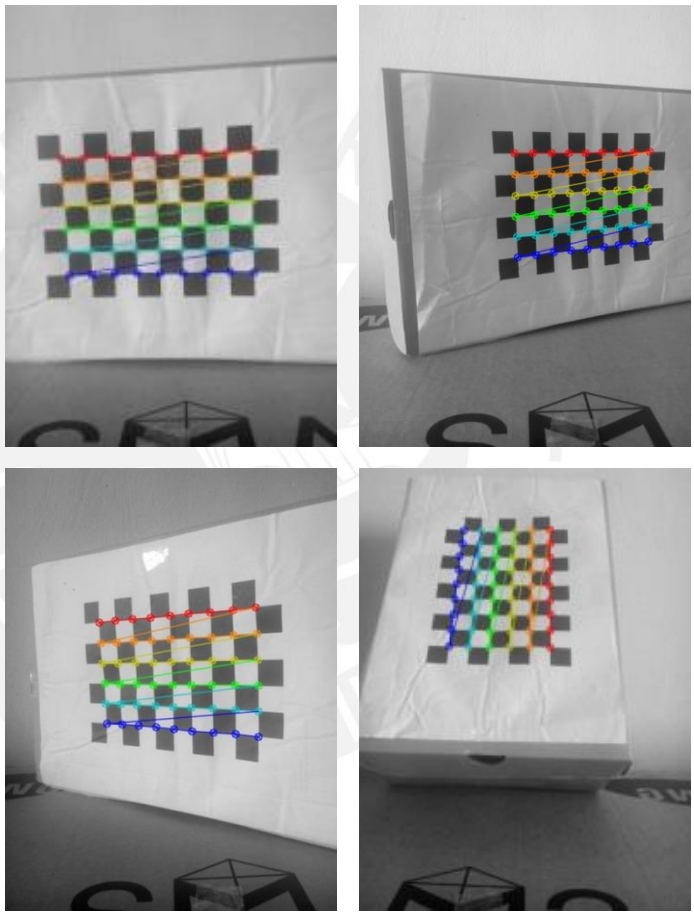


Ilustración 27: Reconocimiento de patrones en las imágenes de calibración.

El siguiente paso es obtener seis matrices importantes para el proceso con la función **stereoCalibrate**. Estas matrices son las matrices de las cámaras M1 y M2, los coeficientes de distorsión D1, D2, la matriz de rotación R y la matriz de traslación T.

Las matrices M1, M2, D1 y D2 son almacenadas en un archivo YAML para acceder a estas posteriormente.

A continuación se procede a utilizar la función **stereoRectify** para obtener los parámetros extrínsecos del sistema. La función recibe como entradas las matrices M1, M2, D1, D2, R y T para generar las matrices R1, R2, P1 y P2; estas matrices ya han sido explicadas en la descripción de la función.

Una vez hallados los parámetros intrínsecos y extrínsecos se culmina con la etapa base del procesamiento. Ambos tipos de parámetro se utilizan para generar dos archivos YAML que se utilizarán en el proceso de rectificación de cada par de imágenes estéreo a analizarse posteriormente. Los archivos de tipo XML/YAML son útiles para almacenar estructuras especiales de OpenCV; es posible crear un archivo, modificarlo y acceder a este. El resultado obtenido para las matrices mencionadas es el siguiente:

4.1.1 Parámetros intrínsecos

a. M1: !!opencv-matrix
   rows: 3
   cols: 3
   dt: d
   data: $\begin{bmatrix} 624.99 & 0 & 254.38 \\ 0 & 624.99 & 313.6 \\ 0 & 0 & 1 \end{bmatrix}$

b. M1: !!opencv-matrix
   rows: 3
   cols: 3
   dt: d
   data: $\begin{bmatrix} 624.99 & 0 & 243.79 \\ 0 & 624.99 & 339.54 \\ 0 & 0 & 1 \end{bmatrix}$

c. D1: !!opencv-matrix
   rows: 1
   cols: 8

dt: d

data: [0.438, -4.55, 0, 0, 0, 0, 0., -18.5]

d. D2: !!opencv-matrix

rows: 1

cols: 8

dt: d

data: [-0.3089, 3.807, 0, 0, 0, 0, 0., 7.5723]

4.1.2 Parámetros extrínsecos

e. R: !!opencv-matrix

rows: 3

cols: 3

dt: d

data: $\begin{bmatrix} 0.996 & 0.0275 & 0.0795 \\ -0.0242 & 0.9988 & -0.0418 \\ -0.08055 & 0.0397 & 0.99596 \end{bmatrix}$

f. T: !!opencv-matrix

rows: 3

cols: 1

dt: d

data: [-2.236, 0.18, 0.2167]

g. R1: !!opencv-matrix

rows: 3

cols: 3

dt: d

data: $\begin{bmatrix} 0.998 & -0.0564 & 0.0135 \\ 0.05612 & 0.998 & -0.0197 \\ -0.0146 & 0.0189 & 0.9997 \end{bmatrix}$

h. R2: !!opencv-matrix

rows: 3

cols: 3

dt: d

data: $\begin{bmatrix} 0.992 & -0.08 & -0.0961 \\ 0.0819 & 0.9965 & 0.0154 \\ 0.0945 & -0.0232 & 0.9952 \end{bmatrix}$

i. P1: !!opencv-matrix

rows: 3

cols: 4

dt: d

data: $\begin{bmatrix} -21488.2 & 0 & 7865.8 & 0 \\ 0 & -21488.2 & -8232.6 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$

j. P2: !!opencv-matrix

rows: 3

cols: 4

dt: d

data: $\begin{bmatrix} -21488.2 & 0 & 7865.8 & 48436.4 \\ 0 & -21488.2 & -8232.6 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$

Para efectos de visualización se han truncado los decimales de los valores de las matrices y se ha mantenido el formato en que se almacenan en el archivo YAML. A continuación se procederá a calcular la matriz fundamental a partir del emparejamiento de puntos de interés en las imágenes para luego pasar a la rectificación.

