Universidad de Costa Rica
Facultad de Ingeniería
Escuela de Ingeniería Eléctrica

IE0499 – Proyecto Eléctrico

Diseño, implementación y prueba de un algoritmo para la estimación de la forma tridimensional de una superficie en exploración, mediante cortes lineales equidistantes del campo de profundidad, suministrado por una cámara estereoscópica

por

Santiago Mora Zúñiga

Ciudad Universitaria Rodrigo Facio

Enero de 2019

IPCV-LAB

Diseño, implementación y prueba de un algoritmo para la estimación de la forma tridimensional de una superficie en exploración, mediante cortes lineales equidistantes del campo de profundidad, suministrado por una cámara estereoscópica

por

Santiago Mora Zúñiga

B44629

IE0499 – Proyecto Eléctrico

Aprobado por

Ing. Geovanni Martínez Castillo, Dr.-Ing
Profesor guía

Ing. Luis Diego Rojas Muñoz, M.Sc.
Profesor lector

Ing. Peter De Ford González, M.Sc.
Profesor lector

Enero de 2019

IPCV-LAB

Resumen

Diseño, implementación y prueba de un algoritmo para la estimación de la forma tridimensional de una superficie en exploración, mediante cortes lineales equidistantes del campo de profundidad, suministrado por una cámara estereoscópica

por

Santiago Mora Zúñiga

Universidad de Costa Rica
Escuela de Ingeniería Eléctrica, IPCV-LAB
Profesor guía: Ing. Geovanni Martínez Castillo, Dr.-Ing
Enero de 2019

El proyecto se llevó a cabo dentro del Labotario de Investigación en Procesamiento de Imágenes y Visión por Computador (IPCV-LAB) de la Escuela de Ingeniería Eléctrica de la Universidad de Costa Rica, el cual consiste en el desarrollo de un algoritmo capaz de describir y estimar la forma irregular de la de la superficie en exploración mediante una red rígida de triángulos constituida por más de cuatro vértices y dos triángulos, obtenida a partir del campo de profundidad proporcionado por una cámara estereoscópica. Así mismo, el algoritmo desarrollado deberá determinar la posición y orientación de la red de triángulos con respecto al sistema de coordenadas de la cámara en cualquier instante de tiempo.

El algoritmo diseñado se implementó en el lenguaje de programación C, haciendo uso de la plataforma de programación ROS. El campo de profundidad se generó con la cámara estereoscópica Bumblebee 2, y haciendo uso de dichos datos se aplicó el método iterativo de RANSAC, junto con un método que se basa en el corte de líneas equidistantes del campo de profundidad, para generar la red de triángulos que estima la forma tridimensional de la porción de la superficie que está siendo capturada por la cámara. Luego, se definió un sistema de coordenadas local sobre la red de triángulos, en uno de los vértices, para poder describir la posición y orientación de la red con respecto al sistema de coordenadas de la cámara izquierda.

A la hora de realizar las pruebas experimentales no se contaba con un método para calcular el error de estimación de forma, entre la forma descrita por la red de triángulos generada por el algoritmo y la forma real de la superficie. Solo se contaba con la imagen capturada por la cámara izquierda y la gráfica de la red de triángulos generada por el algoritmo, para determinar

la precisión y exactitud de este mediante comparación y observación.

Lo resultados de las pruebas experimentales demostraron que el algoritmo es capaz generar una red rígida de triángulos que trata de estimar la forma irregular de casi cualquier superficie que está siendo capturada por la cámara izquierda, a excepción de cuando se tienen objetos altos o muy irregulares sobre la superficie plana.

Palabras claves*: Estereoscopía, Estimación de Forma Tridimensional, Orientación, Robótica, Posición, Visión por Computador.*

Índice general

Índice general vii

Índice de figuras viii

Nomenclatura xiii

1 Introducción 1

2 Objetivos 3
2.1. Objetivo general . . . 3
2.2. Objetivos específicos . . . 3

3 Metodología 5

4 Estereoscopía 7
4.1. La cámara estereoscópica . . . 7
4.2. Pose de un objeto rígido . . . 8
4.2.1. Modelo de posición de un objeto rígido . . . 9
4.2.2. Modelo de orientación de un objeto rígido . . . 9
4.2.3. Modelo de pose de un objeto rígido . . . 10
4.3. Cálculo de la posición tridimensional de un punto utilizando una cámara estereoscópica . . . 11
4.3.1. Obtener una nube de puntos utilizando una cámara estereoscópica . . . 15
4.4. Implementación usando el Sistema Operativo Robótico (ROS) . . . 16

5 Generar una red de triángulos a partir de una nube de puntos 19
5.1. Definición de una red de triángulos . . . 19
5.2. División de la imagen izquierda en líneas verticales . . . 20
5.3. Obtener la linea de mejor ajuste . . . 22
5.4. Obtener los vértices a partir de la línea de mejor ajuste . . . 26
5.4.1. Obtener un segmento de la línea de mejor ajuste a partir de las líneas vistas que pasan por el primer y último píxel de la columna considerada 27

5.4.2. Selección de los vértices 31
5.5. Asignar los vértices a cada uno de los triángulos que componen la red de triángulos 34
5.6. Sistema de coordenadas local de la red de triángulos 37
5.6.1. Posición de la red de triángulos 38
5.6.2. Orientación de la red de triángulos 38
5.6.3. Vértices respecto al sistema de coordenadas local 39
5.7. Implementación del algoritmo para generar una red de triángulos a partir de una nube de puntos. 39

6 Resultados experimentales **43**
6.1. Cámara estereoscópica utilizada 43
6.2. Entorno controlado de pruebas 44
6.3. Planteamiento de las pruebas 46
6.4. Resultados de las pruebas y análisis 48

7 Conclusiones y recomendaciones **57**
7.1. Conclusiones 57
7.2. Recomendaciones 59

Apéndice **60**

A Funcionamiento de ROS **61**

B Algoritmo de RANSAC **65**
B.1. Algoritmo 65
B.1.1. Umbral de distancia t para determinar la compatibilidad dato/modelo . . 66
B.1.2. Umbral de distancia t para determinar la compatibilidad dato/modelo . . 66
B.1.3. Numero de iteraciones N 66
B.1.4. Número mínimo T de puntos en el subconjunto de decisión 66

Bibliografía **69**

Índice de figuras

4.1. Diagrama de una cámara estereoscópica con dos lentes y sus sistemas de coordenadas alineados. 8

4.2. Posición de un objeto rígido cuando ambos sistemas de coordenadas están alineados. 9
4.3. Posición de cualquier punto del objeto con respecto al sistema de coordenadas de referencia. 11
4.4. Sistema de coordenadas de la imagen (c,r), del plano de la cámara (u,v) y de la cámara (x,y,z). 13
4.5. Calculo de la posición tridimensional de un punto a partir de la intersección de la línea vista izquierda con la derecha. 15
4.6. Nube de puntos respecto al sistema de coordenadas de referencia (cámara izquierda). 16
4.7. Implementación de nodos de ROS para obtener la nube de puntos. 17

5.1. Rectángulo modelado con una red de triángulos. 20
5.2. División de la imagen izquierda en líneas verticales. 20
5.3. Conjunto de píxeles de una columna asociados a un conjunto único de puntos de la nube de puntos. 21
5.4. Línea que mejor se ajusta a la porción de la nube de puntos asociada a una columna de la imagen. 23
5.5. Recta que se ajusta a dos puntos elegidos aleatoriamente de la porción de la nube de puntos a considerar. 24
5.6. Distancia entre un punto de la porción de la nube de puntos y la recta estimada. . . 25
5.7. Obtención de la línea que mejor se ajuste a una porción de la nube de puntos mediante el algoritmo de RANCSA. 26
5.8. Segmento de la línea de mejor ajuste que va de un α_1 a un α_2. 27
5.9. Segmento de la línea de mejor ajuste obtenido a partir de las líneas vistas. 28
5.10. División del segmento que se encuentra entre α_1 y α_2 en varios puntos. 31
5.11. Vértice que pertenece a la nube de puntos y que cumple con la condición del ángulo. 33
5.12. Rectángulos y triángulos entre dos columnas de vértices consecutivas. 35
5.13. Numeración de los vértices entre dos columnas consecutivas para $NVC = 7$. 36
5.14. Triángulo compuesto por los vértices A, B y C. 36
5.15. Casos al asignar los vértices a los triángulos entre dos columnas consecutivas. . . . 37
5.16. Sistema de coordenadas local de la red de triángulos y el sistema de coordenadas de referencia. 38
5.17. Archivo .stp que describe la red de triángulos que estima una forma rectangular. . . 40
5.18. Archivo .txt que describe la posición y orientación de la red de triángulos respecto al sistema de coordenadas de la cámara izquierda. 41
5.19. Comunicación entre los nodos necesarios para obtener y graficar una red de triángulos a partir de una nube de puntos. 42

6.1. Cámara Bumblebee 2 de la marca Point Grey. 43
6.2. Espacio para realizar las pruebas experimentales. 44
6.3. Inclinación horizontal de la cámara, utilizada para realizar las pruebas experimentales. 45
6.4. Inclinación vertical de la cámara, utilizada para realizar las pruebas experimentales. 45

6.5. Altura de la cámara respecto del suelo, utilizada para realizar las pruebas experimentales. 46
6.6. Prueba 1. Piedra pequeña sobre la superficie capturada por la cámara. 50
(a). Imagen capturada por la cámara izquierda. 50
(b). Gráfica de la red de triángulos. 50
6.7. Prueba 2. Piedra mediana sobre la superficie capturada por la cámara. 50
(a). Imagen capturada por la cámara izquierda. 50
(b). Gráfica de la red de triángulos. 50
6.8. Prueba 3. Piedra grande sobre la superficie capturada por la cámara. 51
(a). Imagen capturada por la cámara izquierda. 51
(b). Gráfica de la red de triángulos. 51
6.9. Prueba 4. Una piedra grande, una pequeña y una mediana sobre la superficie capturada por la cámara. 51
(a). Imagen capturada por la cámara izquierda. 51
(b). Gráfica de la red de triángulos. 51
6.10. Prueba 5. Varias piedras pequeñas sobre la superficie capturada por la cámara. . . . 52
(a). Imagen capturada por la cámara izquierda. 52
(b). Gráfica de la red de triángulos. 52
6.11. Prueba 6. Varias piedras medianas sobre la superficie capturada por la cámara. . . . 52
(a). Imagen capturada por la cámara izquierda. 52
(b). Gráfica de la red de triángulos. 52
6.12. Prueba 7. Varias piedras grandes sobre la superficie capturada por la cámara. 53
(a). Imagen capturada por la cámara izquierda. 53
(b). Gráfica de la red de triángulos. 53
6.13. Prueba 8. Varias piedras grandes, medianas y pequeñas sobre la superficie capturada por la cámara. 53
(a). Imagen capturada por la cámara izquierda. 53
(b). Gráfica de la red de triángulos. 53
6.14. Prueba 9. Una piedra al lado de otra sobre la superficie capturada por la cámara. . . 54
(a). Imagen capturada por la cámara izquierda. 54
(b). Gráfica de la red de triángulos. 54
6.15. Prueba 10. Una piedra detrás de otra sobre la superficie capturada por la cámara. . . 54
(a). Imagen capturada por la cámara izquierda. 54
(b). Gráfica de la red de triángulos. 54
6.16. Prueba 11. Dinosaurio de plástico (objeto irregular) sobre la superficie capturada por la cámara. 55
(a). Imagen capturada por la cámara izquierda. 55
(b). Gráfica de la red de triángulos. 55
6.17. Prueba 12. Cilindro de piedra (objeto alto) sobre la superficie capturada por la cámara. 56
(a). Imagen capturada por la cámara izquierda. 56
(b). Gráfica de la red de triángulos. 56
(c). Gráfica de la red de triángulos vista desde la izquierda. 56

(d). Gráfica de la red de triángulos vista desde la derecha. 56

A.1. Comunicación entre dos nodos a través de un tópico en ROS. 61
A.2. Función del maestro en ROS. 62
A.3. Ejemplo del uso de tres nodos de ROS para obtener una nube de puntos a partir de una cámara estereoscópica. 63

Nomenclatura

α Factor de magnitud del vector que describe a la línea de mejor ajuste.

α_1 Factor de magnitud que define el punto más cercano de un segmento de la línea de mejor ajuste.

α_2 Factor de magnitud que define el punto más lejano de un segmento de la línea de mejor ajuste.

α_j Factor de magnitud del vector que va de un punto de la nube de puntos a un punto del segmento.

α_l Factor de magnitud del vector proyectado desde el sistema de coordenadas de la cámara izquierda.

α_r Factor de magnitud del vector proyectado desde el sistema de coordenadas de la cámara derecha.

α_{kj} Factor de magnitud del vector que describe la posición tridimensional de un punto del segmento.

$\Re_x$ Ángulo de rotación alrededor del eje x.

$\Re_y$ Ángulo de rotación alrededor del eje y.

$\Re_z$ Ángulo de rotación alrededor del eje z.

$\mathbf{G}$ Vector de posición tridimensional de un objeto respecto al sistema de coordenadas de referencia.

$\mathbf{G}_o^l$ Vector de posición tridimensional de la red de triángulos respecto al sistema de coordenadas de referencia.

$\mathbf{H}$ Vector de posición tridimensional de un punto respecto al sistema de coordenadas local del objeto.

$\mathbf{H}^*$ Vector de posición tridimensional de un punto respecto al sistema de coordenadas de referencia.

$\mathbf{O}$ Vector de orientación de un objeto respecto al sistema de coordenadas de referencia.

$\mathbf{O}_o^l$ Vector de orientación de la red de triángulos respecto al sistema de coordenadas de referencia.

$\mathbf{P}_1$ Primer punto elegido aleatoriamente de una porción de la nube de puntos.

$\mathbf{P}_2$ Segundo punto elegido aleatoriamente de una porción de la nube de puntos.

$\mathbf{P}_j$ Vector de posición tridimensional de un punto del segmento respecto al sistema de coordenadas de referencia.

$\mathbf{P}_k$ Vector de posición tridimensional de un punto de la nube de puntos respecto al sistema de coordenadas de referencia.

$\mathbf{P}_{vk}$ Vector de posición tridimensional de un vértice respecto al sistema de coordenadas de referencia.

$\mathbf{R}$ Matriz de rotación.

$\mathbf{R}_o^l$ Matriz de rotación de la red de triángulos.

$\mathbf{V}_1$ Vector de posición tridimensional del primer vértice respecto al sistema de coordenadas de referencia

$\mathbf{V}_o$ Vector de posición tridimensional de un vértice respecto al sistema de coordenadas local.

θ_{kj} Ángulo entre la línea de mejor ajuste y una línea que contiene un punto de la nube de puntos y un punto del segmento.

θ_{min} Mínimo ángulo aceptable entre la línea de mejor ajuste y una línea que contiene un punto de la nube de puntos y un punto del segmento.

A Número del primer vértice de un triángulo.

B Línea base.

B Número del segundo vértice de un triángulo.

C Número del tercer vértice de un triángulo.

c Eje horizontal de la imagen en píxeles.

C_l cámara izquierda.

C_r cámara derecha.

d_i Distancia entre un punto de la nube de puntos y la línea de mejor ajuste.

d_{max} Mayor distancia aceptable entre un punto de la nube de puntos y la línea de mejor ajuste.

f Distancia focal.

FC Factor de columna.

H Alto de la imagen.

in_T Cantidad total de puntos clasificados como valores típicos.

in_{min} Mínima cantidad aceptable de puntos clasificados como valores típicos.

$IPCV - LAB$ Laboratorio de Investigación en Procesamiento Digital de Imágenes y Visión por Computador.

LTS Soporte a Largo Plazo.

N Número máximo de iteraciones para encontrar la línea de mejor ajuste.

NC Número de columnas.

NC_{2c} Cantidad de rectángulos entre dos columnas consecutivas.

NCI Número de columna inicial.

NT Número de triangulo.

NT_{2c} Cantidad de triángulos entre dos columnas consecutivas.

NVC Número de vértices por columna.

pc_l plano de la cámara izquierda.

pc_r plano de la cámara derecha.

PEC Paso entre columnas.

PEP Paso entre puntos.

r Eje vertical de la imagen en píxeles.

$RANSAC$ Muestra Aleatoria de Consenso.

ROS Sistema Operativo Robótico.

$RViz$ Visualización en ROS.

TIN Malla Irregular Triangulada.

u Eje horizontal del plano de la cámara en centímetros.

u Eje vertical del plano de la cámara en centímetros.

VGA Matriz de Gráficos de Video.

W Ancho de la imagen.

Capítulo 1

Introducción

En el Laboratorio de Investigación en Procesamiento Digital de Imágenes y Visión por Computador (IPCV-LAB), del Departamento de Electrónica y Telecomunicaciones, de la Escuela de Ingeniería Eléctrica, actualmente se está desarrollando un algoritmo de odometría visual monocular, capaz de determinar la posición y la orientación de un robot autónomo de exploración, a partir de la secuencia de imágenes capturada por una cámara monocular, la cual se encuentra atada rígidamente a la estructura del robot, específicamente en un costado, inclinada hacia la superficie en exploración. Para la determinación de la posición y de la orientación del robot en un instante de tiempo arbitrario, se requiere conocer la forma, posición y orientación respecto al sistema de coordenadas local de una porción de la parte de la superficie que está siendo capturada por la cámara monocular. Actualmente, se supone que la superficie es plana y rígida, descrita con una red rectangular constituida por cuatro vértices y dos triángulos. Para la determinación de la pose inicial, se emplea un algoritmo conocido en la literatura científica como algoritmo de calibración de Tsai, el cual utiliza un patrón de calibración que se debe de colocar en la escena al inicio de la secuencia de imágenes para realizar el cálculo. Desafortunadamente, las superficies sobre las cuales se mueve un robot autónomo de exploración son en su mayoría irregulares. En este proyecto, se desarrollará y diseñará un nuevo algoritmo que sea capaz de describir la forma irregular de la superficie en exploración mediante una red rígida de triángulos constituida por más de cuatro vértices y dos triángulos. Estos vértices se ubicarán a lo largo de líneas de corte equidistantes del campo de profundidad suministrado por una cámara estereoscópica modelo Bumblebee2 de la marca Point Grey. Así mismo, se determinará la posición y orientación de la red de triángulos con respecto al sistema de coordenadas de la cámara en cualquier instante de tiempo, para ello no se requerirá de ningún tipo de patrón de calibración, sino que realizará la estimación mediante el análisis de un campo de profundidad que será proporcionado por la cámara.

El algoritmo se va a diseñar e implementar en el lenguaje de programación C, haciendo uso de la plataforma de programación ROS, en su versión Indigo, bajo el sistema operativo Ubuntu 14.04 LTS, mediante un paquete denominado test con un nodo llamado $triangle_net_node$. La salida del algoritmo, estaría constituida por dos archivos de texto. El primero, contiene el número de vértices de la red; seguido por la lista de posiciones tridimensionales de los vértices respecto al sistema de coordenadas local de la red; seguido por el número de triángulos de la red; y finalmente, seguido por la lista de los índices de los vértices que conforman cada uno de los triángulos.

El segundo, deberá contener el vector tridimensional que describe la posición de la red de triángulos respecto al sistema de coordenadas de la cámara estereoscópica; seguido de la matriz de rotación que describe la orientación de la red con respecto al mismo sistema de coordenadas.

Capítulo 2

Objetivos

2.1. Objetivo general

Desarrollar un nodo ROS que sea capaz de describir la forma irregular de la porción de la superficie que está siendo capturada por una cámara estereoscópica, mediante una red rígida de triángulos constituida por más de cuatro vértices y dos triángulos, y que determine la posición y orientación de la red de triángulos con respecto al sistema de coordenadas de la cámara en cualquier instante de tiempo.