4.2 Emparejamiento estéreo sobre par rectificado

Para los resultados se considera necesario mostrar el producto obtenido de cada etapa del proceso que se definió. Por este motivo, lo primero será mostrar el funcionamiento de la implementación de emparejamiento estéreo. Para este fin, a continuación se muestra el resultado de aplicar el algoritmo al mismo par de imágenes que se utilizó en los resultados de los trabajos presentados en el estado del arte:

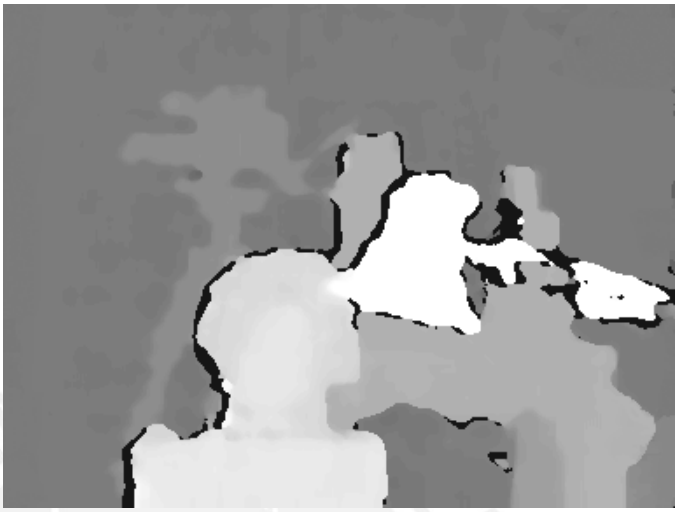


Ilustración 28: Mapa de disparidad con algoritmo SGBM.

Se puede apreciar que, tal como los resultados de los algoritmos expuestos en el estado del arte, se pueden distinguir todas las formas principales de la imagen y de este modo distinguir mediante la intensidad del gris los objetos más cercanos y lejanos. Sin embargo, el proceso ha sido realizado sobre imágenes ideales, a continuación se verá el proceso de rectificación sobre imágenes reales.

4.3 Rectificación

Para lograr la rectificación de las imágenes aun es necesario realizar el cálculo de la matriz fundamental. Para ello es necesario aplicar el algoritmo SURF para ubicación de puntos de interés y el algoritmo FLANN para emparejar los puntos en ambas imágenes. Antes de mostrar los resultados obtenidos con los algoritmos mencionados, se mostrará a continuación dos pares de imágenes que servirán para realizar las pruebas. Es importante mencionar que dichas imágenes representan un caso real donde las imágenes no están alineadas y no se puede aplicar algoritmos de emparejamiento estéreo directamente.

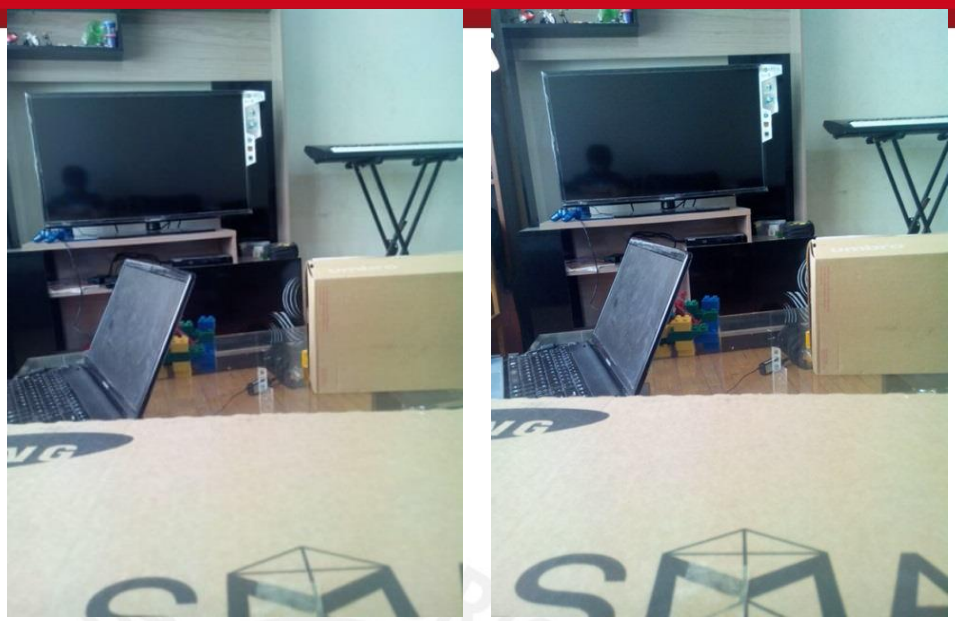


Ilustración 29: Par de imágenes 1.

Ilustración 30: Par de imágenes 2.

Todos los pasos de la rectificación serán aplicados sobre ambos pares de imágenes para determinar el funcionamiento de la implementación. En primer lugar será necesario ubicar puntos de interés en ambas imágenes con el algoritmo SURF. Para el

caso de la implementación, se probaron distintos valores del parámetro minHessian del detector SURF; usualmente este valor es fijado entre 300 y 800 según trabajos realizados con este algoritmo. Se escogió el valor de 390 para este parámetro ya que es el que mejor resultados producía. Este parámetro está relacionado al determinante de la matriz de Hessian que utiliza el algoritmo para determinar si un punto dado es aceptado como punto de interés SURF [10]. Por otro lado, el algoritmo de emparejamiento FLANN requiere como entradas los puntos y descriptores calculados con el algoritmo SURF para determinar los pares correspondientes. Este algoritmo tiene una métrica de distancia para determinar la probabilidad de que un par corresponda al mismo punto, basta aplicar un umbral mínimo sobre esta métrica para filtrar los posibles pares encontrados que sean erróneos. A continuación se muestra el proceso de ubicación de puntos de interés y su emparejamiento para los dos pares de imágenes presentados anteriormente:

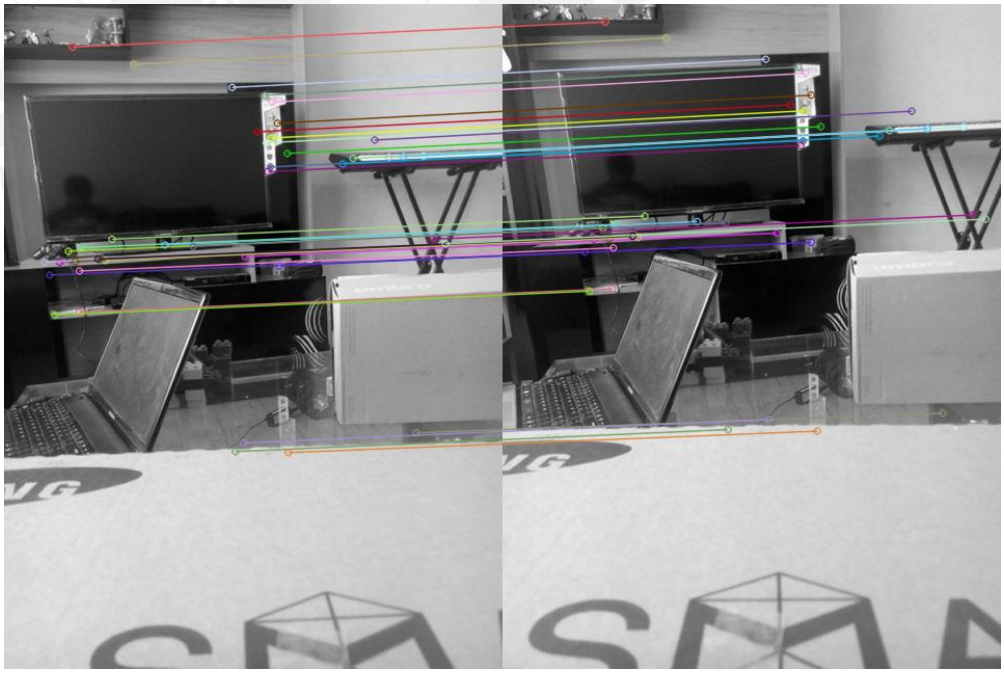


Ilustración 31: Emparejamiento de características en el par de imágenes 1.

Ilustración 32: Emparejamiento de características en el par de imágenes 2.

Para ambos casos el algoritmo FLANN es lo suficientemente exacto para emparejar los descriptores correspondientes a los mismos puntos. Sin embargo, aunque en el segundo caso la cantidad de puntos hallados es menor, la cantidad es suficiente para los siguientes pasos en el proceso de rectificación.

En este punto cabe destacar la naturaleza de los emparejamientos; las líneas que unen los puntos de interés, para ambas imágenes, son notoriamente paralelas pero no están alineadas horizontalmente con las bases de las imágenes. Esta observación confirma que el sistema, aunque se dispuso de la manera más cercana posible a la configuración canónica, no está rectificado ni alineado. En el caso de un par de imágenes perfectamente alineadas o rectificadas, las líneas que unen los puntos de interés son horizontales tal como se muestra en la siguiente figura.

Ilustración 33: Emparejamiento de características en un par de imágenes rectificado.

Gracias a esta observación ya se puede determinar que una imagen está ligeramente inclinada respecto a la otra. A continuación es necesario el cálculo de la matriz fundamental. El cálculo se realiza con la función **findFundamentalMat** la cual recibe como entrada los arreglos de puntos encontrados anteriormente. Se utiliza el método iterativo RANSAC sin especificar máscara alguna. Las matrices fundamentales encontradas para cada par de imagen son los siguientes:

$$F_{par\_imagenes\_1} = \begin{bmatrix} -0.00002047 & 0.00005364 & -0.0128699 \\ -0.000054646 & 0.000001389 & 0.0237734 \\ 0.0120076 & -0.0260659 & 1 \end{bmatrix}$$

$$F_{par\_imágenes\_2} = \begin{bmatrix} 0.0000842 & 0.0009451 & -0.224353 \\ -0.0009036 & -0.000060996 & 0.251736397 \\ 0.1708915 & -0.2157697 & 1 \end{bmatrix}$$

Una vez hallada la matriz fundamental, se utiliza la función **stereoUncalibrated** para calcular las dos matrices H1 y H2 las cuales son homografías de proyección de una imagen respecto a la otra. Las matrices de homografía relacionan los puntos en dos imágenes con una ecuación similar a la matriz fundamental o esencial. Estas matrices de homografía halladas son útiles para generar los parámetros finales que servirán para determinar los mapas de reubicación de las imágenes. Los parámetros quedan finalmente del siguiente modo:

R1 = M1 * H1 * M1

R2 = M2 * H2 * M2

P1 = M1

P2 = M2

El siguiente paso es generar los mapas de reubicación con la función **initUndistortRectifyMap** la cual recibe como entrada las matrices halladas en el paso anterior y genera dos mapas por cada imagen para ser transformadas, las sentencias quedan del siguiente modo:

```
initUndistortRectifyMap(M1, D1, R1, P1, imagen1.size(), CV_16SC2, rmap
    [0][0], rmap[0][1])

initUndistortRectifyMap(M2, D2, R2, P2, imagen2.size(), CV_16SC2, rmap
    [1][0], rmap[1][1])
```

Los mapas generados por la función denotados por rmap son utilizados para reubicar los puntos de ambas imágenes con la función *remap*. Las sentencias utilizadas se presentan a continuación:

```
remap(imagen1, imagen1_rectificada,rmap [0][0], rmap[0][1],
    CV_INTER_LINEAR)

remap(imagen2, imagen2_rectificada, rmap [1][0], rmap[1][1],
    CV_INTER_LINEAR)
```

Esta última función transforma ambas imágenes en función a los mapas hallados. Luego que este paso es realizado las imágenes estarán rectificadas y se puede pasar al proceso de emparejamiento estéreo para obtener el mapa de disparidad final.

Para terminar con la sección de rectificación se procederá a mostrar los resultados de la rectificación para los dos pares de imágenes seleccionados:

Ilustración 34: Resultado de rectificación del par de imágenes 1.

Ilustración 35: Resultado de rectificación del par de imágenes 2.

En

ambos casos, las imágenes han sido modificadas para alinear los puntos característicos de estas. En el primer par de imágenes la distorsión es mínima mientras que en el segundo caso se observa una ligera rotación de ambas imágenes. La cantidad de puntos hallados en el segundo caso es la causa de esta rotación; sin embargo, en el mapa de disparidad generado a partir de las imágenes rotadas se pueden apreciar las formas de los objetos cercanos.

4.4 Emparejamiento estéreo

Este último paso es el más sencillo en cuestión de implementación debido a que la función está optimizada y requiere pocas entradas. En el caso de la rectificación, era necesario realizar varios pasos, calcular parámetros y almacenarlos para su posterior uso. El emparejamiento estéreo se realiza con una sola función y, si las imágenes están correctamente rectificadas se garantiza un resultado favorable. Sin embargo, el algoritmo SGBM requiere de la inicialización de todos los parámetros pertenecientes a su clase. Entre los parámetros más importantes se tiene la disparidad mínima, disparidad máxima y el tamaño de ventana. Antes de mostrar los resultados del mapa de disparidad para los pares de imágenes dados, se mostrará la importancia de la rectificación. A continuación se tiene el resultado de aplicar el algoritmo de emparejamiento al primer par de imágenes sin rectificar; se puede notar que el algoritmo, al buscar puntos característicos que no están alineados, produce un resultado prácticamente aleatorio.

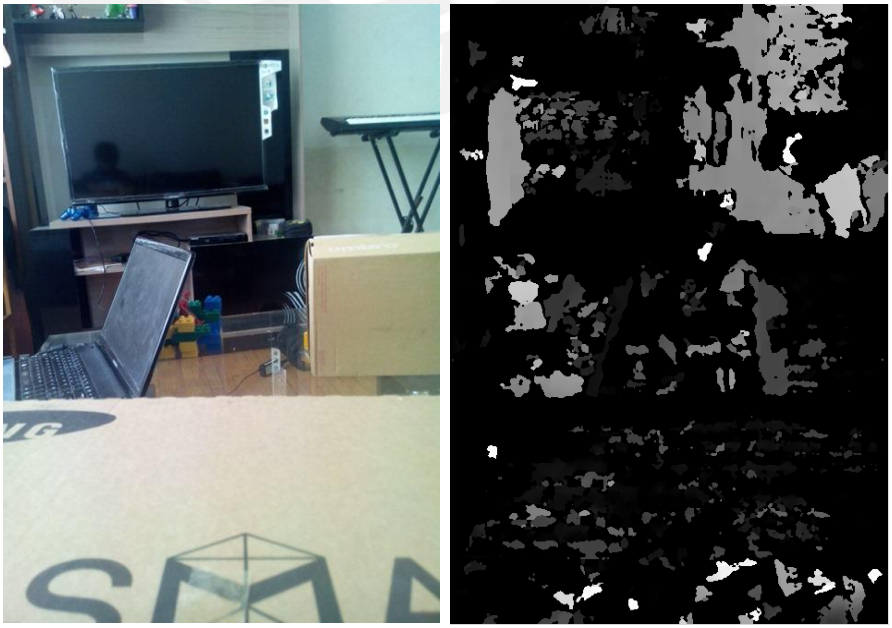


Ilustración 36: Resultado del mapa de disparidad para un par de imágenes no rectificado.

Lo primero que se puede notar en el mapa de disparidad generado es que no se distingue ninguna de las formas que existen en la imagen original; principalmente la computadora portátil y la caja rectangular que son los objetos medianamente cercanos a la cámara. Por otro lado, también se observa que la disparidad generada carece de sentido debido a que el fondo se ha detectado como un nivel de disparidad alto; es decir, un nivel de gris claro. Como se ha mencionado, el algoritmo asume que las imágenes están rectificadas y realiza su búsqueda horizontalmente; esta es la razón por la cual el proceso de rectificación es muy importante para el sistema.

Finalmente se procede a aplicar el algoritmo SGBM sobre los dos pares de imágenes rectificados hallados con la función remap. El resultado de los mapas de disparidad se muestra a continuación:

Ilustración 37: Resultado del mapa de disparidad para el par de imágenes rectificadas 1.