2.2. Objetivos específicos

- Instalar el Sistema Operativo Robótico (ROS).
- Interpretar el campo de profundidad generado por una cámara estereoscópica.
- Estimar las posiciones tridimensionales de cada uno de los vértices que constituyen red de triángulos en el espacio a partir de una nube de puntos.
- Asignar los índices de los vértices que conforman cada uno de los triángulos de la red de triángulos.
- Estimar los parámetros que describen la posición y orientación de la red de triángulos.
- Implementar un nodo ROS que sea capaz de suscribirse a un tópico, realizar los cálculos necesarios a partir de los mensajes recibidos, y colocar los resultados en dos archivos finales.

Capítulo 3

Metodología

- Estudiar, instalar y probar el sistema operativo Ubuntu 14.04 LTS.
- Instalar, estudiar y probar la plataforma de programación ROS, en su versión Indigo (ROS-Indigo).
- Estudiar el funcionamiento de la cámara estereoscópica real modelo BumbleBee2 de la marca Point Grey.
- Estudiar, instalar, calibrar y probar en ROS-Indigo, bajo el sistema operativo Ubuntu 14.04 LTS, la cámara estereoscópica BumbleBee2.
- Estudiar, instalar y probar en ROS-Indigo, bajo el sistema operativo Ubuntu 14.04 LTS, el visualizador de datos RVIZ, para ver la nube de puntos generados por cámara estereoscópica BumbleBee2.
- Diseñar, implementar y probar, en ROS-Indigo bajo el sistema operativo Ubuntu 14.04 LTS, un nodo escrito en el lenguaje de programación C, capaz de escribir en un archivo de texto, cada cierta cantidad de tiempo, el número de puntos característicos y las posiciones tridimensionales de los mismos (nube de puntos), generados por la cámara estereoscópica, así como visualizarlos con el visor basado en OpenGl, desarrollado por el estudiante Greivin Oviedo.
- Diseñar, implementar y probar en ROS-Indigo, bajo el sistema operativo Ubuntu 14.04 LTS, un nodo escrito en el lenguaje de programación C, capaz de construir una red de triángulos que describa una porción de la forma tridimensional de la superficie en el campo de visión de la cámara estereoscópica BumbleBee2, partiendo del hecho de que la superficie podría ser irregular (no ser plana). La red de triángulos deberá estar constituida por más de 4 vértices y 2 triángulos, donde los vértices deberán ubicarse a lo largo de líneas de corte equidistante del campo de profundidad suministrado por la cámara estereoscópica; y deberá guardarse en un archivo de texto de la siguiente manera: primero el número de vértices de la red; seguido por la lista de posiciones tridimensionales de los vértices respecto al sistema de coordenadas local de la red; seguido por el número de triángulos de la red; y finalmente, seguido por la lista de los índices de los vértices que conforman cada uno de los

triángulos. Asimismo, en otro archivo de texto, deberá guardarse el vector tridimensional que describe la posición de la red de triángulos respecto al sistema de coordenadas de la cámara estereoscópica; seguido de la matriz de rotación que describe la orientación de la red con respecto al mismo sistema de coordenadas.

Capítulo 4

Estereoscopía

La estereoscopía hace referencia a cualquier técnica capaz de recoger información visual tridimensional y/o crear la sensación visual de relieve (ilusión de profundidad) mediante una imagen estereográfica, un estereograma o una imagen tridimensional. Las cámaras estereoscópicas se basan en estas técnicas para obtener imágenes tridimensionales y a partir de estas (imagen izquierda y derecha) es posible calcular la posición tridimensional de ciertos puntos de la imagen respecto a un sistema de coordenadas de referencia.

4.1. La cámara estereoscópica

Las cámaras estereoscópicas son un arreglo de dos cámaras que se encuentran separadas por una distancia fija, como se aprecia en la figura 6.1, y que se basan en un principio muy básico, el de la visión humana, para obtener imágenes tridimensionales y posiciones tridimensionales (estimación de profundidad).

El método utilizado por los seres humanos para determinar la profundidad del entorno que los rodea mediante la visión binocular es la estereoscopía. El cerebro de un ser humano es capaz de mezclar las imágenes que proporciona cada ojo (imagen izquierda e imagen derecha), las cuales tienen diferencias debido a la separación horizontal de los ojos, y es a partir esa disparidad que al mezclarlas se logra la percepción de la profundidad, esa sensación de relieve. Así pues, capturando dos imágenes con una separación como la de la visión humana, se está imitando el comportamiento de los ojos humanos, que es lo que tratan de hacer las cámaras estereoscópicas.

Las imágenes tridimensionales se obtienen mediante una búsqueda de puntos característicos[1] en las imágenes proporcionadas por las cámaras izquierda y derecha y se establecen correspondencias entre estos. Cuando se tienen todos estos puntos referidos al plano de la cámara, se puede calcular la posición tridimensional respecto a un sistema de coordenadas. Este procedimiento se explicará detalladamente más adelante.

[1]Un punto característico es un punto único, o casi único, es parametrizable de tal manera que se puede comparar con otros puntos en otra imagen. Un ejemplo de estos puntos son los que se encuentran en las esquinas, no en los bordes, ya que contienen suficiente información para poder ser seleccionados de un cuadro a otro. La definición de una esquina se basa en la matriz de las derivadas de segundo orden de las intensidades de la imagen. Para obtener más detalles sobre los puntos característicos, consultar [4] y [2].

La cámara estereoscópica más común es la que está compuesta por dos lentes (uno izquierdo y uno derecho) y los sistemas de coordenadas de estos encuentran alineados, como se aprecia en la siguiente figura:

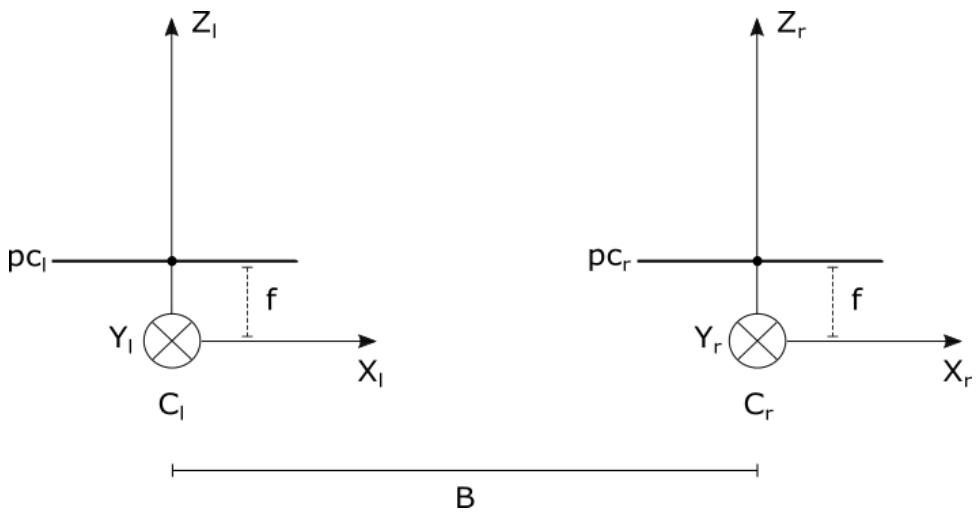


Figura 4.1: Diagrama de una cámara estereoscópica con dos lentes y sus sistemas de coordenadas alineados.

donde:

- C_l es la cámara izquierda.
- C_r es la cámara derecha.
- pc_l es el plano de la cámara izquierda.
- pc_r es el plano de la cámara derecha.
- (X_l, Y_l, Z_l) representa el sistema de coordenadas de la cámara izquierda.
- (X_r, Y_r, Z_r) representa el sistema de coordenadas de la cámara derecha.
- B: es la línea base (distancia entre los orígenes de los sistemas de coordenadas de las cámaras).
- f es la distancia focal.

4.2. Pose de un objeto rígido

Cuando se trabaja con posiciones tridimensionales obtenidas a partir de las imágenes capturadas por una cámara estereoscópica, se quiere que todos los puntos $(x, y, z)^T$ estén referidos a un

mismo sistema de coordenadas. Por ello, para saber la pose de un objeto, se necesita describir la posición y orientación de todos los puntos respecto a un sistema de coordenadas común que se le conoce como referencia.

4.2.1. Modelo de posición de un objeto rígido

Se tienen dos sistemas de coordenadas, $(X, Y, Z)^T$ representa el sistema de coordenadas de referencia y $(U, V, W)^T$ representa el sistema de coordenadas local del objeto, estos se pueden apreciar en la figura 4.2.

La posición del objeto con respecto al sistema de coordenadas de referencia se representa con el vector tridimensional $\mathbf{G}$, compuesto por las coordenadas $(G_x, G_y, G_z)^T$. Este vector se extiende desde el origen del sistema de coordenadas de referencia, hacia otro punto extremo que corresponde al origen del sistema de coordenadas local del objeto, como se observa en la figura 4.2.

Si los dos sistemas se encuentran alineados, se puede conocer la posición respecto al sistema de coordenadas de referencia de un punto $\mathbf{H}^*$ que se encuentra referido al sistema de coordenadas local del objeto, esto aplicando la ecuación (4.1).

$$\mathbf{H} = \mathbf{H}^* + \mathbf{G} \tag{4.1}$$

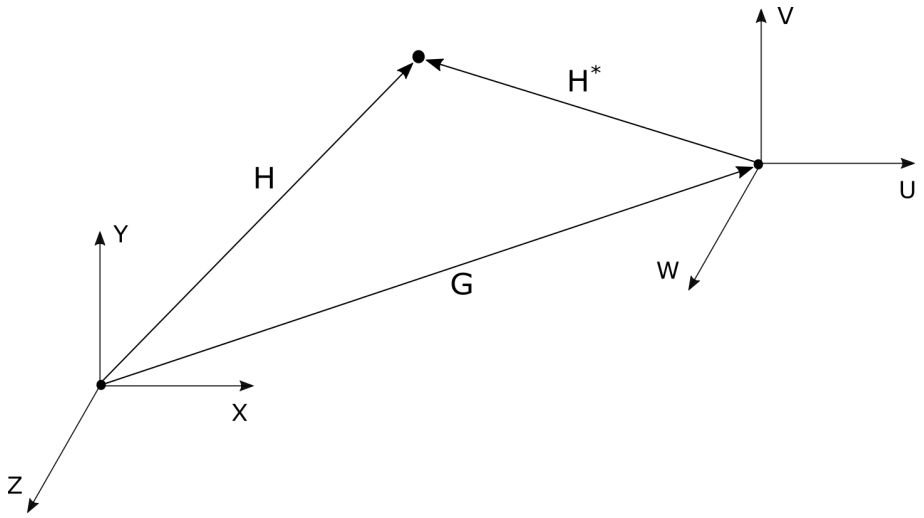


Figura 4.2: Posición de un objeto rígido cuando ambos sistemas de coordenadas están alineados.

4.2.2. Modelo de orientación de un objeto rígido

La orientación que tiene un objeto rígido se puede describir mediante la rotación de su sistema de coordenadas, $(U, V, W)^T$, con respecto a cada uno de los ejes que conforman el sistema de

coordenadas de referencia, $(X, Y, Z)^T$. Donde se parte del hecho de que primero se gira todo el sistema de coordenadas respecto al eje X, luego respecto al eje Y y por último, respecto al eje Z.

La orientación de un objeto se puede representar con la matriz 3x3 de la ecuación (4.2), se le conoce como la matriz de rotación y sus entradas dependen de los tres ángulos de orientación alrededor de cada eje.

$$\mathbf{R} = \begin{pmatrix} r_1 & r_2 & r_3 \\ r_4 & r_5 & r_6 \\ r_7 & r_8 & r_9 \end{pmatrix} \tag{4.2}$$

Las entradas que conforman la matriz de rotación se pueden obtener con las ecuaciones de la (4.3) a la (4.11), estas ecuaciones son función de los ángulos de orientación $\mathbf{O} = (\Re_X, \Re_Y, \Re_Z)^T$ respecto a cada uno de los ejes del sistema de coordenadas de referencia.

$$r_1 = cos(\Re_Y) \cdot cos(\Re_Z) \tag{4.3}$$

$$r_2 = sin(\Re_X) \cdot sin(\Re_Y) \cdot cos(\Re_Z) - cos(\Re_X) \cdot sin(\Re_Y) \tag{4.4}$$

$$r_3 = cos(\Re_X) \cdot sin(\Re_Y) \cdot cos(\Re_Z) + sin(\Re_X) \cdot sin(\Re_Z) \tag{4.5}$$

$$r_4 = cos(\Re_Y) \cdot sin(\Re_Z) \tag{4.6}$$

$$r_5 = sin(\Re_X) \cdot sin(\Re_Y) \cdot sin(\Re_Z) + cos(\Re_X) \cdot cos(\Re_Z) \tag{4.7}$$

$$r_6 = cos(\Re_X) \cdot sin(\Re_Y) \cdot sin(\Re_Z) - sin(\Re_X) \cdot cos(\Re_Z) \tag{4.8}$$

$$r_7 = -sin(\Re_Y) \tag{4.9}$$

$$r_8 = sin(\Re_X) \cdot cos(\Re_Y) \tag{4.10}$$

$$r_9 = cos(\Re_X) \cdot cos(\Re_Y) \tag{4.11}$$

4.2.3. Modelo de pose de un objeto rígido

Si se conoce la pose del objeto, es decir, si se conoce la posición y orientación de este, se puede obtener la posición de cualquier punto del objeto con respecto al sistema de coordenadas de referencia (como se aprecia en la figura 4.3) haciendo uso de la ecuación (4.12) que se conoce como la ecuación de pose.

$$\mathbf{H} = \mathbf{R} \cdot \mathbf{H}^* + \mathbf{G} \tag{4.12}$$

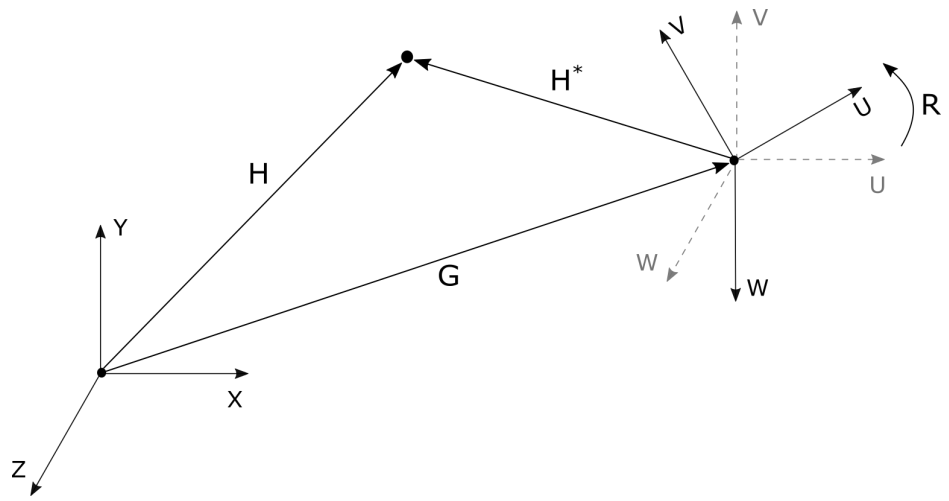


Figura 4.3: Posición de cualquier punto del objeto con respecto al sistema de coordenadas de referencia.

4.3. Cálculo de la posición tridimensional de un punto utilizando una cámara estereoscópica

Para calcular la posición tridimensional de un punto respecto a un sistema de coordenadas de referencia (en este caso es el sistema de coordenadas de la cámara izquierda) se debe de conocer la posición y orientación del sistema de coordenadas de la cámara derecha con respecto al sistema de referencia.

Como se mencionó antes, para la mayoría de las cámaras estereoscópicas ambos sistemas de coordenadas se encuentran alineados y sus orígenes están separados por una distancia B conocida como línea base (observar la figura 4.5). Por lo tanto, la matriz de rotación que describe la orientación del sistema de coordenadas de la cámara derecha con respecto a la referencia es la matriz identidad dada por la ecuación (4.13), es decir, los tres ángulos de orientación son 0°.

$$\mathbf{R}_r^l = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{4.13}$$

Además, al aplicar la ecuación de pose para $\mathbf{H}^* = (0, 0, 0)$ (origen) y $\mathbf{G} = (B, 0, 0)$ se obtiene la expresión (4.14) que describe la posición del sistema de coordenadas de la cámara derecha con respecto al sistema de coordenadas de referencia.

$$\begin{pmatrix} X_r^l \\ Y_r^l \\ Z_r^l \end{pmatrix} = \begin{pmatrix} B \\ 0 \\ 0 \end{pmatrix} \tag{4.14}$$

Luego de que se conoce la posición y orientación del sistema de coordenadas de la cámara derecha, se busca un punto característico sobre el plano de la imagen de la cámara izquierda y se traza una línea, conocida como línea vista izquierda (se denota como l_l), que va desde el origen del sistema de coordenadas de la cámara izquierda, pasando por el plano de la cámara en el punto característico $(u_i, v_i)^T$ ubicado en el píxel $(c_i, r_i)^T$ de la imagen, luego por el punto característico respecto a la cámara $\mathbf{P}_i = (u_i, v_i, f)^T$ (f es la distancia focal) y se extiende hasta infinito. Lo descrito anteriormente se aprecia en la figura 4.4 y en la figura 4.5.

Para poder definir la ecuación que describe la línea vista, se tiene que realizar una transformación del sistema de coordenadas de la imagen en píxeles $(c, r)^T$, donde c apunta hacia la derecha, v apunta hacia abajo y el origen se encuentran en la esquina superior izquierda de la imagen, al sistema de coordenadas del plano de la cámara en unidades métricas $(u, v)^T$ cuyo origen se encuentra en el píxel central de la imagen. Esto se resume de forma visual en la figura 4.4. Con las ecuaciones (4.15) y (4.16) se puede realizar la transformación al sistema de coordenadas $(c, r)^T$.

$$u_i = \frac{N_{c,u}}{N_{f,u}} \cdot d_u \cdot (c_i - C_c) \tag{4.15}$$

Donde:

- u_i es el valor en centímetros sobre el eje u del plano de la cámara.
- $N_{c,u}$ es el número de sensores por línea.
- $N_{f,u}$ es el número de píxeles por línea.
- d_u es la distancia entre sensores en u (en centímetros).
- c_i es el valor en píxel sobre el eje c de la imagen.
- C_c es el ancho de la imagen dividido entre dos.

$$v_i = d_v \cdot (r_i - C_r) \tag{4.16}$$

Donde:

- v_i es el valor en centímetros sobre el eje v del plano de la cámara.
- d_v es la distancia entre sensores en v (en centímetros).
- r_i es el valor en píxel sobre el eje r de la imagen.

- C_r es el alto de la imagen dividido entre dos.