Ilustración 38: Resultado del mapa de disparidad para el par de imágenes rectificadas 2.

En la ilustración 37 se pueden apreciar en el mapa de disparidad tres formas básicas de la imagen. La computadora, la caja rectangular y los juegos de fichas son detectados con un nivel de gris medio, sugiriendo de este modo que la distancia a la que se encuentra es media respecto al fondo. En el caso de la caja más cercana sobre la que se apoya el sistema no se detectó correctamente debido al gran problema que tienen los algoritmos SAD. La falta de texturas dificulta la identificación de los pixeles que deben ser emparejados y, al ser la caja en su mayor parte de un solo color, la disparidad en esa zona no es clara. El televisor es identificado como parte del fondo y se le asigna un nivel de gris 0 que indica que la distancia es mucho mayor. El rango de colores está entre 0 y 255 representados por los colores negro y blanco respectivamente.

En el par de imágenes de la ilustración 38 se nota la ligera rotación debido al proceso previo de rectificación que introduce toda la zona blanca ubicada en la zona inferior; sin embargo, si se pueden apreciar las formas principales de la imagen original. Se ha medido que los niveles de gris que identifican a los objetos cercanos al arreglo de cámaras están entre veinte y ochenta. La guitarra y la caja presentan niveles de gris entre veinte y cuarenta mientras que la mesa y los dos frascos niveles entre sesenta y ochenta; por este motivo, con el objetivo de separar la zona de interés para la evasión

de obstáculos se pasará cada pixel por un filtro que solo deja pasar los valores entre veinte y ochenta. Por otro lado, el problema que se presentó en el par de imágenes anterior no se da en este caso, y las zonas más cercanas a las cámaras son representadas con un nivel de gris cercano al blanco (255). Finalmente, el fondo de la imagen es detectado correctamente con niveles de gris cercanos a 0 y se distingue claramente de los objetos cercanos al arreglo de cámaras estéreo.

4.5 Post-procesamiento del mapa de disparidad

Para lograr uniformizar el mapa de disparidad se ha aplicado un filtro de la mediana; además se ha aplicado un umbral sobre el resultado para obtener las zonas de interés. El filtro sirve para suavizar el cambio en los niveles de gris del mapa de disparidad mientras que el proceso de umbralización permitirá aislar las zonas de interés. Las siguientes imágenes muestran el resultado de aplicar ambos procesos sobre los pares de imágenes.

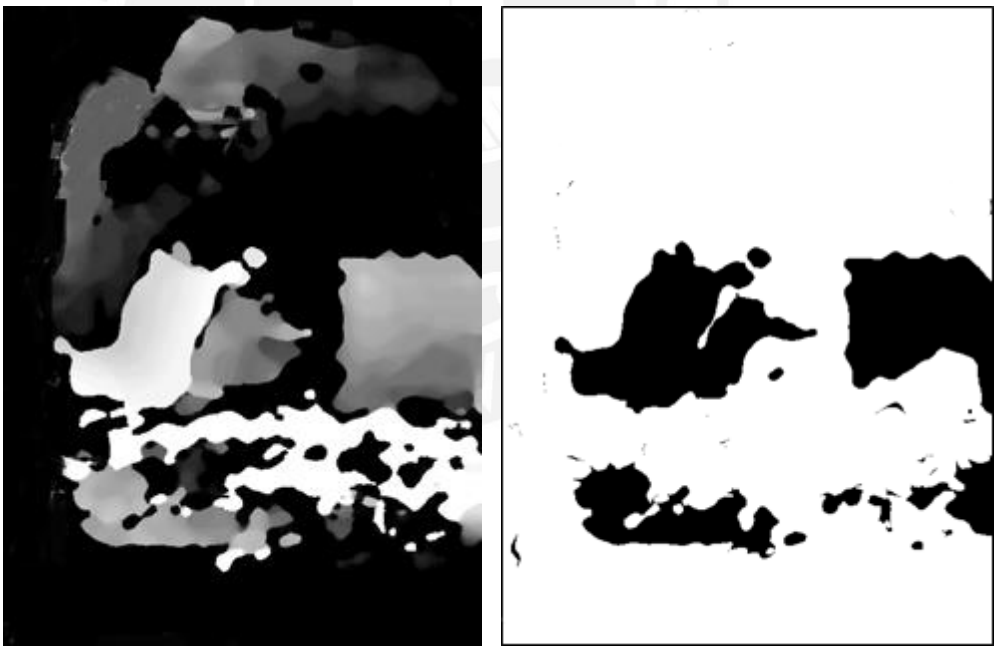


Ilustración 39: Filtrado y umbralización del par de imágenes 1.