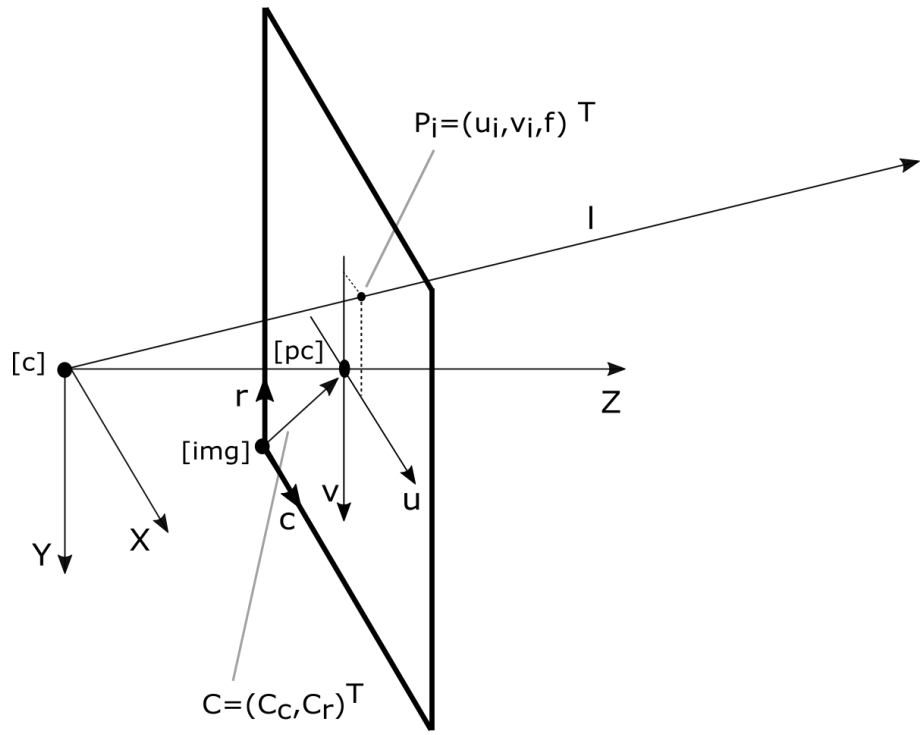


Figura 4.4: Sistema de coordenadas de la imagen (c,r), del plano de la cámara (u,v) y de la cámara (x,y,z).

Una vez que se tiene el valor del punto característico en el sistema de coordenadas del plano de la cámara $(u, v)^T$, se define la ecuación (4.17) que corresponde a la ecuación de una recta que describe a la línea vista izquierda. Esta pasa por el origen y por el punto $(c_{i,l}, r_{i,l}, f)$ que representa al punto característico respecto al sistema de coordenadas de la cámara izquierda; α_l es el factor de magnitud y f es la distancia focal que es igual para ambas cámaras.

$$\begin{pmatrix} x_l \\ y_l \\ z_l \end{pmatrix} = \alpha_l \begin{pmatrix} c_{i,l} \\ r_{i,l} \\ f \end{pmatrix}, \quad 0 \leq \alpha_l \leq +\infty \tag{4.17}$$

Para la cámara derecha se realiza el mismo procedimiento, primero se busca el punto característico encontrado en la imagen de la cámara izquierda en la imagen de la cámara derecha para establecer una correspondencia. Una vez que se identificó el punto y que se pasó al sistema de coordenadas de la cámara derecha, se traza la línea vista derecha (l_r) como se observa en la figura 4.20, que va desde el origen, pasando por el punto característico, hasta infinito. La ecuación (4.18) corresponde a la ecuación de una recta que describe a la línea vista derecha.

$$\begin{pmatrix} x_r \\ y_r \\ z_r \end{pmatrix} = \alpha_r \begin{pmatrix} c_{i,r} \\ r_{i,r} \\ f \end{pmatrix}, \quad 0 \leq \alpha_r \leq +\infty \tag{4.18}$$

Como se conoce la posición y orientación del sistema de coordenadas de la cámara derecha respecto al sistema de coordenadas de referencia, se puede referir la ecuación que describe a la línea vista derecha al sistema de coordenadas de referencia utilizando la ecuación de pose (4.12), el resultado es la ecuación (4.19).

$$\begin{pmatrix} x_r^l \\ y_r^l \\ z_r^l \end{pmatrix} = \alpha_r \begin{pmatrix} c_{i,r} \\ r_{i,r} \\ f \end{pmatrix} + \begin{pmatrix} B \\ 0 \\ 0 \end{pmatrix}, \quad 0 \leq \alpha_r \leq +\infty \tag{4.19}$$

Ahora se debe de encontrar el valor de α_l y de α_r que satisfacen la ecuación (4.20), lo que quiere decir que la línea vista derecha referida al sistema de coordenadas de la cámara izquierda se interseca en un punto en común **H** con la línea vista izquierda, como se observa en la figura 4.5. Finalmente, ese punto de intersección corresponde a la posición tridimensional del punto característico respecto al sistema de coordenadas de la cámara izquierda.

$$\begin{pmatrix} x_r^l \\ y_r^l \\ z_r^l \end{pmatrix} = \begin{pmatrix} x_l \\ y_l \\ z_l \end{pmatrix} \tag{4.20}$$

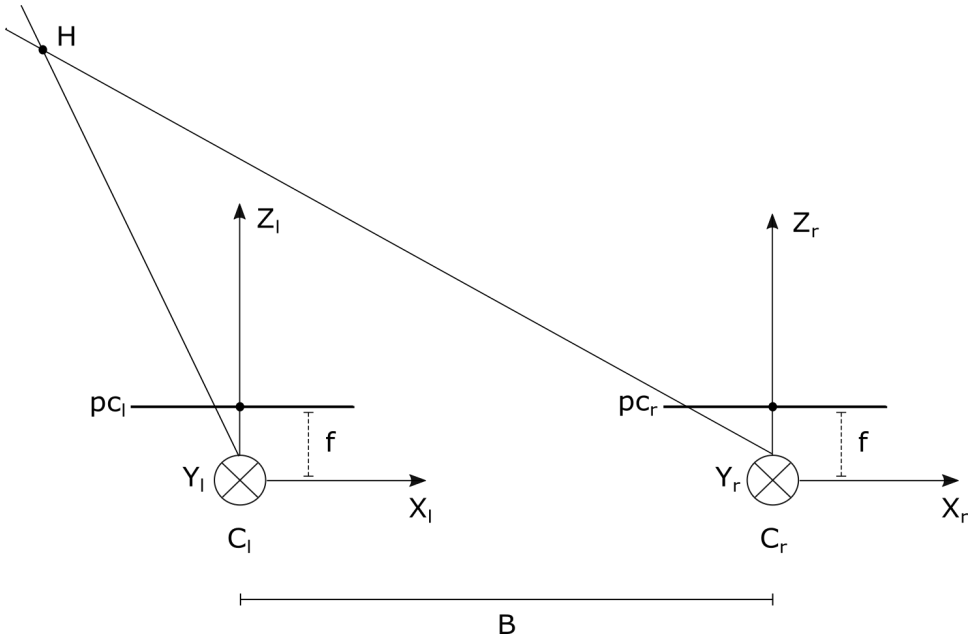


Figura 4.5: Calculo de la posición tridimensional de un punto a partir de la intersección de la línea vista izquierda con la derecha.

4.3.1. Obtener una nube de puntos utilizando una cámara estereoscópica

Si todo el procedimiento anterior se repite para un conjunto de puntos característicos en la imagen izquierda y derecha, se obtiene lo que se conoce como una "nube de puntos", como se aprecia en la figura 4.6. La nube de puntos es simplemente un conjunto de puntos de los cuales se conoce sus posiciones tridimensionales respecto al sistema de coordenadas de la imagen izquierda y, además, cada uno de estos puntos está asociado a un punto característico en la imagen izquierda. Además, como se observa en la figura 4.6, puede haber píxeles cuya posición tridimensional se desconozca, esto debido a que no tienen asociados un punto característico y por lo tanto, no se puede realizar el procedimiento para el cálculo de la posición tridimensional.

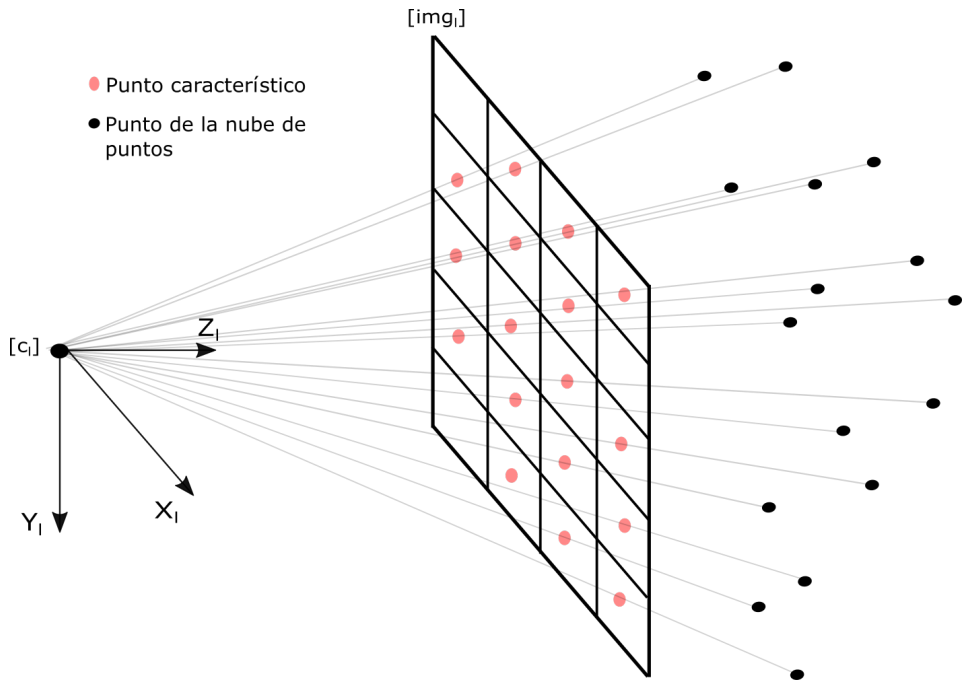


Figura 4.6: Nube de puntos respecto al sistema de coordenadas de referencia (cámara izquierda).

4.4. Implementación usando el Sistema Operativo Robótico (ROS)

El sistema operativo robótico (ROS[2]) es un marco o plataforma flexible para escribir software para robots. La principal característica es que permite la comunicación e interacción entre procesos, conocidos como nodos.

Para obtener las posiciones tridimensionales respecto al sistema de coordenadas de la cámara izquierda de cada uno de los puntos que conforman la nube de puntos, es decir, para obtener la nube de puntos a partir del arreglo estereoscópico se utilizaron tres nodos de ROS de código abierto.

El primer nodo llamado camera1394stereo es el encargado de capturar o tomar las imágenes de la cámara izquierda y de la cámara derecha.

El segundo nodo utilizado se llama camera_calibration, este nodo se encarga de calibrar el arreglo estereoscópico (la cámara izquierda y derecha) mediante el método de calibración de Zhang [6]. La calibración es de suma importancia para lograr una extracción precisa de la información métrica a partir de las imágenes proporcionadas por el arreglo estereoscópico.

El ultimo nodo llamado stereo_image_proc lo que hace es tomar los parámetros finales obtenidos a partir de la calibración del arreglo estereoscópico para corregir cualquier distorsión y empieza a buscar puntos característicos en las imágenes capturadas por la cámara izquierda y por

[2]Un mayor desarrollo sobre el funcionamiento de ROS se puede encontrar en el apéndice A.

la cámara derecha, para posteriormente buscar la posición tridimensional del punto respecto al sistema de coordenadas de la cámara izquierda. Lo anterior lo repite para un conjunto de puntos característicos para obtener finalmente la nube de puntos.

En la figura 4.7 se muestra un diagrama que describe la forma en que se implementan cada uno de los nodos en ROS para obtener la nube de puntos.

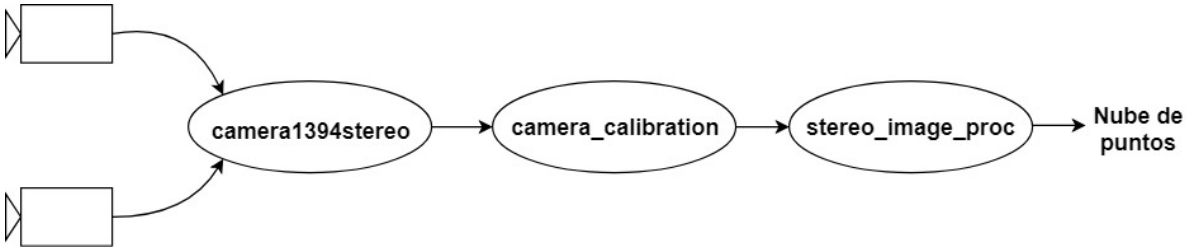


Figura 4.7: Implementación de nodos de ROS para obtener la nube de puntos.

Capítulo 5

Generar una red de triángulos a partir de una nube de puntos

A partir de la nube de puntos, es decir, a partir del conjunto de puntos de los cuales se conoce sus posiciones tridimensionales respecto al sistema de coordenadas de referencia (cámara izquierda), dados por las imágenes tomadas con la cámara estereoscópica (imagen izquierda e imagen derecha) mediante el método descrito anteriormente de corte de líneas vistas pasando por puntos característicos y usando el método iterativo conocido como RANSAC (random sample consensus), junto con un método que se basa en el corte de líneas equidistantes del campo de profundidad, es posible generar un red de triángulos que estima la forma tridimensional de la porción de la superficie que está siendo capturada por la cámara.

Entonces en este capítulo se describe todo el procedimiento matemático para describir la forma irregular de la superficie en exploración mediante una red rígida de triángulos constituida por más de cuatro vértices y dos triángulos, donde los vértices se ubicarán a lo largo de líneas de corte equidistante del campo de profundidad suministrado por la cámara estereoscópica. Además, se describe el procedimiento para determinar la posición y orientación de la red de triángulos con respecto al sistema de coordenadas de la cámara izquierda.

5.1. Definición de una red de triángulos

Una red de triángulos, también conocida como red o malla irregular triangulada (TIN, por sus siglas en ingles), es una representación de una superficie continua mediante facetas triangulares [5]. Dicho de otra manera, es la conexión de una serie de puntos mediante una red de triángulos irregulares, de tal forma que los vértices se corresponden con dichos puntos y, además, se conocen las coordenadas x, y y z de donde se localizan con respecto al sistema de coordenadas local.

Entonces, una red de triángulos es una representación de la superficie terrestre física, formada por nodos y líneas distribuidas irregularmente con coordenadas tridimensionales (x, y, z) que están dispuestas en una red de triángulos no superpuestos.

La red de triángulos más simple posible es la que se observa en la figura 5.1, la cual está compuesta por cuatro vértices y dos triángulos. Esta red describe o modela una forma rectangular de dimensiones nxm.

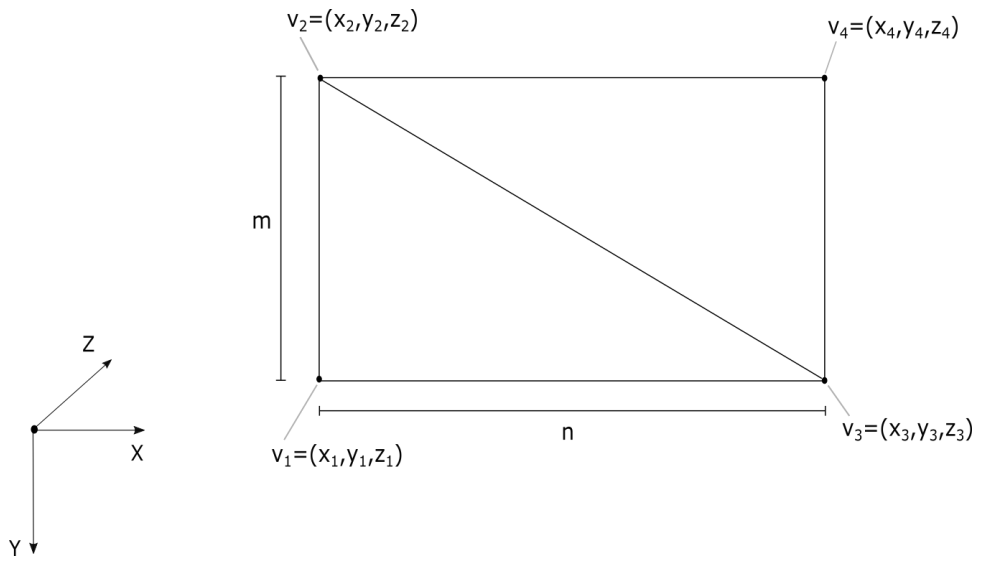


Figura 5.1: Rectángulo modelado con una red de triángulos.

5.2. División de la imagen izquierda en líneas verticales

Todo el procedimiento matemático y los cálculos necesarios para generar una red de triángulos a partir de una nube de puntos, se basa en el hecho de considerar que la imagen de la cámara izquierda (el sistema de coordenadas de esta cámara es el de referencia) se encuentra dividida en líneas verticales, es decir, en columnas, como se observa en la figura 5.2. Entonces el número de columnas será exactamente igual al ancho de la imagen (por ejemplo, 640 columnas para una imagen VGA 640x480) y el número de píxeles por cada columna o línea vertical será exactamente igual al alto de la imagen (por ejemplo, 480 píxeles por columna para una imagen VGA 640x480).

Entonces esta consideración surge debido a que la mayoría de los cálculos que se tienen que realizar para generar la red de triángulos se hacen por columna, es decir, el procedimiento matemático se repite para ciertas columnas de la imagen, esto se verá en detalle más adelante.

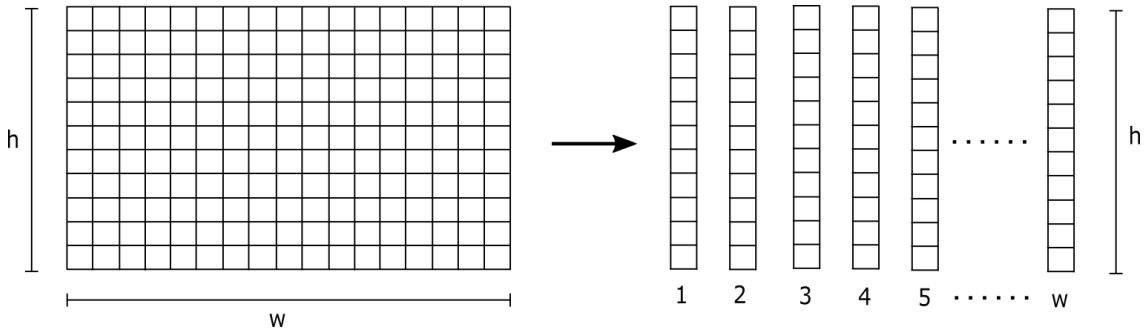


Figura 5.2: División de la imagen izquierda en líneas verticales.

Considérese una sola linea vertical (columna) de la imagen izquierda, que como se mencionó anteriormente, va a tener un numero de píxeles exactamente igual al alto de la imagen. Además, como se observa en la figura 5.3, cada una de estas columnas van a estar asociadas a una porción única e irrepetible de la nube de puntos, es decir, para cada píxel de una columna se conoce su posición tridimensional en el espacio con respecto al sistema de coordenadas de la cámara izquierda y por lo tanto el conjunto de píxeles de una columna corresponde a un conjunto único de puntos tridimensionales de la nube de puntos. Cabe destacar que puede haber píxeles cuya posición tridimensional se desconozca, entonces se parte del hecho de que se conoce el el número de puntos tridimensionales por columna y los píxeles que no tiene una posición tridimensional asociada.