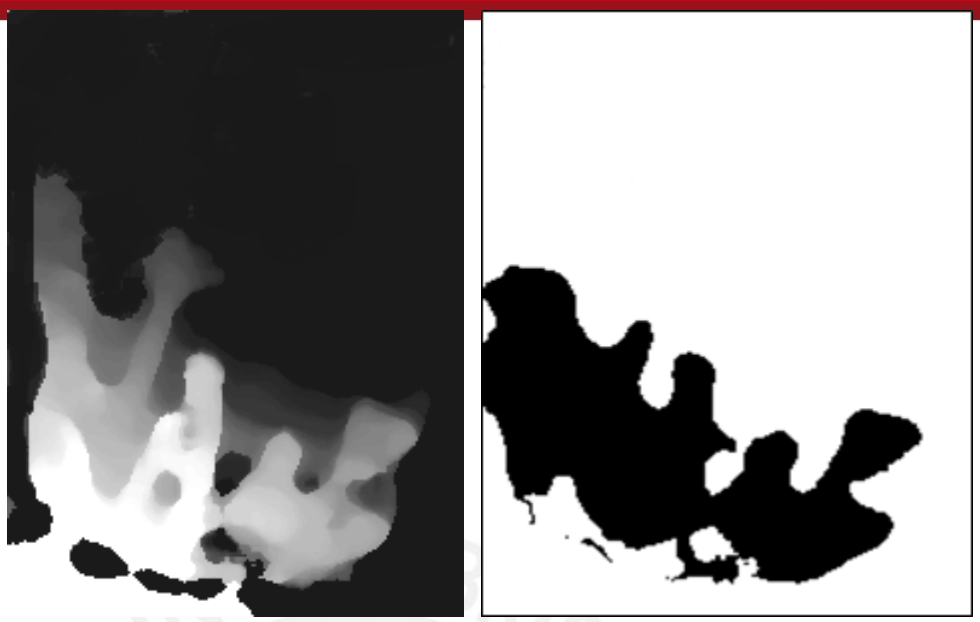
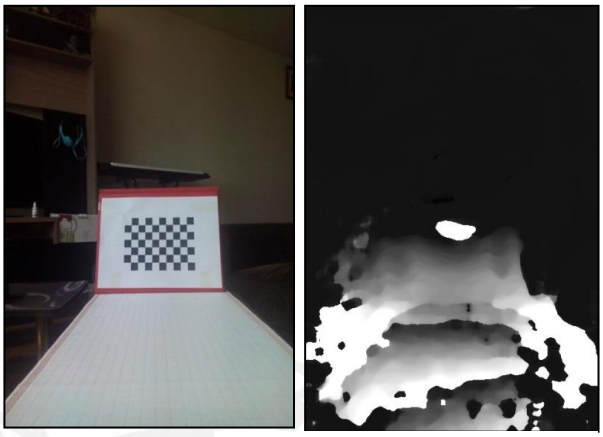


Ilustración 40: Filtrado y umbralización del par de imágenes 2.

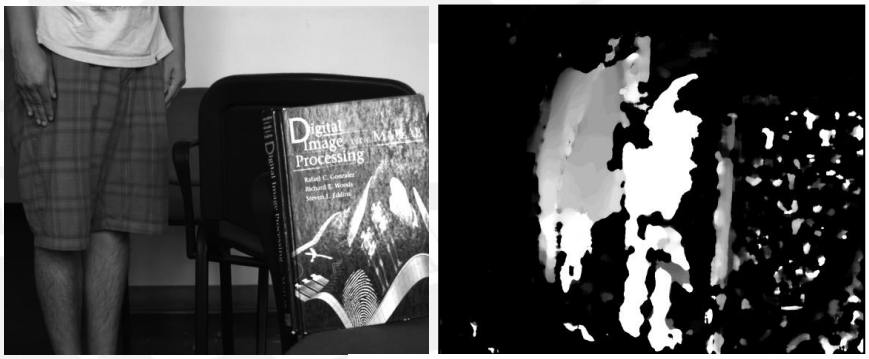
En ambos casos se logra suavizar los mapas de disparidad y a continuación aislar la zona de interés cuyo nivel de disparidad indica una distancia media respecto al fondo. Sin embargo, al igual que el mapa de disparidad inicial del primer par, se detectan zonas erróneas debido al error que se presentó por la falta de texturas en la caja. Por otro lado, en el segundo par si se aísla correctamente la zona donde se encuentran los objetos próximos al UAV.

Para los dos pares de imágenes estudiados es posible aislar del fondo la zona donde los objetos se encuentran a una distancia media. Para el fin de evitar obstáculos es necesario detectar los objetos a una distancia media pues, a medida que la unidad aérea no tripulada avance, esta distancia se reducirá. Se puede entender como distancia media a un rango de intensidad de los pixeles para los cuales, dada una velocidad determinada del UAV, se presenta riesgo de colisión. Este rango de valores debe determinarse con pruebas en el vuelo real del UAV en función de que tan rápido se requiere que reaccione el autopiloto de este. Para evitar que se produzca la colisión debe tomarse la decisión en función a los objetos que se encuentran en un nivel de gris medio con respecto al fondo y esto es posible debido a que dichos niveles de gris son detectados por el algoritmo y la zona de interés puede ser aislada.

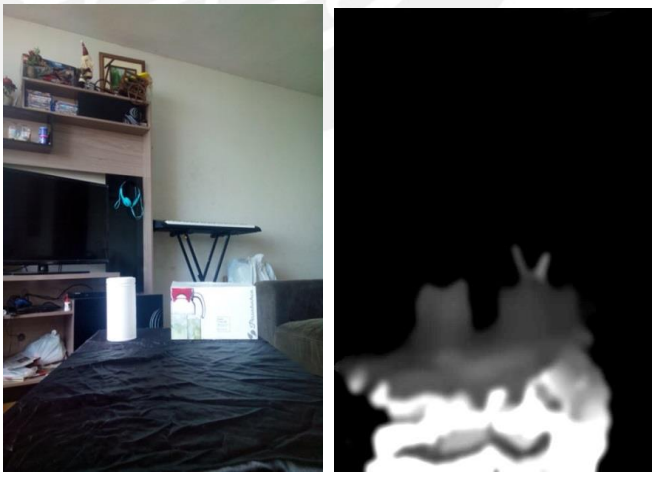
Para analizar los resultados de la implementación, a continuación se muestran pruebas adicionales que se realizaron sobre pares estéreo de distintos escenarios:



Par número 1



Par número 2



Par número 3

Ilustración 41: Pruebas adicionales de la implementación desarrollada.