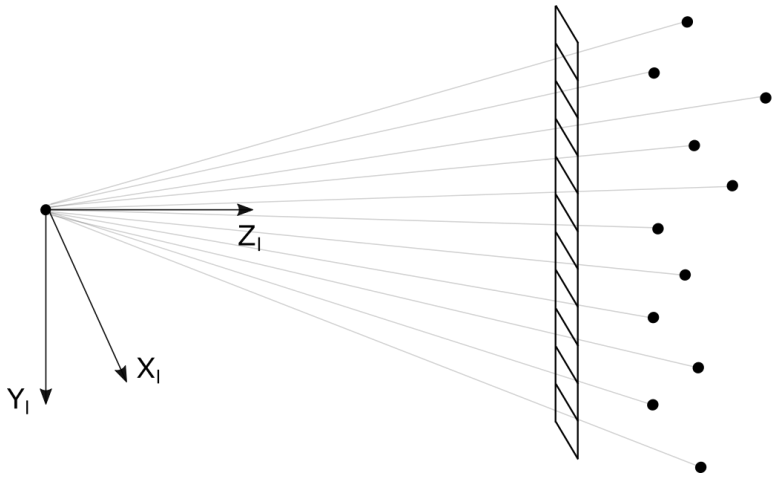


Figura 5.3: Conjunto de píxeles de una columna asociados a un conjunto único de puntos de la nube de puntos.

Por otro lado, los cálculos necesarios para generar la red de triángulos no se realizan en cada una de las columnas de la imagen, solamente en ciertas columnas. Entonces para elegir estas columnas se utilizan tres conceptos.

El primero es el factor de columna (FC) que se define como el porcentaje (expresado en forma decimal de 0 a 1) de líneas verticales que serán parte de los cálculos necesarios para llegar a obtener la red de triángulos. Esta dado por la ecuación (5.1), donde W es el ancho de la imagen (número total de columnas) y NC es el numero de líneas verticales que se desea que tenga la red de triángulos final.

El segundo es el paso entre columnas (PEC) que se define como el numero de columnas que hay entre dos columnas consecutivas que se eligieron, es decir, cada cuantas columnas se va a elegir una nueva columna que sea parte de los cálculos para obtener la red de triángulos. Esta dado por la ecuación (5.2), donde $\lfloor x \rfloor$ es la función floor que da como resultado el mayor entero menor o igual que x, es decir, se redondea hacia abajo el factor de línea.

El ultimo es el numero de columna inicial (NCI) que se define como el numero de la columna

de la imagen izquierda a partir de la cual se debe de empezar a realizar los respectivos cálculos para obtener la red de triángulos. Esta dada por la ecuación (5.3), donde $\lceil x \rceil$ es la función ceil que da como resultado al menor entero mayor o igual que x, es decir, redondea hacia arriba.

$$FC = \frac{W}{NC} \tag{5.1}$$

$$PEC = \lfloor FC \rfloor \tag{5.2}$$

$$NCI = \lceil \frac{W - PEC \cdot (NC - 1) - 1}{2} \rceil \tag{5.3}$$

Entonces las columnas de la imagen izquierda que se van a considerar a la hora de realizar los cálculos necesarios para obtener la red de triángulos se obtienen a partir de los conceptos anteriores, con la ecuación 5.4 que proporciona el numero de columna de la imagen para un i dado que va desde cero hasta el numero de líneas verticales menos uno.

$$columna_i = NCI + PEC \cdot i, \quad 0 \leq i \leq NC - 1 \tag{5.4}$$

5.3. Obtener la linea de mejor ajuste

Se desea encontrar la línea o recta en el espacio con respecto al sistema de coordenadas de la cámara izquierda (referencia) que mejor se ajusta al conjunto de puntos tridimensionales de la nube de puntos que corresponde o se asocia con el conjunto de píxeles de una columna de la imagen izquierda que se eligió con la ecuación 5.4, esto como se observa en la figura 5.4. Se requiere que mínimo dos puntos de la porción de la nube de puntos pertenezcan a esta recta o línea en el espacio, de tal forma que si esta recta se proyecta al plano de la cámara y luego se pasa al sistema de coordenadas de la imagen corresponda a la columna a partir de la cual esta se obtuvo. Cabe destacar que esta línea de mejor ajuste es necesaria para posteriormente poder encontrar los vértices que componen la red de triángulos.

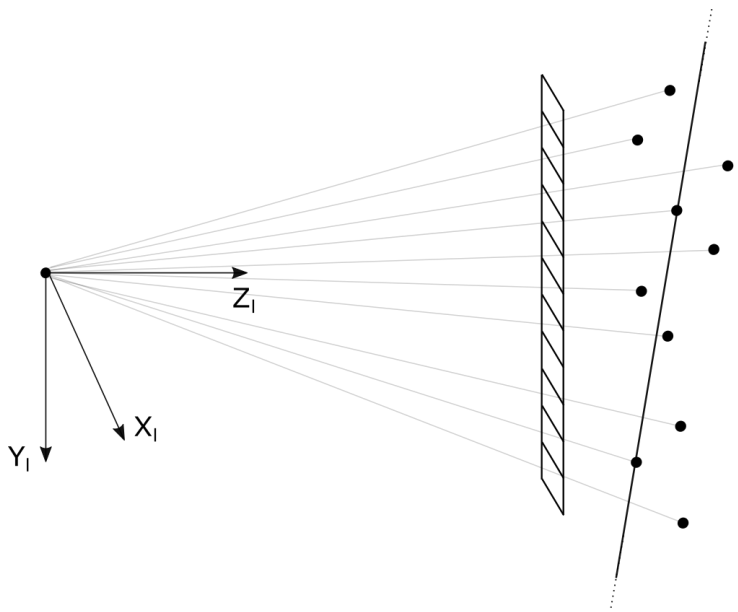


Figura 5.4: Línea que mejor se ajusta a la porción de la nube de puntos asociada a una columna de la imagen.

Para calcular la ecuación de la recta en el espacio, con respecto al sistema de coordenadas de referencia, que mejor se ajusta a una porción de la nube puntos asociada a una columna, es decir, para obtener la línea de mejor ajuste, se utiliza el algoritmo de RANSAC[1]. Este método se utiliza para estimar los parámetros de un modelo matemático (en este caso la ecuación de una recta) a partir de un conjunto de datos (en este caso la porción de la nube de puntos) y que se ajusta de manera óptima a estos.

El primer paso es seleccionar aleatoriamente dos puntos de la porción de la nube de puntos que se está considerando, a estos dos puntos se les llamara P_1 y P_2 y sus posiciones tridimensionales están dadas por (x_1, y_1, z_1) y (x_2, y_2, z_2), respectivamente. A partir de estos puntos se obtiene la ecuación (5.5), que corresponde al modelo matemático de una recta que se ajusta a los dos puntos elegidos aleatoriamente, es decir, los puntos pertenecen a la recta. Este primer paso se puede apreciar en la figura 5.5.

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \alpha \cdot \begin{pmatrix} x_2 - x_1 \\ y_2 - y_1 \\ z_2 - z_1 \end{pmatrix} + \begin{pmatrix} x_1 \\ y_1 \\ z_1 \end{pmatrix}, \quad -\infty \leq \alpha \leq +\infty \tag{5.5}$$

[1]El fundamento teórico del algoritmo de RANSAC se puede encontrar en el apéndice B.

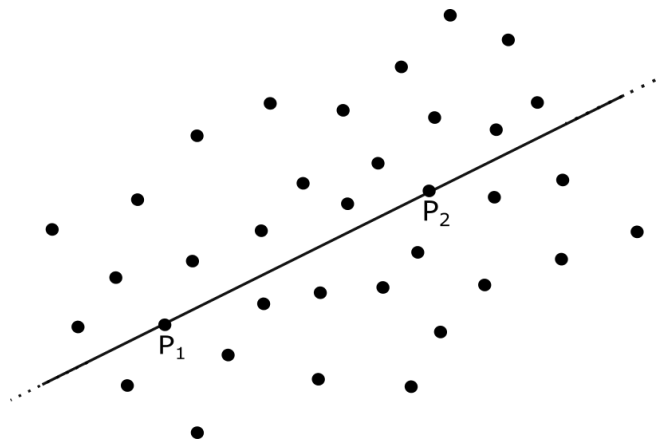


Figura 5.5: Recta que se ajusta a dos puntos elegidos aleatoriamente de la porción de la nube de puntos a considerar.

El segundo paso consiste en verificar cuales puntos de todo el conjunto de puntos (porción de la nube de puntos) son consistentes con la ecuación de la recta obtenida en el primer paso, ecuación (5.5). Aquellos puntos que se ajustan bien a la recta estimada y que por lo tanto se clasifican como parte del conjunto de consenso (inliers), son aquellos que cumplen con una cierta condición, en este caso esa condición consiste en que la distancia del punto a la recta estimada tiene que ser menor que una distancia máxima fija (d_{max}) para ser considerado un inlier, esto se observa en la figura 5.6.

Entonces se quiere encontrar la distancia entre un punto (x_i, y_i, z_i) de la porción de la nube de puntos y la recta estimada, para ello se define un vector $\mathbf{Q}$ que va del punto de interés a un punto de la recta y está dado por la ecuación 5.6.

$$\mathbf{Q} = \begin{pmatrix} x_1 - x_i \\ y_1 - y_i \\ z_1 - z_i \end{pmatrix} = \begin{pmatrix} q_x \\ q_y \\ q_z \end{pmatrix} \tag{5.6}$$

Luego se obtiene el vector $\mathbf{V}$ dado por la ecuación (5.8), este vector es perpendicular al vector $\mathbf{Q}$ y al vector director de la recta $\mathbf{V}_{dir}$ (ecuación (5.7)) y se obtiene mediante el producto cruz de esos dos vectores.

$$\mathbf{V}_{dir} = \begin{pmatrix} x_1 - x_2 \\ y_1 - y_2 \\ z_1 - z_2 \end{pmatrix} = \begin{pmatrix} v_{dir-x} \\ v_{dir-y} \\ v_{dir-z} \end{pmatrix} \tag{5.7}$$

$$\mathbf{V} = \begin{pmatrix} q_y \cdot v_{dir-x} - q_z \cdot v_{dir-y} \\ q_z \cdot v_{dir-x} - q_x \cdot v_{dir-z} \\ q_x \cdot v_{dir-y} - q_y \cdot v_{dir-x} \end{pmatrix} = \begin{pmatrix} v_x \\ v_y \\ v_z \end{pmatrix} \tag{5.8}$$

Se obtiene el módulo del vector $\mathbf{V}$ y el módulo del vector director de la recta $\mathbf{V}_{dir}$, esto mediante las ecuaciones (5.9) y (5.10) respectivamente.

$$|\mathbf{V}| = \sqrt{v_x^2 + v_y^2 + v_z^2} \tag{5.9}$$

$$|\mathbf{V}_{dir}| = \sqrt{v_{dir-x}^2 + v_{dir-y}^2 + v_{dir-z}^2} \tag{5.10}$$

Finalmente, la distancia entre un punto (x_i, y_i, z_i) del conjunto de puntos y la recta estimada dada por la ecuación (5.5) se calcula con la ecuación (5.11), que corresponde al módulo del vector $\mathbf{V}$ dividido entre el módulo del vector $\mathbf{V}_d ir$.

$$d_i = \frac{|\mathbf{V}|}{|\mathbf{V}_{dir}|} \tag{5.11}$$

Entonces un punto es considerado un inlier si se cumple con la condición dada por la expresión (5.12), como se observa en la figura 5.6.

$$di \leq d_{max} \longrightarrow P_i = (x_i, y_i, z_i) \in [inliers] \tag{5.12}$$

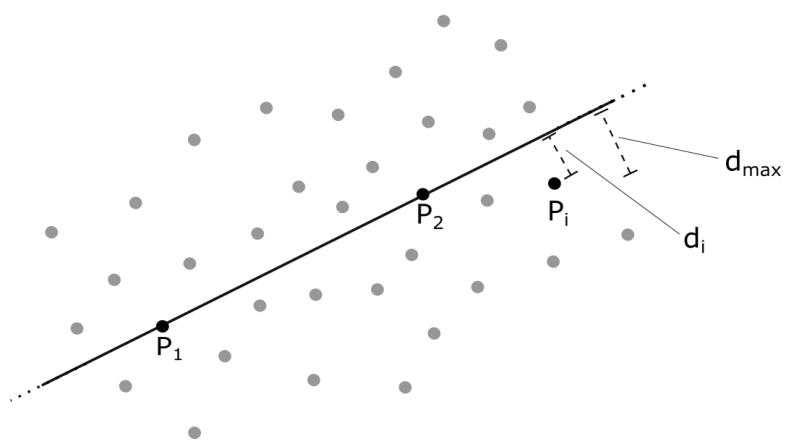


Figura 5.6: Distancia entre un punto de la porción de la nube de puntos y la recta estimada.

Si se calcula la distancia que existe entre cada uno de los puntos del conjunto de puntos y la recta estimada y esa distancia se va comparando con la condición de la expresión (5.12), se va a tener como resultado el número total de puntos que se consideran parte del conjunto de consenso, inliers (denotado como in_T).

El modelo estimado (ecuación 5.5) es razonablemente bueno si el número total de puntos clasificados como inliers (in_T) cumplen con una condición mínima, es decir, con un valor mínimo denotado como in_{min}. Entonces, si se cumple la condición dada por la expresión (5.13) se finaliza

el procedimiento y quiere decir que la ecuación 5.5 es suficientemente buena para ser considerada como la ecuación de la línea de mejor ajuste, esto se ejemplifica en la figura 5.7.

$$in_T \geq in_{min} \longrightarrow fin \tag{5.13}$$

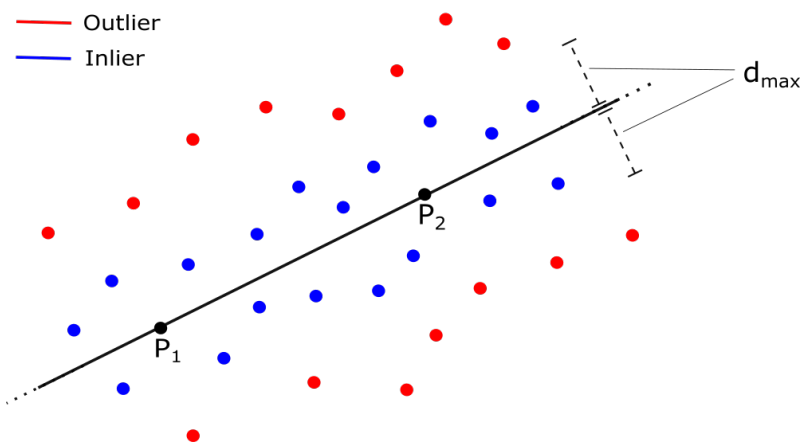


Figura 5.7: Obtención de la línea que mejor se ajuste a una porción de la nube de puntos mediante el algoritmo de RANCSA.

Si no se cumple con la condición dada por la expresión (5.13), se debe repetir todo el procedimiento anterior un número fijo de veces (N veces) hasta encontrar un modelo que cumpla con la condición. En caso de que el procedimiento se haya repetido las N veces y no se cumplió la condición, se elige como mejor modelo aquel que tiene el mayor número de inliers (el mayor n_T), entonces se debe de ir conservando el modelo con el mayor in_T.

Todo el procedimiento descrito anteriormente se debe de realizar para cada una de las columnas de la imagen que se vayan eligiendo con la ecuación 5.4. Entonces al final cada una de las columnas consideradas van a tener asociadas la ecuación de una línea que mejor se ajusta a la respectiva porción de la nube de puntos.

5.4. Obtener los vértices a partir de la línea de mejor ajuste

Ahora que se conoce para cada columna elegida la ecuación de la línea que mejor se ajusta a la respectiva porción de la nube de puntos, se puede proceder a encontrar y seleccionar los vértices a partir de los cuales se construye cada uno de los triángulos que constituyen la red de triángulos.

5.4.1. Obtener un segmento de la línea de mejor ajuste a partir de las líneas vistas que pasan por el primer y último píxel de la columna considerada

Como todas las líneas de mejor ajuste obtenidas son infinitas (el factor de magnitud α va de $-\infty$ a $+\infty$) es muy difícil obtener los vértices a partir de estas por que se tendrían que considerar los infinitos puntos que pertenecen a la línea. Entonces lo ideal sería encontrar un segmento de la línea de mejor ajuste que vaya de un α_1 a un α_2, como se aprecia en la figura 5.8.

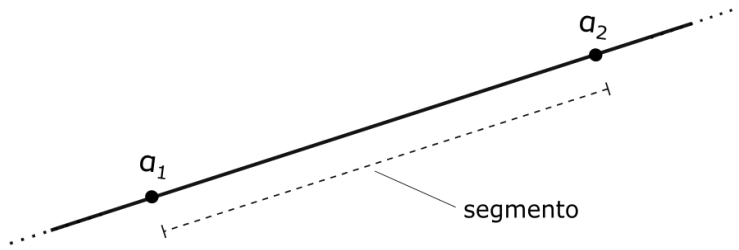


Figura 5.8: Segmento de la línea de mejor ajuste que va de un α_1 a un α_2.

La idea es que este segmento sea tal que cuando se proyecta al plano de la cámara y luego se pasa al sistema de coordenadas de la imagen, su longitud en píxeles corresponda a la longitud de la respectiva columna, donde la longitud en píxeles de cualquier columna es igual al alto de la imagen. Para que lo anterior se cumpla para cualquier columna dada por la ecuación 5.4, se debe de encontrar el α_1 que garantiza la distancia mínima entre la respectiva línea de mejor ajuste y una línea vista que pasa por el píxel p_1=$(columna_i, 0)^T$ de la imagen y análogamente se debe de encontrar el α_2 que garantiza la distancia mínima entre la línea de mejor ajuste y una línea vista que pasa por el píxel p_2=$(columna_i, H)^T$ de la imagen, donde H es el alto de la imagen. En la figura 5.9 se aprecia un resumen visual de lo antes mencionado.

Para hallar la menor distancia entre las dos rectas l y l_1 y para determinar el respectivo $alpha$ que pertenece a la recta l y al vector de menor modulo que une ambas rectas, se puede construir un plano que sea paralelo a las dos rectas y que contenga a la recta l. El vector normal a este plano ($\mathbf{A}_1$) debe de ser perpendicular a las dos rectas, por lo tanto, está dado por el producto vectorial del vector director de l con el vector director de l_1. Ahora bien, todos los puntos de la recta l_1 están a la misma distancia de este plano, entre ellos el más cercano a la recta l. Por tanto, la distancia entre las dos rectas es la distancia de la recta l_1 a este plano. La distancia de un punto a un plano es la proyección sobre el vector perpendicular al plano ($\mathbf{A}_1$) de un vector con un extremo en el punto y otro en el plano ($\mathbf{B}$). Entonces, este vector va de un punto de la recta l a un punto de la recta l_1. Así pues, la distancia entre dos rectas (d_{min1}) es el producto punto de $\mathbf{A}_1$ con $\mathbf{B}$, entre el módulo de $\mathbf{A}_1$. Luego, el vector de menor modulo ($\mathbf{N}_1$) que une las dos rectas es el que tiene la dirección de $\mathbf{A}_1$ con un módulo igual al valor de d_{min1}. Para determinar los dos puntos de las rectas l y l_1 que une este vector, se puede considerar los puntos genéricos dados por la ecuación de las rectas ($\mathbf{l}$ y $\mathbf{l}_1$. Así pues, el vector que une dos puntos cualesquiera de las

rectas ($\mathbf{D}_1$) es el vector que va de $\mathbf{l}_1$ a ($\mathbf{l}$. Los puntos que están más cerca uno de otro son tales que el vector $\mathbf{D}_1$ es paralelo a $\mathbf{N}_1$. Al imponer esa condición queda un sistema de tres ecuaciones con dos incógnitas ($\mathbf{N}_1$x$\mathbf{D}_1$=0). El determinante siempre será nulo, lo cual indica que una de las ecuaciones es combinación lineal de las otras. Eliminando una de ellas y resolviendo se obtiene finalmente el valor de *alpha*.