En el primer par se observa que hay una hoja con un tablero de ajedrez a una distancia cercana al arreglo de cámaras así como una alfombrilla que va desde las cámaras hasta el tablero de ajedrez. En el mapa de disparidad hallado, se puede notar que toda la zona cercana al arreglo de cámaras se identifica como cercana. Debido a los filtros de suavizado y umbralización se obtiene todo el fondo de un tono muy oscuro. Esta zona delimitada por los colores claros es la que debe analizarse para decidir si representa o no un riesgo de colisión para el UAV y así mandar la señal de cambio de dirección al autopiloto para evitar daños a la aeronave.

Por otro lado, el segundo par muestra un escenario distinto donde todos los objetos a observarse están relativamente cerca al arreglo de cámaras. El mapa de disparidad obtenido logra identificar correctamente a la persona que está de pie y a la zona visible de la silla negra. Estos dos objetos presentan tonos de gris acorde al principio básico de disparidad; es decir, la silla, que está ligeramente más cerca que la persona, presente un tono de gris más claro. Sin embargo, el libro que se encuentra muy cerca al arreglo de cámaras no logra ser bien identificado excepto por ciertos puntos que delimitan su forma rectangular. Como se ha visto en las pruebas anteriores, objetos muy cercanos no son reconocidos correctamente en algunos casos.

Para finalizar con la presentación de los resultados, en el tercer par se tiene un escenario parecido al de la primera imagen. Sin embargo, en este caso se tiene todo una plataforma de color negro que llega a los dos objetos independientes que se muestran. Para este caso particular, se pasó la imagen por varias etapas de suavizado y, con el costo de perder nitidez, se elimina el ruido en el mapa de disparidad. En este mapa de disparidad se puede notar claramente que la zona de la plataforma negra y los objetos que se encuentran a una distancia media son representados por tonos de gris que se van degradando mientras más lejanos son los objetos. Para esta prueba se logra identificar correctamente los objetos y sus distancias al arreglo de cámaras.

5 Conclusiones

Un sistema de adquisición de imágenes real no puede llegar a conformar un sistema canónico; por este motivo existirá siempre una rotación y traslación entre las cámaras. Aunque a simple vista las imágenes sugieren estar correctamente alineadas, en realidad las líneas que unen sus puntos característicos no son horizontales. Por este motivo, al ejecutar el algoritmo sobre el par de imágenes originales, se genera un mapa de disparidad totalmente errado debido a la rotación y traslación que hay entre las cámaras. Se concluye pues, que no es posible prescindir del proceso de rectificación previo al emparejamiento estéreo.

El objetivo de la rectificación es modificar el par de imágenes estéreo de tal forma que todos sus puntos característicos estén alineados. Por este motivo, debido a que las cámaras se encuentran dispuestas físicamente de un modo cercano a un sistema canónico, se espera que la modificación de las imágenes debido a la rectificación sea mínima. Sin embargo, en los resultados se observa que la transformación de las imágenes resulta en una rotación de las imágenes que genera vacíos y, en consecuencia, pérdida de información. Esta distorsión genera falsos positivos y algunos errores en el mapa de disparidad. La distorsión por la rectificación está relacionada con la cantidad de puntos característicos emparejados pues, como se ve en el segundo par de imágenes en los resultados, todos los puntos se encuentran concentrados en la zona central de la imagen.

Según los resultados obtenidos durante la etapa de pruebas, se determina que todo el proceso propuesto funciona de manera aceptable aproximadamente el 60% de las veces que es ejecutado el programa. Se entiende por aceptable el hecho de encontrar un contraste claro de las formas de los objetos más cercanos respecto a los más lejanos. Hay casos como las imágenes de la computadora portátil o el libro, donde el error encontrado es significativo y el mapa de disparidad no es fiable para la decisión en la evasión de obstáculos pues el objeto más cercano no es correctamente identificado.

Durante la etapa de pruebas, en primer lugar se verificó el resultado del algoritmo de emparejamiento. Para este fin se utilizó el par de imágenes Tsukuba que se caracteriza por estar perfectamente alineado. Se obtiene como resultado un mapa de disparidad correcto y muy cercano al estado del arte presentado. Sin embargo, una vez que se trabaja con imágenes reales que requieren ser procesados con el algoritmo de rectificación, el resultado del mapa de disparidad varía en su exactitud. Por este motivo, se concluye que el proceso de rectificación requiere ser mejorado para evitar la pérdida de información en las imágenes así como la introducción de distorsión que genera errores en el mapa de disparidad final.

Según sea el nivel de error presentado en el mapa de disparidad, es posible aislar una zona de acuerdo a dos umbrales superior e inferior para determinar si existe riesgo de colisión. Para este fin, es necesario hacer pruebas con el hardware a utilizarse debido a que los umbrales a utilizarse dependen de la separación que existe entre las cámaras. En imágenes como el último ejemplo presentado, es posible determinar a partir del mapa de disparidad solo la zona que representa riesgo de colisión; sin embargo, como ya se hizo notar, en imágenes que presentan errores y falsos positivos no se puede aislar solo la zona de interés sino también zonas erradas.

6 Recomendaciones

Es necesario mejorar o buscar otro método para la ubicación de puntos característicos debido a que, como en el caso del segundo par de imágenes estéreo, los puntos encontrados son pocos y se concentran en una zona de la imagen. Esto ocasiona que los parámetro para las funciones de reubicación de puntos solo esté en función a una zona de la imagen.