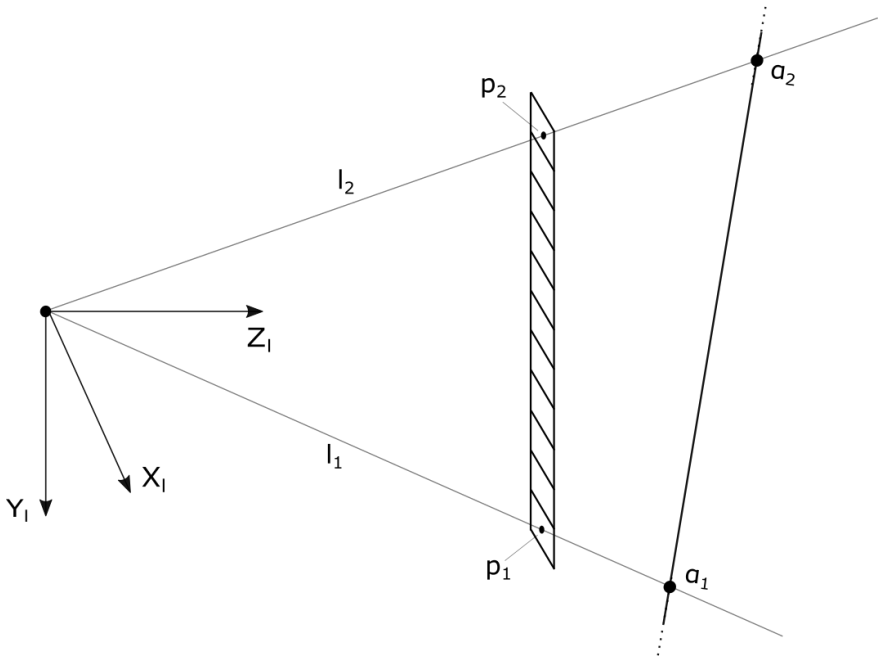


Figura 5.9: Segmento de la línea de mejor ajuste obtenido a partir de las líneas vistas.

Lo primero sería obtener la ecuación de la línea vista l_1 que pasa por el píxel p_1=$(columna_i, 0)^T$, para ello se debe de realizar una transformación del sistema de coordenadas de la imagen en píxeles al sistema de coordenadas del plano de la cámara en unidades métricas, como se vio en el capítulo 3. Entonces dicha transformación se puede realizar utilizando la ecuación (4.15) para c_i=$columna_i$ y la ecuación (4.16) para r_i=0. La posición del píxel p_1 respecto al sistema de coordenadas del plano de la cámara está dada por la expresión (5.14).

$$\mathbf{p}_1^{pc} = \begin{pmatrix} \frac{N_{c,u}}{N_{f,u}} \cdot d_u \cdot (columna_i - C_c) \\ d_v \cdot (C_r - 1) \end{pmatrix} = \begin{pmatrix} u_1 \\ v_1 \end{pmatrix} \tag{5.14}$$

Ahora que se tiene el píxel respecto al plano de la cámara y considerando que f es la distancia focal, se obtiene la ecuación (5.15) que corresponde a la ecuación de la línea vista l_1 que sale del origen del sistema de coordenadas de la cámara izquierda y pasa por el punto p_1^{pc} en el plano de la cámara.

$$\begin{pmatrix} x_{l1} \\ y_{l1} \\ z_{l1} \end{pmatrix} = \alpha_{l1} \cdot \begin{pmatrix} u_1 \\ v_1 \\ f \end{pmatrix}, \quad 0 \leq \alpha_{l1} \leq +\infty \tag{5.15}$$

Para obtener la ecuación de la línea vista l_2 que pasa por el píxel p_2=$(columna_i, H)^T$ se repite el procedimiento seguido anteriormente para obtener la ecuación de la línea vista l_1, considerando c_i=$columna_i$ y r_i=H. Entonces se obtiene como resultado la transformación dada por la ecuación (5.16) y la recta de la línea vista l_2 dada por la ecuación (5.17).

$$\mathbf{p}_1^{pc} = \begin{pmatrix} \frac{N_{c,u}}{N_{f,u}} \cdot d_u \cdot (columna_i - C_c) \\ d_v \cdot (C_r - H - 1) \end{pmatrix} = \begin{pmatrix} u_2 \\ v_2 \end{pmatrix} \tag{5.16}$$

$$\begin{pmatrix} x_{l2} \\ y_{l2} \\ z_{l2} \end{pmatrix} = \alpha_{l2} \cdot \begin{pmatrix} u_2 \\ v_2 \\ f \end{pmatrix}, \quad 0 \leq \alpha_{l2} \leq +\infty \tag{5.17}$$

Supóngase que la ecuación de la línea de mejor ajuste asociada a la $columna_i$ elegida, está dada por la ecuación (5.18).

$$\begin{pmatrix} x_l \\ y_l \\ z_l \end{pmatrix} = \alpha \cdot \begin{pmatrix} v_x \\ v_y \\ v_z \end{pmatrix} + \begin{pmatrix} x_i \\ y_i \\ z_i \end{pmatrix}, \quad -\infty \leq \alpha \leq +\infty \tag{5.18}$$

Lo que sigue es encontrar el $\alpha 1$ de la línea de mejor ajuste dada por la ecuación (5.18) que garantiza la distancia mínima entre esta y la línea vista l_1 dada por la ecuación (5.15). Para ello se debe de calcular el producto cruz de los vectores directores de esas dos rectas, este vector esta dado por la ecuación (5.19) y se denota como $\mathbf{A}_1$.

$$\mathbf{A}_1 = \begin{pmatrix} v_1 \cdot v_z - f \cdot v_y \\ f \cdot v_x - u_1 \cdot v_z \\ u_1 \cdot v_y - v_1 \cdot v_x \end{pmatrix} = \begin{pmatrix} a_{x1} \\ a_{y1} \\ a_{z1} \end{pmatrix} \tag{5.19}$$

Luego se obtiene el vector $\mathbf{B}$ dado por la ecuación (5.20), este es un vector que va de un punto de la recta de mejor ajuste a un punto de la recta l_1 (el origen de coordenadas).

$$\mathbf{B} = \begin{pmatrix} 0 - x_i \\ 0 - y_i \\ 0 - z_i \end{pmatrix} = \begin{pmatrix} b_x \\ b_y \\ b_z \end{pmatrix} \tag{5.20}$$

Ahora se calcula el producto punto entre el vector $\mathbf{A}_1$ y el vector $\mathbf{B}$, este se denota como C_1 y está dado por la ecuación (5.21).

$$C_1 = b_x \cdot a_{x1} + b_y \cdot a_{y1} + b_z \cdot a_{z1} \tag{5.21}$$

Se calcula el modulo del vector $\mathbf{A}_1$, el cual esta dado por la ecuación (5.22).

$$|\mathbf{A}_1| = \sqrt{a_{x1}^2 + a_{y1}^2 + a_{z1}^2} \tag{5.22}$$

Luego se calcula la distancia mínima entre la recta de mejor ajuste y la línea vista l_1, esta distancia está dada por la ecuación (5.23) y corresponde a dividir el escalar C_1 entre el módulo del vector $\mathbf{A}_1$.

$$d_{min1} = \frac{C_1}{|\mathbf{A}_1|} \tag{5.23}$$

Ahora se obtiene el vector $\mathbf{N}_1$, este corresponde al vector de menor modulo que une a las dos rectas, tiene la misma dirección que $\mathbf{A}_1$ con un módulo igual a d_{min1}. Este vector está dado por la ecuación (5.24).

$$\mathbf{N}_1 = \frac{d_{min1}}{|\mathbf{A}_1|} \cdot \begin{pmatrix} a_{x1} \\ a_{y1} \\ a_{z1} \end{pmatrix} = \begin{pmatrix} n_{x1} \\ n_{y1} \\ n_{z1} \end{pmatrix} \tag{5.24}$$

Ahora se procede a determinar el vector que une dos puntos cualesquiera de las rectas y se obtiene al restar la ecuación de la línea de mejor ajuste con la ecuación de la línea vista l_1. Este vector se denota como $\mathbf{D}_1$ y está dado por la ecuación (5.25).

$$\mathbf{D}_1 = \begin{pmatrix} v_x \cdot \alpha - u_1 \cdot \alpha_{l1} + x_i \\ v_y \cdot \alpha - v_1 \cdot \alpha_{l1} + y_i \\ v_z \cdot \alpha - f \cdot \alpha_{l1} + z_i \end{pmatrix} \tag{5.25}$$

El punto de la recta de mejor ajuste que está más cerca de un punto de la línea de mejor ajuste, son tales que el vector $\mathbf{D}_1$ es paralelo al vector $\mathbf{N}_1$. Imponiendo esta condición, se obtiene un sistema de tres ecuaciones con dos incógnitas, en la expresión (5.26) se observan dos de estas ecuaciones que surgen de igualar a cero el componente x y el componente y del producto cruz. En la expresión (5.27) se muestran estas dos ecuaciones, pero de forma simplificada para facilitar los cálculos más adelante.

$$\mathbf{N}_1 x \mathbf{D}_1 = 0 \longrightarrow \begin{array}{l} (n_{y1} \cdot v_z - n_{z1} \cdot v_y) \cdot \alpha + (n_{y1} \cdot f - n_{z1} \cdot v_1) \cdot \alpha_{l1} + (n_{y1} \cdot z_i - n_{z1} \cdot y_i) = 0 \\ (n_{x1} \cdot v_z - n_{z1} \cdot v_x) \cdot \alpha + (n_{x1} \cdot f - n_{z1} \cdot u_1) \cdot \alpha_{l1} + (n_{x1} \cdot z_i - n_{z1} \cdot x_i) = 0 \end{array} \tag{5.26}$$

$$\mathbf{N}_1 x \mathbf{D}_1 = 0 \longrightarrow \begin{array}{l} nd_{x1-\alpha} \cdot \alpha + nd_{x1-\alpha_{l1}} \cdot \alpha_{l1} + nd_{x1} = 0 \\ nd_{y1-\alpha} \cdot \alpha + nd_{y1-\alpha_{l1}} \cdot \alpha_{l1} + nd_{y1} = 0 \end{array} \tag{5.27}$$

Resolviendo el sistema de dos ecuaciones para α se obtiene finalmente el valor de $\alpha 1$ que está dado por la expresión (5.28).

$$\alpha_1 = \frac{nd_{x1} \cdot nd_{y1-\alpha_{l1}} - nd_{x1-\alpha_{l1}} \cdot nd_{y1}}{nd_{x1-\alpha_{l1}} \cdot nd_{y1-\alpha} - nd_{x1-\alpha} \cdot nd_{y1-\alpha_{l1}}} \tag{5.28}$$

Para encontrar el valor α_2, se repite todo el procedimiento descrito anteriormente para encontrar el valor de $\alpha 1$ a partir de la línea vista l_1, con la única diferencia de que ahora para los cálculos se considera la ecuación de la línea vista l_2 que está dada por la ecuación (5.17) en lugar de la ecuación de la línea vista l_1. Entonces llegado a este punto se tiene el segmento dado por la expresión (5.29) y es a partir de este segmento que se calculan los vértices.

$$\begin{pmatrix} x_l \\ y_l \\ z_l \end{pmatrix} = \alpha \cdot \begin{pmatrix} v_x \\ v_y \\ v_z \end{pmatrix} + \begin{pmatrix} x_i \\ y_i \\ z_i \end{pmatrix}, \quad \alpha_1 \leq \alpha \leq \alpha_2 \tag{5.29}$$

Todo el procedimiento anterior se debe de realizar para cada una de las líneas de mejor ajuste que están asociadas a las columnas que se consideran para obtener la red de triángulos. Entonces al final, cada una de las columnas consideradas van a tener asociadas un α_1 y un α_2 que definen un segmento de la respectiva línea de mejor ajuste, como el que está dado por la ecuación (5.29).

5.4.2. Selección de los vértices

Ahora que se conoce para cada columna elegida el α_1 y el α_2 que delimitan un segmento de la línea de mejor ajuste, se puede proceder a seleccionar los vértices a partir de este segmento.

Para poder seleccionar los vértices lo que se hace es ir avanzando sobre el segmento desde α_1 hasta α_2 cada cierta distancia, es decir, se divide el segmento en varios puntos, donde la distancia entre puntos consecutivos se le conoce como el paso entre puntos (PEP), esto como se observa en la figura 5.10. El valor del paso entre puntos (PEP) depende totalmente del valor de α_2, el valor de α_1 y del número de vértices por columna (NVC), como se aprecia en la ecuación (5.30). El número de vértices por línea es el número de puntos en los que se divide el segmento y se puede definir como la cantidad vértices que se deben de obtener por cada columna elegida, de tal forma que el número total de vértices de la red de triángulos sea el número de columnas por el numero de vértices por columna.

$$PEP = \frac{\alpha_2 - \alpha_1}{NVC} \tag{5.30}$$

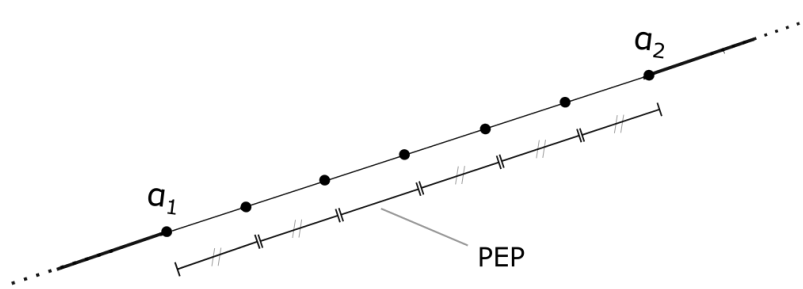


Figura 5.10: División del segmento que se encuentra entre α_1 y α_2 en varios puntos.

Cada uno de los puntos en los que se divide el segmento tienen un valor de α asociado que se denota como α_j y es a partir de este que se puede calcular la posición tridimensional del punto. Con la expresión de la ecuación (5.31) se puede obtener el valor de α_j que está asociado a un punto $\mathbf{P}_j$ que pertenece al segmento para un j dado que va desde cero hasta el número de vértices por columna menos uno.

$$\alpha_j = \alpha_1 + PEP \cdot j, \quad 0 \leq j \leq NVC - 1 \tag{5.31}$$

Para encontrar la posición tridimensional del punto $\mathbf{P}_j = (x_j, y_j, z_j)^T$ que pertenece al segmento, se evalúa la ecuación de la recta de mejor ajuste dada por la expresión (5.18) en el respectivo valor de α_j. Entonces la posición del punto número j en que se divide el segmento está dada por la ecuación (5.32).

$$\mathbf{P}_j = \begin{pmatrix} x_j \\ y_j \\ z_j \end{pmatrix} = \alpha_j \cdot \begin{pmatrix} v_x \\ v_y \\ v_z \end{pmatrix} + \begin{pmatrix} x_i \\ y_i \\ z_i \end{pmatrix}, \quad \alpha_1 \leq \alpha \leq \alpha_2 \tag{5.32}$$

Una vez que se conoce la posición tridimensional de cada uno de los puntos en los que se divide el segmento, se puede proceder a seleccionar los vértices asociados a ese segmento. Para ello lo que se hace es ir punto por punto, desde el punto dado por $j = 0$ hasta el punto dado por $j = NVC - 1$, y a cada uno de estos puntos se les asocia un vértice que se pueden dividir en dos tipos según la forma en cómo se seleccionan.

El primer tipo de vértice es aquel punto que forma parte de la respectiva porción de la nube de puntos asociada a la columna y que cumple con una cierta condición para poder ser clasificado como vértice. Esta condición consiste en que el ángulo θ que se forma entre la línea de mejor ajuste y una línea que contiene al punto $\mathbf{P}_j$ que se está considerando del segmento y al punto $\mathbf{P}_{vk}$ de la porción de la nube de puntos, sea mayor o igual que un cierto valor denotado como θ_{min}, esto como se observa en la figura 5.11.

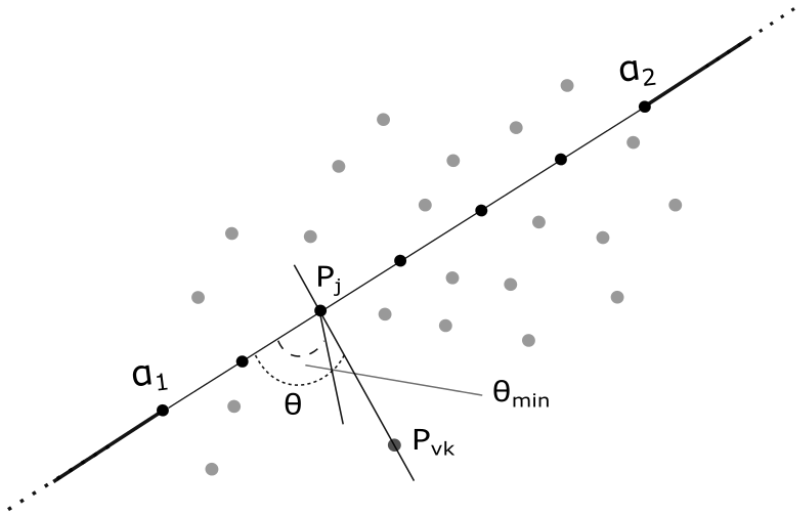


Figura 5.11: Vértice que pertenece a la nube de puntos y que cumple con la condición del ángulo.

Para encontrar el primer tipo de vértice $\mathbf{P}_{vk}$ asociado al punto P_j se debe de seguir el siguiente procedimiento para encontrar el ángulo entre dos rectas. Primero se debe de calcular una recta que pasa por el punto $\mathbf{P}_k = (x_k, y_k, z_k)^T$ de la respectiva porción dela nube de puntos y por el punto $\mathbf{P}_j$, esta recta está dada por la ecuación (5.33).

$$\begin{pmatrix} x_{kj} \\ y_{kj} \\ z_{kj} \end{pmatrix} = \alpha_{kj} \cdot \begin{pmatrix} x_k - x_j \\ y_k - y_j \\ z_k - z_j \end{pmatrix} + \begin{pmatrix} x_j \\ y_j \\ z_j \end{pmatrix} = \alpha_{kj} \cdot \begin{pmatrix} v_{xkj} \\ v_{ykj} \\ v_{zkj} \end{pmatrix} + \begin{pmatrix} x_j \\ y_j \\ z_j \end{pmatrix}, \quad -\infty \leq \alpha_{kj} \leq +\infty \tag{5.33}$$

Lo siguiente seria calcular el valor absoluto del producto punto de los vectores directores de ambas rectas. Este se obtiene con la ecuación (5.34) y se denota como A.

$$A = |v_x \cdot v_{xkj} + v_y \cdot v_{ykj} + v_z \cdot v_{zkj}| \tag{5.34}$$

Luego se calcula el módulo del vector director de la recta de mejor ajuste y el módulo del vector director de la recta que pasa por el punto $\mathbf{P}_j$ y el punto $\mathbf{P}_k$, estos se denotan como B y C respectivamente y están dados por las ecuaciones (5.35) y (5.36).

$$B = \sqrt{v_x^2 + v_y^2 + v_z^2} \tag{5.35}$$

$$C = \sqrt{v_{xkj}^2 + v_{ykj}^2 + v_{zkj}^2} \tag{5.36}$$

El coseno del ángulo θ_{kj} que se desea calcular está dado por la ecuación (5.37) y corresponde a dividir el valor absoluto del producto punto de los vectores directores de las rectas, entre la multiplicación del módulo de cada uno de estos. Luego se puede despejar el ángulo θ_{kj} al sacar

un arcocoseno a ambos lados de la ecuación (5.37) y así se obtiene el ángulo θ_{kj} dado por la expresión (5.38) que se pasó de radianes a grados.

$$cos(\theta_{kj}) = \frac{A}{B \cdot C} \tag{5.37}$$

$$\theta_{kj} = acos(\frac{A}{B \cdot C}) \cdot \frac{180}{\pi} \tag{5.38}$$

Si el ángulo θ_{kj} cumple con la condición dada por la expresión (5.39), el punto P_k de la porción de la nube de puntos pasa a ser un vértice de la red de triángulos que se denota como $P_v k$.

$$\theta_{kj} \geq \theta_{min} \longrightarrow P_k = P_{vk} \in [vertices] \tag{5.39}$$

Para cada uno de los puntos P_j en los que se divide el segmento se debe de recorrer todos los puntos P_k de la respectiva porción de la nube de puntos (donde k va desde 0 hasta el número de puntos tridimensionales de esa columna) y por cada uno de estos se debe de seguir el procedimiento anterior para encontrar el respectivo ángulo $\theta_k j$. Se deja de recorrer la porción dela nube de puntos cuando se encuentre un vértice, es decir, se pasa al siguiente P_j cuando se cumpla la condición dada por la expresión (5.39).

El segundo tipo de vértices son puntos que pertenecen al segmento y surgen cuando para un punto P_j no se encontró ningún punto de la porción de la nube de puntos que cumpliera con la condición del ángulo. Entonces en este caso lo que se hace es elegir como vértice ese punto P_j del segmento que no tiene asociado ningún punto de la nube de puntos que cumpla con la condición, como se resume en la expresión (5.40).

$$\theta_{kj} < \theta_{min} \longrightarrow P_j = P_{vk} \in [vertices] \tag{5.40}$$

Todo el procedimiento anterior se debe de realizar para cada uno de los segmentos que están asociadas a las columnas que se consideran para obtener la red de triángulos. Entonces al final, se va a tener una nube de vértices, es decir, un conjunto de puntos tridimensionales que pertenecen a la nube de puntos y/o a los segmentos y que ahora como un todo corresponden a los vértices que constituyen la red de triángulos. Cada columna tendrá asociada una porción de esa nube de vértices.

5.5. Asignar los vértices a cada uno de los triángulos que componen la red de triángulos

Finalmente, se puede proceder a obtener la red de triángulos a partir de los vértices obtenidos en la sección anterior, que corresponden a una nube de vértices cuyas posiciones tridimensionales se conocen y pertenecen a la nube de puntos y/o a los respectivos segmentos de cada una de las columnas elegidas.

Se parte del hecho de que se conoce la cantidad total de vértices (NV) que es igual al número de líneas verticales (NC) que se desea que tenga la red de triángulos por el número de vértices

por columna (NVC) que es igual para cada columna elegida. Dicho de otra forma, se sabe la cantidad de columnas que va a tener la red de triángulos y el número de vértices por columna que es igual para todas.

Antes de asignar los vértices es necesario hay que tener claro dos conceptos y sus respectivas ecuaciones. El primero es el número de cuadros entre dos columnas consecutivas de la red de triángulos, este se denota como NC_{2c} y se puede definir como la cantidad de rectángulos consecutivos que se pueden obtener con todos los vértices de dos columnas consecutivas, como se aprecia en la figura 5.12. Esta dado por la expresión (5.41) y depende del número de vértices por columna (NVC). El segundo es el número de triángulos entre dos columnas consecutivas de la red de triángulos (NT_{2c}) que parte del simple hecho de que un cuadrado se puede dividir en dos triángulos. Esta dado por la ecuación (5.42).

$$NC_{2c} = NVC - 1 \tag{5.41}$$

$$NT_{2c} = 2 \cdot NC_{2c} \tag{5.42}$$

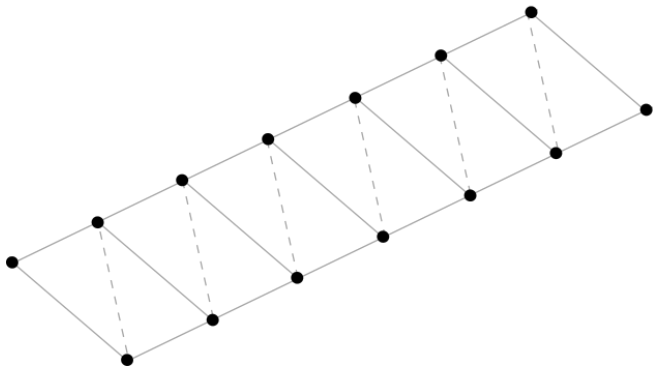


Figura 5.12: Rectángulos y triángulos entre dos columnas de vértices consecutivas.

Ahora se procede a asignar los vértices a cada uno de los triángulos que componen la red de triángulos. Para ello se supone que todos los vértices están numerados del 1 al NVC, en la figura 5.13 se aprecia un ejemplo de la numeración de los vértices para la primera y la segunda columna para $NVC = 7$. También se supone que todos los triángulos de la red de triángulos están compuestos por tres vértices (A, B y C) asignados como se observa en la figura 5.14.

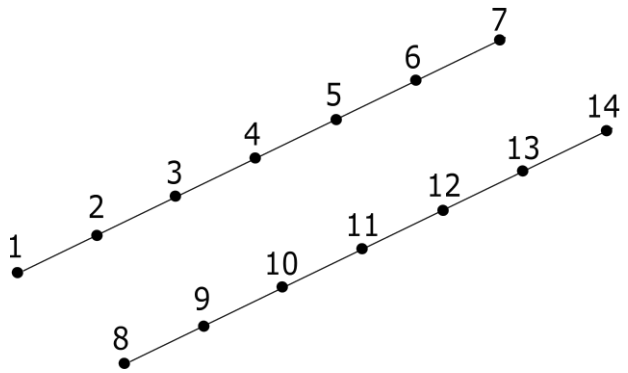


Figura 5.13: Numeración de los vértices entre dos columnas consecutivas para $NVC = 7$.

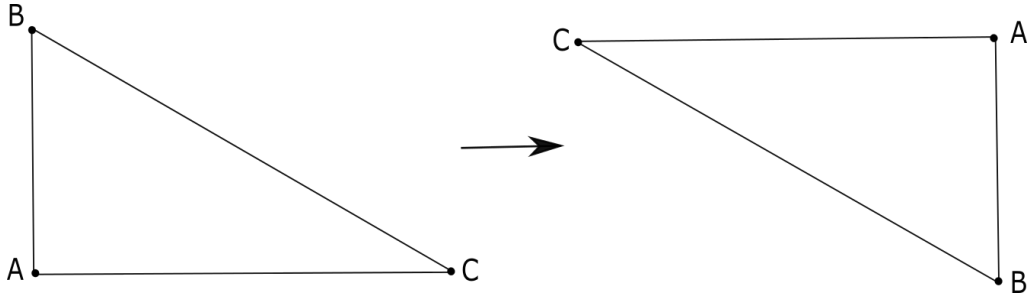


Figura 5.14: Triángulo compuesto por los vértices A, B y C.

Si se define la variable j como el número de triángulo por columna, que va desde 1 hasta NT_{2c}, y la variable i como el par de columnas consecutivas, que va desde 1 hasta $NC - 1$, entonces el número de triángulo de la red de triángulos (NT) está dado por la expresión (5.43).

$$NT = j + (i - 1) \cdot NT_{2c}, \quad 1 \leq i < NC, \quad 1 \leq j \leq NT_{2c} \tag{5.43}$$

Ahora se quiere asignar los vértices que componen al triángulo número NT de la red de triángulos para un j y un i dado. Esto se logra con la expresión (5.44) y con la expresión (5.45).

$$j \leq NC_{2c} \longrightarrow \begin{array}{c} A = j + (i - 1) \cdot NVC \\ B = A + 1 \\ C = A + NVC \end{array} \tag{5.44}$$

$$j > NC_{2c} \longrightarrow \begin{array}{c} B = j + 1 + (i - 1) \cdot NVC \\ A = B + 1 \\ C = A - NVC \end{array} \tag{5.45}$$

Finalmente, para asignar los vértices a todos los triángulos de la red de triángulos lo que se tiene que hacer es que para cada par de columnas consecutivas que tienen asociadas una cantidad fija de vértices (NVC) los cuales se numeran como en la figura 5.13, se debe de asignar los respectivos vértices que constituyen a cada uno de los triángulos que puede haber entre esas dos columnas. Es decir, para cada valor de i, que va desde 1 hasta $NC - 1$, se debe de recorrer todos los valores de j que va desde 1 hasta NT_{2c} y evaluar las expresiones (5.44) o (5.45). Y es de esta forma como se obtendría la red de triángulos.

Cabe destacar que para la asignación de los vértices se tienen dos casos, el primero caso es cuando el número de triangulo entre dos columnas consecutivas (j) es menor o igual que la mitad del número de triángulos entre dos columnas (NC_{2c}) y el segundo caso es cuando el número de triangulo entre dos columnas consecutivas es mayor que la mitad del número de triángulos entre dos columnas. Estos dos casos se deben a que primero se asignan los vértices a los triángulos que tienen un lado que coincide con la columna izquierda y luego a los triángulos que tienen un lado que coincide con la columna derecha, como se aprecia en la figura 5.15.

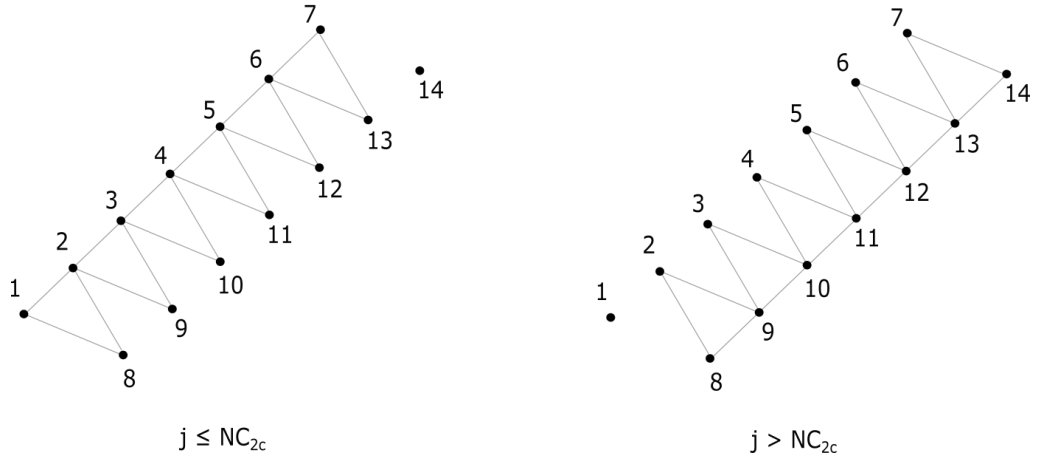


Figura 5.15: Casos al asignar los vértices a los triángulos entre dos columnas consecutivas.

5.6. Sistema de coordenadas local de la red de triángulos

Para poder conocer la posición y orientación de la red de triángulos respecto al sistema de coordenadas de la cámara izquierda definido previamente como $(X_l, Y_l, Z_l)^T$, se debe de establecer un sistema de coordenadas local de la red de triángulos definido por los ejes $(U, V, W)^T$, como se aprecia en la figura 5.16.

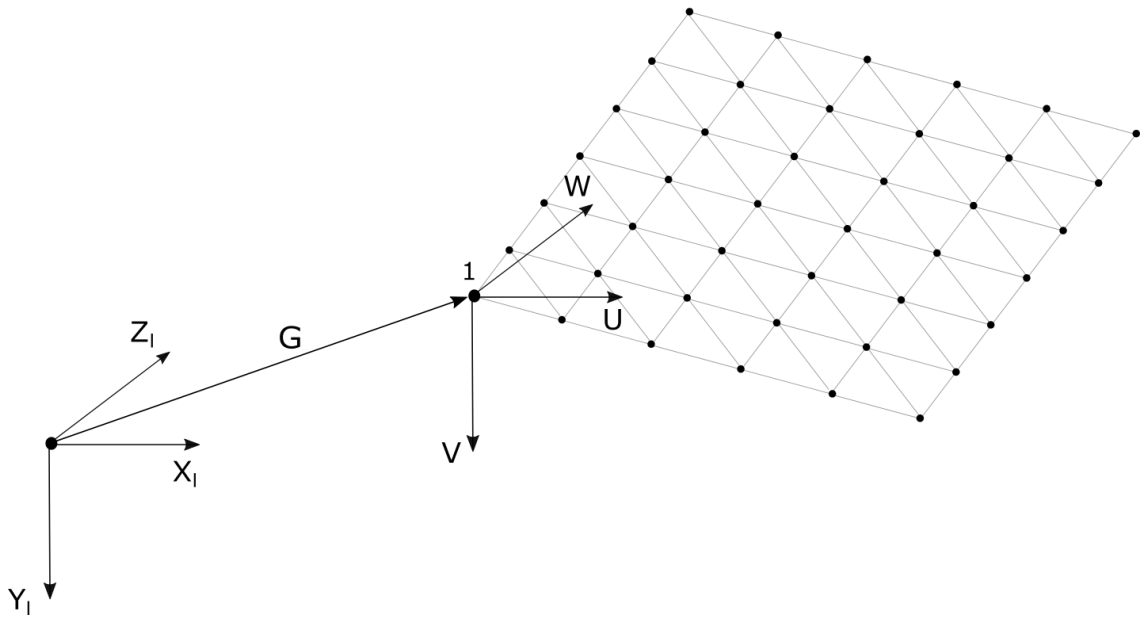


Figura 5.16: Sistema de coordenadas local de la red de triángulos y el sistema de coordenadas de referencia.

5.6.1. Posición de la red de triángulos

El origen del sistema de coordenadas local de la red de triángulos, que define la posición respecto al sistema de coordenadas de la cámara izquierda (referencia), se ubica en el primer vértice de la red de triángulos, es decir, en la esquina inferior izquierda de esta, como se observa en la figura 5.16.

Si la posición tridimensional del primer vértice está dada por la expresión (5.46), entonces la posición del origen del sistema de coordenadas local del objeto (red de triángulos) con respecto al sistema de coordenadas de referencia está dada por la expresión (5.47).

$$\mathbf{V}_1 = \begin{pmatrix} x_{v1} \\ y_{v1} \\ z_{v1} \end{pmatrix} \tag{5.46}$$

$$\mathbf{G}_o^l = \mathbf{V}_1 = \begin{pmatrix} x_{v1} \\ y_{v1} \\ z_{v1} \end{pmatrix} = \begin{pmatrix} G_x \\ G_y \\ G_z \end{pmatrix} \tag{5.47}$$

5.6.2. Orientación de la red de triángulos

El sistema de coordenadas local de la red de triángulos $(U, V, W)^T$ se encuentra totalmente alineado con el sistema de coordenadas de la cámara izquierda $(X_l, Y_l, Z_l)^T$. Por lo tanto, la matriz de

rotación que describe la orientación relativa del sistema de coordenadas local del objeto con respecto al sistema de coordenadas de la cámara izquierda, corresponde a la matriz identidad y está dada por la expresión (5.48). Además, los tres ángulos de orientación son 0°, por lo que la orientación del sistema de coordenadas de la red de triángulos con respecto al sistema de coordenadas de la cámara izquierda está dada por la expresión (5.49).

$$\mathbf{R}_o^l = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{5.48}$$

$$\mathbf{O}_o^l = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} \Re_x \\ \Re_y \\ \Re_z \end{pmatrix} \tag{5.49}$$

5.6.3. Vértices respecto al sistema de coordenadas local

Los vértices de la red de triángulos que se obtuvo anteriormente a partir de la nube de puntos, están definidos con respecto al sistema de coordenadas de la cámara izquierda.

Ahora que se conoce la posición y orientación de la red de triángulos respecto al sistema de coordenadas de la cámara izquierda, se puede utilizar la ecuación (4.1) para definir todos los vértices respecto al sistema de coordenadas local de la red. Si el vector que define la posición tridimensional de los vértices respecto a la cámara izquierda se denota como V_l, entonces el vector que de fine la posición tridimensional de los vértices respecto al sistema de coordenadas local de la red se obtiene con la ecuación (5.50).

$$\mathbf{V}_o = \mathbf{V}_l - \mathbf{G}_o^l \tag{5.50}$$

Finalmente, si se aplica la ecuación anterior para todos los vértices de la red de triángulos, se tendrían todos estos con respecto al sistema de coordenadas local.

5.7. Implementación del algoritmo para generar una red de triángulos a partir de una nube de puntos.

Todo el procedimiento matemático (algoritmo) descrito anteriormente, para describir y estimar la forma irregular de la superficie en exploración mediante una red rígida de triángulos constituida por más de cuatro vértices y dos triángulo y para determinar la posición y orientación de esta con respecto al sistema de coordenadas de la cámara izquierda, se implementó en el lenguaje de programación C, haciendo uso de la plataforma de programación ROS[2].

El resultado final o salida del algoritmo implementado, está constituida por dos archivos. El primero es un archivo en formato .stp (archivo CAD 3D STEP que se califica como un archivo

[2]Un mayor desarrollo sobre el funcionamiento de ROS se puede encontrar en el apéndice A.

de imágenes 3D) el cual describe la red de triángulos que estima la forma tridimensional de la superficie, este archivo contiene lo siguiente:

- La cantidad total de vértices.
- Una lista con todas las posiciones tridimensionales de cada uno de los vértices con respecto al sistema de coordenadas local de la red.
- La cantidad total de triángulos.
- Una lista con los índices de los vértices que conforman cada uno de los triángulos de la red.

A manera de ejemplo, en la figura 5.17 se muestra el archivo .stp que se obtendría para la red de triángulos más simple posible (modela un rectángulo con 4 vértices y 2 triángulos) que se muestra en la figura 5.1 de la sección 5.1.

```
#define 5 superficie
4
x1 y1 z1
x2 y2 z2
x3 y3 z3
x4 y4 z4
2
1 2 3
4 3 2
```

Figura 5.17: Archivo .stp que describe la red de triángulos que estima una forma rectangular.

El segundo es un archivo en formato .txt, en este se muestra la posición del sistema de coordenadas local del objeto (red de triángulos) respecto al sistema e coordenadas de la cámara izquierda, la matriz de rotación que describe la orientación relativa del sistema de coordenadas local del objeto con respecto al sistema de coordenadas de la cámara izquierda y los tres ángulos de orientación extraídos de dicha matriz. En la figura 5.18 se muestra un ejemplo de la forma que tendría este archivo.

```
1  Posicion de la red (centimetros):
2  Gx = x1
3  Gy = y1
4  Gz = z1
5
6  Orientacion de la red
7    Matriz de rotacion:
8  1 0 0
9  0 1 0
10 0 0 1
11   Angulos de orientacion (grados):
12 Rx = 0.0
13 Ry = 0.0
14 Rz = 0.0
```

Figura 5.18: Archivo .txt que describe la posición y orientación de la red de triángulos respecto al sistema de coordenadas de la cámara izquierda.

El algoritmo implementado visto desde ROS es un nodo llamado $triangle_net_node$ y su funcionamiento depende totalmente de los tres nodos de ROS de código abierto descritos en la sección 4.4 y más específicamente en el apéndice A. Además, para para poder graficar la red de triángulos se utiliza una modificación el nodo llamado $Nodo_Remoto_Graficador$ creado por Greivin Oviedo [3], el cual utiliza OpenGL para graficar una red de triángulos descrita por un archivo .stp como el que se definió anteriormente.