Para garantizar que los parámetros hallados en la fase de calibración sean siempre los correctos, se recomienda la construcción de un sistema de visión estéreo no removible. Esto se debe a que cualquier ligera variación en la posición de las cámaras variará los parámetros hallados con la calibración. Por este motivo, al ser la calibración un proceso que solo se realiza una vez antes de comenzar con las pruebas, se requiere que los datos de calibración sean válidos siempre.

Se considera necesario como trabajo futuro mejorar el proceso de rectificación debido a que es un proceso complejo que en algunos casos introduce distorsiones en los bordes de las imágenes. Debido a que la etapa de emparejamiento estéreo tiene buenos resultados en imágenes correctamente rectificadas el problema de identificación de algunos objetos se encuentra en la rectificación. Si bien estos ruidos pueden ser filtrados, los resultados experimentales obtenidos en esta tesis demuestran que en muchos casos los errores de rectificación degradan significativamente la detección de las zonas cercanas al UAV en el mapa de disparidad.

Finalmente, se recomienda utilizar otras funciones o métodos de OpenCV de filtrado, emparejamiento o reubicación de puntos para comparar el resultado final con el obtenido en la presente tesis.

7 Bibliografía

[1] Gary Bradsky y Adrian Kaehler

2008 *Learning OpenCV*. Primera edición. Estados Unidos: O'Reilly.

[2] Stan Birchfield y Carlo Tomasi

1996 *Depth discontinuities by pixel-to-pixel stereo.* Technical Report STAN-CSTR-96-1573. Stanford University.

[3] Shan-shan Chen, Wu-hengZuo y Zhi-linFeng

2011 *Depth estimation via stereo vision using Birchfield's algorithm.*En IEEE 3rd International Conference.

[4] John F. Keane y Stephen S. Carr

2013 *A brief history of early unmanned aircraft.* En John Hopkins APL technical digest.

[5] ElanDubrofsky

2009 *Homography estimation.* University of British Columbia

[6] Taihú Pire

2012 *Evasión de obstáculos usando visión estéreo.* VII Jornadas Argentinas de Robótica.

[7] L. García, A. Dzul, R. Lozano y C. Pégard

2011 *Combining stereo vision and inertial navigation system fora quad-rotor UAV.* Springer Science Business Media.

[8] Peter van Blyenburgh

1999 *UAVs: an overview.*Air & Space Europe.Vol 1.

[9] HeikoHirschmüller

2005 *Accurate and efficient stereo processing by semi-globalmatching and mutual information.* Vol.2 .Computer Vision and Pattern Recognition, 2005.CVPR 2005.IEEE Computer Society Conference.

[10] H. Bay, T. Tuytelaars y L. van Gool

2006 *SURF: Speeded up robust features.* Proceedings of the Ninth European Conference on Computer Vision.

[11] H. Soltani, H. D. Taghirad y A.R. NorouzzadehRavari

2012 *Stereo-based visual navigation of mobile robots in unknown environments.* En 20th Iranian Conference on Electrical Engineering.

[12] D. Hausamann, W. Zirnig y G. Schreier

2003 *Monitoring of gas transmission pipelines-a customer driven civil UAV application.* ODAS conference.

[13] J. Olivares, J. Hormigo, J. Villalba e I. Benavides

2004 *Minimum sum of absolute differences implementation in a singles FPGA device.* IEEE Int. Conf. on Field Programmable Logic.

[14] O. Schreer

1998 *Stereo vision-based navigation in unknown indoor environment.* 5th European Conference on Computer Vision.

[15] M. Samadi, M. Fauzi Othman y S. Amin

2013 *Stereo vision based robots: fast and robust obstacle detection method.* Center for artificial intelligence and robotics. Universidad Tecnológica de Malasia.

[16] D. Scharstein y R. Szeliski

2002 *A taxonomy and evaluation of dense two-frame stereo correspondence algorithms.* International Journal of Computer Vision.

[17] M. Chen, Z. Vai y Y. Wang

2012 *A method for mobile robot obstacle avoidance based on stereo vision.* 10th IEEE International Conference, Industrial Informatics (INDIN).

[18] Zhengyou Zhang

1996 *Determining the epipolar geometry and its uncertainty: a review*. International Journal of Computer Vision, 27(2).7.

[19] G. Olague, F. Fernandez, C. Perez y E. Lutton

2006 *The infection algorithm: an artificial epidemic approach for dense stereo correspondence.* Artificial Life.

[20] K. Ambrosch y W. Kubinger

2010 *Accurate hardware-based stereo vision.* CVIU.

[21] M. Mozerov y J. van Weijer

2014 *Accurate stereo matching by two step global optimization*. Suscrito a TIP 2014.

[22] Q. Yang, P. Ji, D. Li, S. Yao y M. Zhang

2013 *Near real-time stereo matching using adaptive guided filtering.* Suscrito a *Image and Vision Computing 2013*

Referencias web

Middlebury

2014 Middlebury stereo evaluation
Consulta: 03 de Junio del 2014
Disponible en: <http://vision.middlebury.edu/stereo/eval/>

Barrick

2014 Comunicados: Tecnología de alto vuelo
Consulta: 25 de Noviembre del 2014
Disponible en:
<http://barricklatam.com/barrick/noticias/comunicados/tecnologia-de-alto-vuelo/2014-11-11/223941.html>

OpenCV

2014 OpenCV: tutorials and documentation
Consulta: 24 de Noviembre del 2014
Disponible en: <http://opencv.org >

International Potato Center

2014 Invasion of the potato drones
Consulta: 03 de Junio del 2014
Disponible en:
<http://cipotato.org/press-room/blogs/invasion-of-the-potato-drones/>