En la figura 5.19, se muestra la comunicación entre todos los nodos necesarios para poder obtener el archivo .stp que describe la red de triángulos que estima la forma tridimensional de la superficie, el archivo .txt que describe la posición y orientación de esta y la respectiva gráfica. El nodo llamado $camera1394stereo$ toma la imagen izquierda y derecha capturadas por la cámara estereoscópica y las publica como un mensaje a través del tópico $/image_raw$. Luego, se ejecuta el nodo $camera_calibration$ que se suscribe a $/image_raw$ y calcula los parámetros de calibración. Luego, se ejecuta el nodo $stereo_image_proc$ que toma la información de $/image_raw$ y utiliza los parámetros de calibración para luego publicar en $/image_rect$ las imágenes calibradas y además, publica la nube de puntos en el tópico llamado $/points2$. Después, se ejecuta el nodo $triangle_net_node$ que está suscrito al tópico $/points2$ (recibe la nube de puntos) y realiza todos los cálculos necesarios para obtener la red de triángulos y por lo tanto los dos archivos de salida. Justo antes de que el nodo $triangle_net_node$ termine de ejecutarse, este escribe un 1 en un archivo de texto llamado $enable$, indicando que finalizo los cálculos. Luego, el nodo $Nodo_Remoto_Graficador$ está constantemente abriendo y leyendo lo que hay en el archivo $enable$, cuando este detecta que hay un 1 entonces gráfica la respectiva red de triángulos a partir del archivo .stp e inmediatamente después este vuelve a poner un 0 en $enable$.

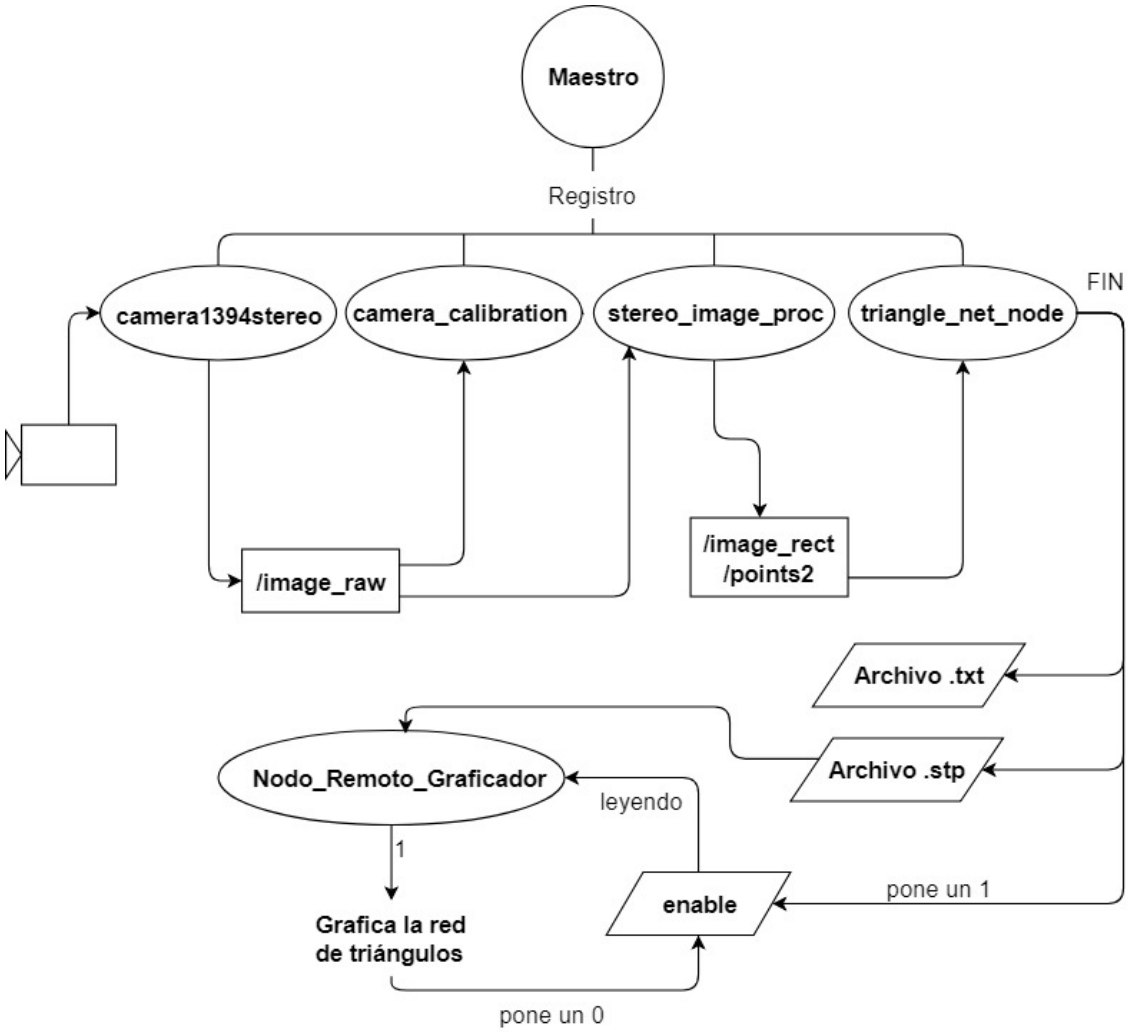


Figura 5.19: Comunicación entre los nodos necesarios para obtener y graficar una red de triángulos a partir de una nube de puntos.

Capítulo 6

Resultados experimentales

6.1. Cámara estereoscópica utilizada

Se utilizó la cámara estereoscópica que se observa en la figura 6.1, la cual es una cámara Bumblebee 2 de la marca Point Grey modelo BB2-03S2M-60. Está compuesta por dos lentes, y sus sistemas de coordenadas se encuentran alineados.

Sus características y especificaciones más sobresalientes son las siguientes:

- Tipo de sensor de imagen: Sony ICX424 CCD.
- Distancia entre sensores: 7.4µ m.
- Máxima resolución: 648x488.
- Cuadros por segundo: 48.
- Línea base: 12cm.
- Distancia focal: 6mm.
- Interfaz: IEEE-1394a de 6 pines para control de cámara y transmisión de datos de vídeo.

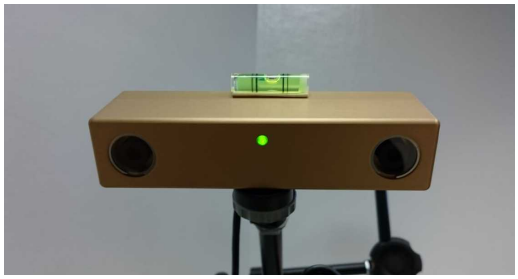


Figura 6.1: Cámara Bumblebee 2 de la marca Point Grey.

6.2. Entorno controlado de pruebas

Se requiere que todas las pruebas se realicen en un entorno controlado, es decir, en un espacio definido para tal fin donde se puedan controlar y modificar una serie de factores que influyen directamente en los resultados de las pruebas. En este caso, para poder comparar una prueba con otra se requiere que los factores se mantengan fijos, es decir, que se fijen en un cierto valor establecido.

El entorno controlado o espacio utilizado para realizar las pruebas se aprecia en la figura 6.2, este se encuentra dentro del Laboratorio de Investigación en Procesamiento de Imágenes y Visión por Computador (IPCV-LAB). Consiste en una porción del suelo del laboratorio, donde la cámara estereoscópica Bumblebee 2 está puesta de tal manera que apunta directamente a esta porción. Cabe destacar, que la cámara esta puesta sobre un soporte de brazo que se encuentra fijo en una mesa del laboratorio, este soporte permite seis grados de libertad.

Figura 6.2: Espacio para realizar las pruebas experimentales.

En este entorno existen cuatro factores que se pueden controlar y modificar: el tipo de suelo, la inclinación horizontal de la cámara, la inclinación vertical de la cámara y la altura de la cámara respecto del suelo. En cuanto al tipo de suelo, se utilizó un muro cerámico con acabado de piedra, como se aprecia en la figura 6.2, para poder tener un buen contraste entre el suelo y los objetos que se vayan a poner sobre este para cada prueba a realizar. La inclinación de la cámara con respecto a la horizontal se fijó en aproximadamente 40°, como se observa en la figura 6.3, para ello se utilizó un inclinómetro (herramienta utilizada para medir la inclinación con respecto a la horizontal). Mientras que la inclinación de la cámara con respecto a la vertical se fijó en aproximadamente 0°, como se observa en la figura 6.4, para ello se utilizó un nivel de burbuja (instrumento utilizado para verificar si hay algún grado de inclinación respecto a la vertical). Por último, la altura de la cámara respecto del suelo se fijó en 90 centímetros, como se puede ver en la figura 6.5, donde se utilizó una cinta métrica para la medición.

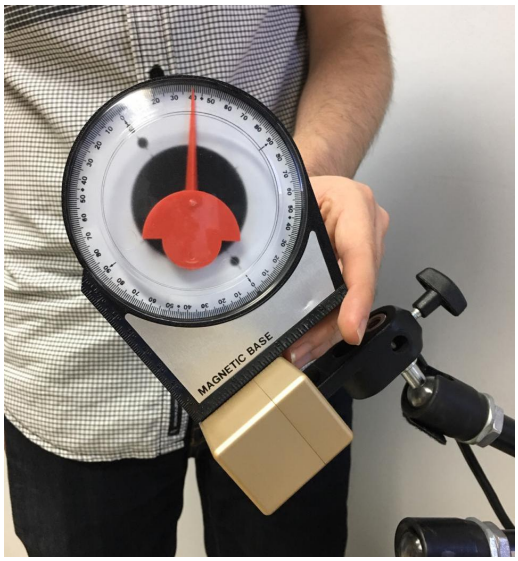


Figura 6.3: Inclinación horizontal de la cámara, utilizada para realizar las pruebas experimentales.

Figura 6.4: Inclinación vertical de la cámara, utilizada para realizar las pruebas experimentales.

Figura 6.5: Altura de la cámara respecto del suelo, utilizada para realizar las pruebas experimentales.

6.3. Planteamiento de las pruebas

Para poder comprobar la funcionalidad del algoritmo que genera una red de triángulos que estima la forma tridimensional de la porción de la superficie que está siendo capturada por la cámara, se deben de llevar a cabo varias pruebas que sometan al algoritmo a una serie de casos en cuanto a la forma irregular de la superficie, es decir, que cada prueba consista en variar la forma irregular de la superficie que está siendo capturada por la cámara, para poder determinar mediante observación que tan preciso es el algoritmo. El resultado de cada una de estas pruebas será la gráfica de la respectiva red de triángulos vista desde la cámara y la imagen capturada por la cámara izquierda, esto para poder determinar visualmente por comparación que tanto se asemeja la red de triángulos a la superficie descrita por la imagen capturada.

Las pruebas llevadas a cabo consisten en poner distintos objetos de diferentes tamaños, específicamente piedras, sobre la superficie del espacio destinado para realizar dichas pruebas experimentales (figura 6.2). Específicamente, se quiere determinar si el algoritmo funciona correctamente cuando en la superficie se tienen objetos pequeños, medianos y grandes, y combinaciones de estos. También se tienen otras pruebas en las que se quiere determinar qué pasa cuando dos objetos están muy juntos, cuando se tiene un objeto muy alto o cuando se tiene un objeto con una forma muy irregular. Entonces se plantean un total de 12 pruebas que se clasifican de acuerdo al objeto o los objetos que se pondrán sobre la superficie del espacio de pruebas:

- Prueba 1: una piedra pequeña.
- Prueba 2: una piedra mediana.

- Prueba 3: una piedra grande.
- Prueba 4: una piedra pequeña, una mediana y una grande.
- Prueba 5: varias piedras pequeñas.
- Prueba 6: varias piedras medianas.
- Prueba 7: varias piedras.
- Prueba 8: varias piedras pequeñas, medianas y grandes.
- Prueba 9: una piedra al lado de otra piedra.
- Prueba 10: una piedra detrás de otra piedra.
- Prueba 11: un objeto con una forma muy irregular, específicamente un dinosaurio de plástico.
- Prueba 12: un objeto alto, específicamente un cilindro de piedra puesto sobre su base.

Cabe destacar que los resultados de las pruebas mencionadas anteriormente no solamente dependen del entorno de control, como se mencionó en la sección anterior, también dependen de los valores que tengan los parámetros de control del algoritmo. Estos parámetros son variables cuyo valor se elige de forma externa al algoritmo mediante un archivo de texto y repercuten directamente en la salida del algoritmo, en la forma de la red de triángulos. Un ejemplo de estos parámetros es el número de columnas que se desea que tenga la red de triángulos final. Entonces, para este experimento que está conformado por las 12 pruebas anteriores, se utilizaron los siguientes valores de los parámetros de control:

- Numero de columnas de la red de triángulos: 55.
- Numero de vértices por columna de la red de triangulas: 44.
- Porcentaje mínimo de posiciones tridimensionales por columna: 55.
- Numero de iteraciones del algoritmo de RANSAC: 40.
- Distancia máxima punto a línea (en centímetros) para el algoritmo de RANSAC: 0,08.
- Porcentaje mínimo de valores típicos (*inliers*) para el algoritmo de RANSAC: 85

6.4. Resultados de las pruebas y análisis

Los resultados de cada una de las 12 pruebas que se plantearon en la sección anterior, es decir, la gráfica de la respectiva red de triángulos vista desde la cámara y la imagen capturada por la cámara izquierda para cada prueba, se observan desde la figura 6.6 hasta la figura 6.16.

De la figura 6.6 a la figura 6.8 se muestran los resultados obtenidos para las primeras tres pruebas experimentales, que consisten en poner una única piedra cuyo tamaño se va variando. En cuanto a la primera prueba que consiste en poner una piedra pequeña, al comparar la figura 6.6b con la figura 6.6a se puede determinar por observación y comparación que la red de triángulos generada por el algoritmo es capaz de estimar la forma irregular de la superficie capturada por la cámara izquierda; esto debido a que en la gráfica de la red de triángulos se logra apreciar que hay un objeto relativamente pequeño que sobresale de la superficie plana y que corresponde a una aproximación de la forma y tamaño en general que se observa que tiene la piedra en la figura 6.6a. Para la segunda prueba que consiste en poner una piedra mediana, al comparar la figura 6.7b con la figura 6.7a se llega exactamente a las mismas observaciones de la primera prueba, ya que en la red de triángulos se puede apreciar que hay un objeto de mayor tamaño que el de la prueba anterior, que está sobre la superficie plana y que corresponde a una aproximación de la lo que se observa en la figura 6.7a. Para la tercera prueba que consiste en poner una piedra grande al comparar la figura 6.8b con la figura 6.8a, se llega exactamente a las mismas observaciones de la primera prueba y de la segunda prueba, que se genera una red de triángulos que trata de estimar y aproximar la forma irregular de la superficie.

En la figura 6.9 se muestran los resultados obtenidos para la cuarta prueba que consiste en poner todas las piedras de las pruebas anteriores sobre la superficie. Al comparar los resultados, la figura 6.9b con la figura 6.9a, se determina que la red de triángulos generada por el algoritmo trata de estimar la forma y tamaño de los objetos que se encuentran sobre la superficie plana, pero en comparación con las pruebas anteriores se puede apreciar que faltan columnas. Esto sucede porque hay columnas de la imagen cuya porción de la nube de puntos a la que están asociadas es prácticamente nula o muy pequeña, es decir, para la mayoría de los píxeles de esas columnas se desconoce su respectiva posición tridimensional, y entonces el algoritmo desecha este tipo de columnas dando como resultado una red de triángulos como la de la figura 6.9b. Lo anterior se debe a que la cantidad de puntos tridimensionales obtenidos (nube de puntos) varía según la forma que tenga la superficie capturada por la cámara, es decir, dependiendo de la superficie se encuentran más o menos puntos característicos entre un caso y otro. También la iluminación es un factor que afecta la cantidad de puntos obtenidos.

De la figura 6.10 a la figura 6.12 se muestran los resultados obtenidos para la quinta, sexta y séptima prueba, que consisten en poner varias piedras de un mismo tamaño y este se va variando. En cuanto a la quinta prueba que consiste en poner varias piedras pequeñas, al realizar la comparación entre la figura 6.10b y la figura 6.10a se determina que la red de triángulos generada por el algoritmo da una aproximación de la forma irregular de la superficie capturada por la cámara izquierda, aunque se puede apreciar que hay columnas faltantes como en el caso de la cuarta prueba, pero en una menor proporción. Para la sexta prueba (varias piedras medianas) y la séptima prueba (varias piedras grandes), al comparar la 6.11b con la figura 6.11a y la figura

6.12b con la figura 6.12a, respectivamente, se llega exactamente a las mismas observaciones de la quinta prueba, con la diferencia de que el problema las columnas faltantes se presenta en una mayor proporción.

En la figura 6.13 se muestran los resultados obtenidas para la octava prueba que consiste en poner la mayoría de las piedras utilizadas en las tres pruebas anteriores. Al comparar los resultados, la figura 6.13b con la figura 6.13a, se puede observar que la red de triángulos generada por el algoritmo da una estimación general de la forma irregular de la superficie. Además, se puede apreciar hay muchas columnas faltantes, más que en cualquier prueba anterior, y esto causa que la red de triángulos no sea capaz de aproximar la forma y tamaño de los objetos pequeños que sobresalen de la superficie plana, como en el caso de la piedra pequeña que se encuentra abajo a la derecha.

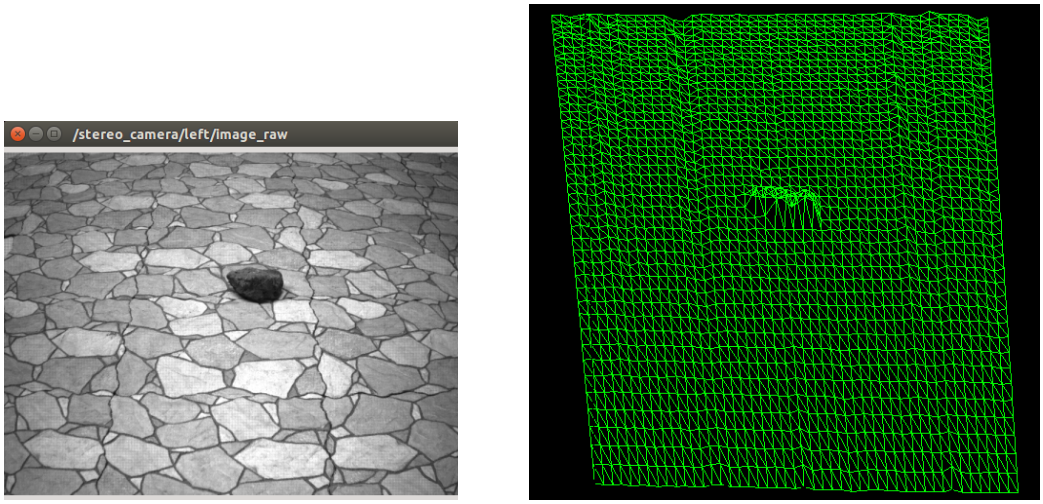
En la figura 6.14 y en la figura 6.15 se muestran los resultados de la novena prueba y de la décima prueba, estas consisten en poner dos piedras pegadas, pero de manera diferente. Para la novena prueba (piedra al lado de otra), al realizar la comparación entre la figura 6.14b y la figura 6.14a se determina que la red de triángulos cumple con las expectativas, ya que en esta se puede observar fácilmente que hay un objeto pegado a otro objeto y además trata de estima la forma que estos tienen. En cuanto a la décima prueba (piedra detrás de otra), al comparar la figura 6.15b con la figura 6.15a se llega a lo mismo que con la novena prueba, se puede observar fácilmente que hay un objeto detrás de otro.

En la figura 6.16 se muestran los resultados de la undécima prueba, en la cual se utilizó un dinosaurio de plástico para ver qué sucede con un objeto con una forma muy irregular. Al comparar el resultado de la figura 6.16b con el de la figura 6.16a, se puede apreciar que la red de triángulos obtenida da una muy mala aproximación de la forma general que se observa que tiene el objeto, solamente aproxima la silueta superior de este, la que va desde la cola hasta la cabeza. Además, se aprecia que hay líneas que van desde el suelo hasta la silueta superior, pasando por alto la parte media e inferior del objeto. Lo anterior se debe a que cuando se tienen objetos tan irregulares y con formas muy complicadas, la búsqueda de los puntos característicos se hace demasiado difícil y por lo tanto se obtienen muy pocas posiciones tridimensionales en el objeto, en este caso solo se obtuvieron puntos en la parte de arriba.

Finalmente, en la figura 6.17 se tienen los resultados de la última prueba, la duodécima prueba, en la cual se utilizó un cilindro de piedra puesto sobre su base, para observar que sucede con un objeto alto. Si se compara el resultado de la figura 6.17b con el de la figura 6.17a, se puede observar que la red de triángulos obtenida por el algoritmo da una mala aproximación de la forma en general que se observa que tiene el objeto. Además, en las figuras 6.17c y 6.17d se puede apreciar que se presenta el mismo problema de la prueba anterior, líneas que van desde la superficie hasta la parte superior, ignorando el resto del objeto.

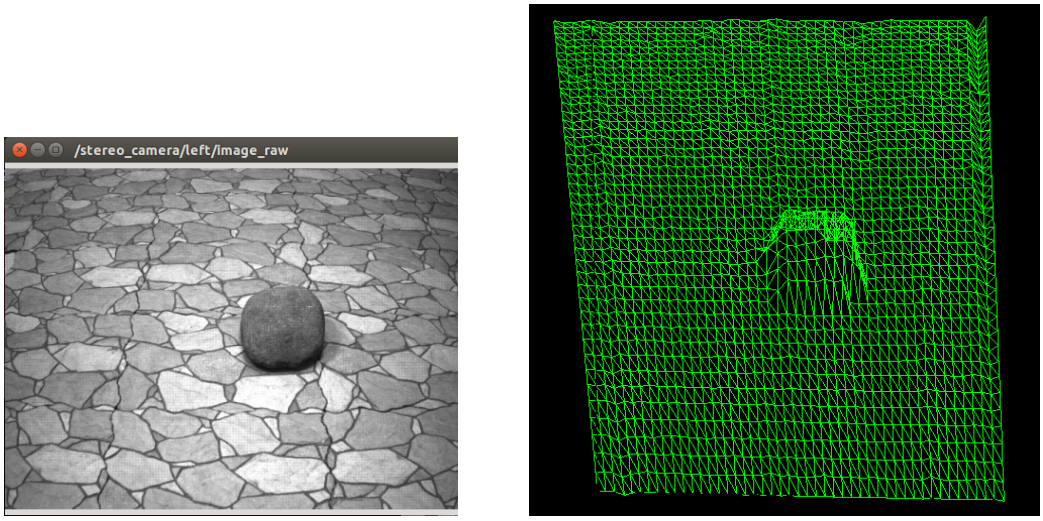
Cabe destacar que, en todas las pruebas la red de triángulos genera por el algoritmo es solamente una aproximación de la forma y tamaño en general que se observa que tienen los objetos en la imagen de la superficie real capturada por la cámara izquierda. El hecho de que la red de triángulos siempre sea una aproximación se debe principalmente a que es muy difícil encontrar las condiciones ideales que permitan obtener la mayor cantidad de puntos característicos sobre el objeto y que por lo tanto se genere una red de triángulos que de una excelente aproximación.

Siempre van a haber puntos faltantes sobre el objeto y eso causa que el algoritmo que genera la red de triángulos tenga que aproximar esos puntos faltantes.



(a) Imagen capturada por la cámara izquierda. (b) Gráfica de la red de triángulos.

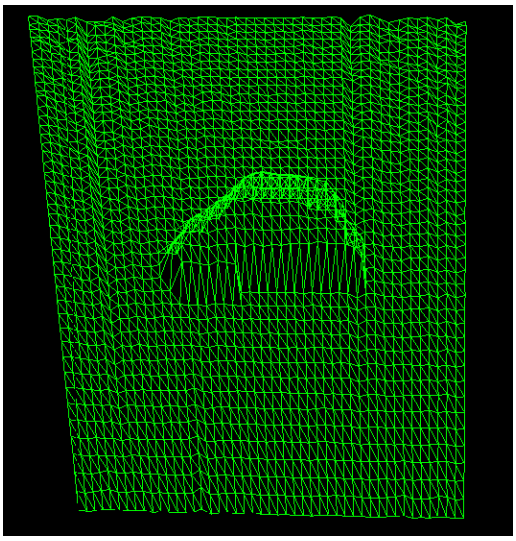
Figura 6.6: Prueba 1. Piedra pequeña sobre la superficie capturada por la cámara.



(a) Imagen capturada por la cámara izquierda. (b) Gráfica de la red de triángulos.

Figura 6.7: Prueba 2. Piedra mediana sobre la superficie capturada por la cámara.

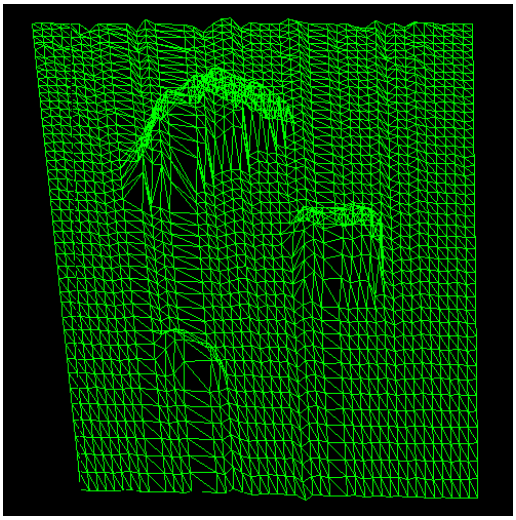
(a) Imagen capturada por la cámara izquierda.



(b) Gráfica de la red de triángulos.

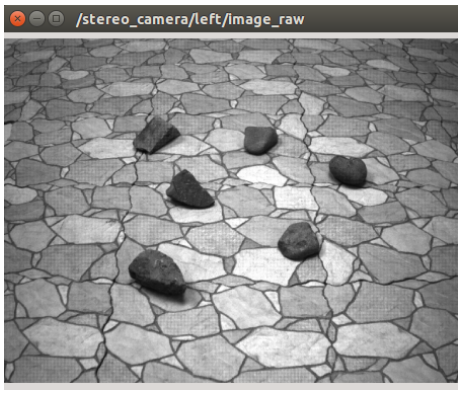
Figura 6.8: Prueba 3. Piedra grande sobre la superficie capturada por la cámara.

(a) Imagen capturada por la cámara izquierda.

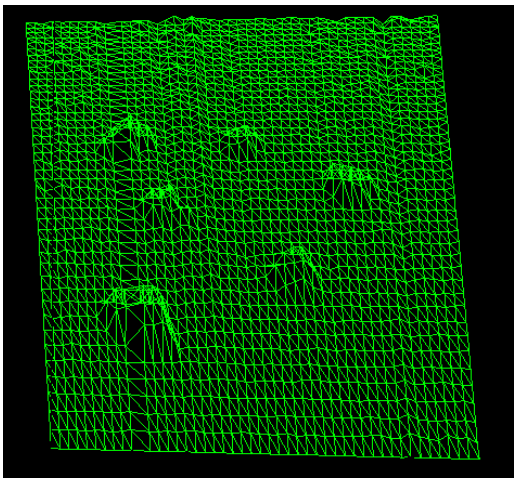


(b) Gráfica de la red de triángulos.

Figura 6.9: Prueba 4. Una piedra grande, una pequeña y una mediana sobre la superficie capturada por la cámara.



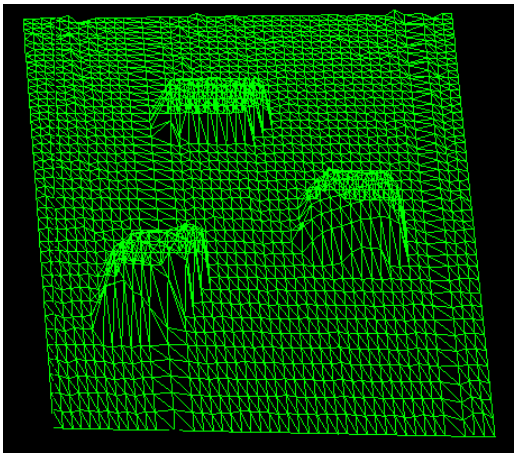
(a) Imagen capturada por la cámara izquierda.



(b) Gráfica de la red de triángulos.

Figura 6.10: Prueba 5. Varias piedras pequeñas sobre la superficie capturada por la cámara.

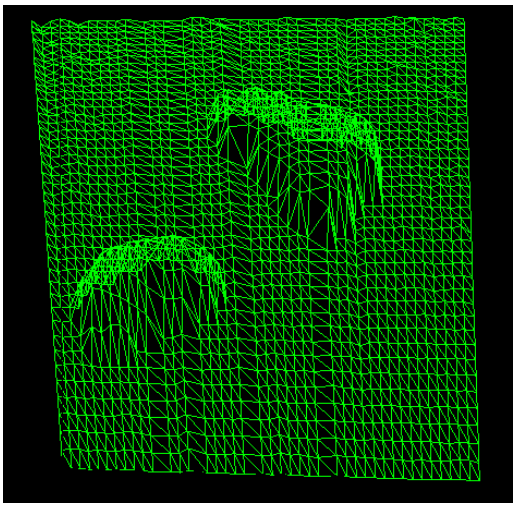
(a) Imagen capturada por la cámara izquierda.



(b) Gráfica de la red de triángulos.

Figura 6.11: Prueba 6. Varias piedras medianas sobre la superficie capturada por la cámara.

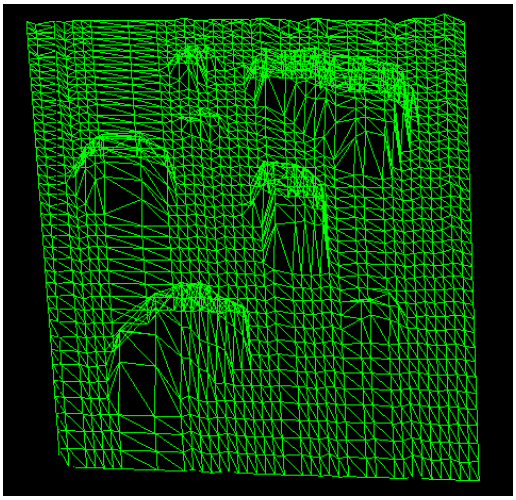
(a) Imagen capturada por la cámara izquierda.



(b) Gráfica de la red de triángulos.

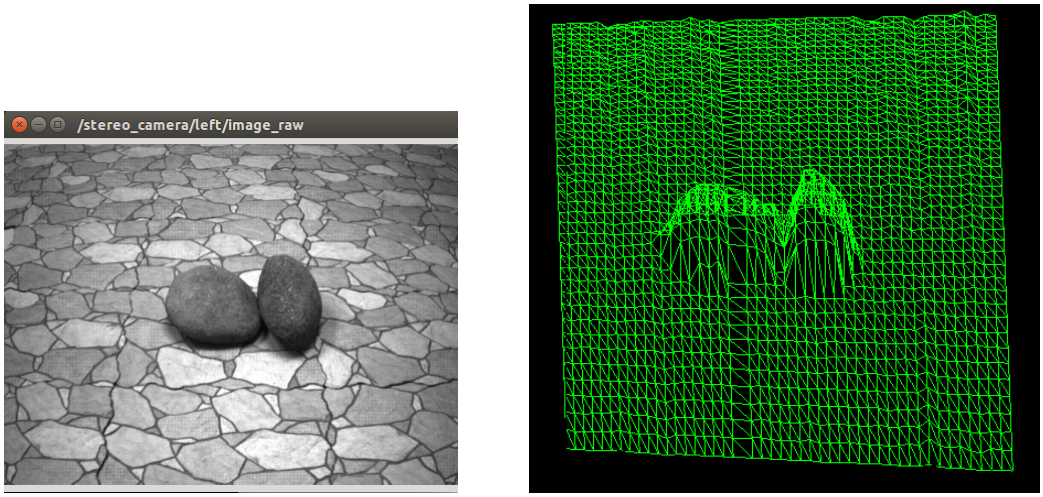
Figura 6.12: Prueba 7. Varias piedras grandes sobre la superficie capturada por la cámara.

(a) Imagen capturada por la cámara izquierda.



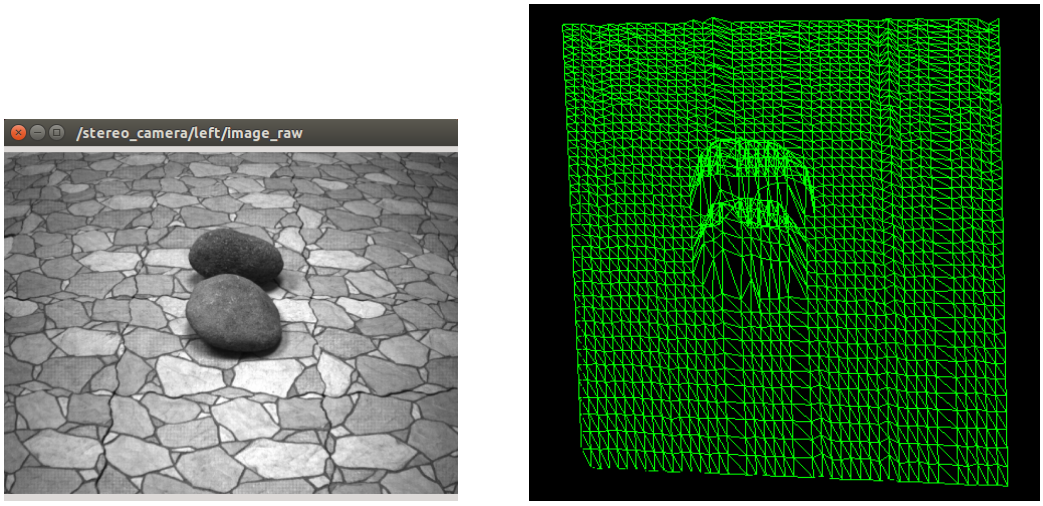
(b) Gráfica de la red de triángulos.

Figura 6.13: Prueba 8. Varias piedras grandes, medianas y pequeñas sobre la superficie capturada por la cámara.



(a) Imagen capturada por la cámara izquierda. (b) Gráfica de la red de triángulos.

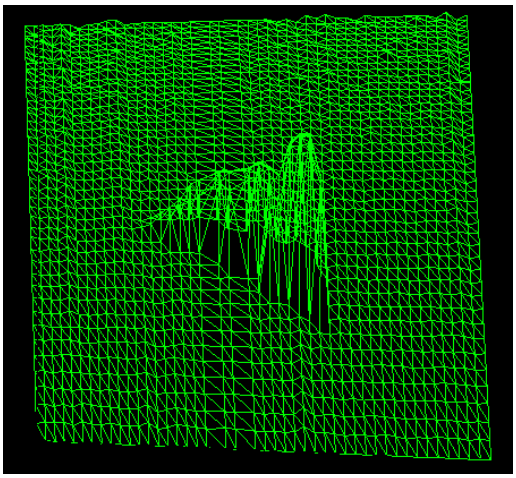
Figura 6.14: Prueba 9. Una piedra al lado de otra sobre la superficie capturada por la cámara.



(a) Imagen capturada por la cámara izquierda. (b) Gráfica de la red de triángulos.

Figura 6.15: Prueba 10. Una piedra detrás de otra sobre la superficie capturada por la cámara.

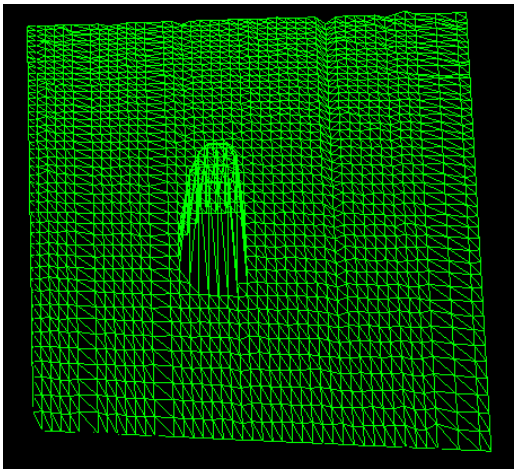
(a) Imagen capturada por la cámara izquierda.



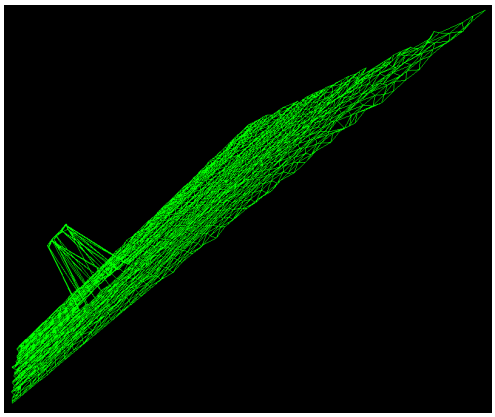
(b) Gráfica de la red de triángulos.

Figura 6.16: Prueba 11. Dinosaurio de plástico (objeto irregular) sobre la superficie capturada por la cámara.

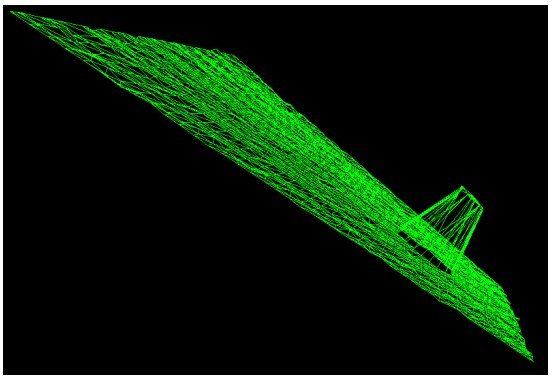
(a) Imagen capturada por la cámara izquierda.



(b) Gráfica de la red de triángulos.



(c) Gráfica de la red de triángulos vista desde la izquierda.



(d) Gráfica de la red de triángulos vista desde la derecha.

Figura 6.17: Prueba 12. Cilindro de piedra (objeto alto) sobre la superficie capturada por la cámara.

Capítulo 7

Conclusiones y recomendaciones

7.1. Conclusiones

Para poder implementar en el lenguaje de programación C el procedimiento matemático propuesto, capaz de generar una red de triángulos que describe y estima la forma irregular de la porción de la superficie que está siendo capturada por la cámara estereoscópica y, que determina la posición y orientación de la red de triángulos con respecto al sistema de coordenadas de la cámara izquierda, es necesario que primero se calibre el arreglo estereoscópico a partir de un nodo de ROS llamado *camera_calibration*, este calcula los parámetros que permiten corregir cualquier tipo de distorsión causada por los lentes de las cámaras.

Con la cámara estereoscópica calibrada, es posible obtener una nube de puntos a partir de las dos imágenes capturadas por esta. Para ello se busca un conjunto de puntos característicos en la imagen capturada por la cámara izquierda y también en la imagen capturada por la cámara derecha, de tal forma que correspondan al mismo píxel en ambas imágenes, estos son puntos que sobresalen en la imagen. Por cada uno de los puntos característicos encontrados se trazan líneas, conocidas como línea vista izquierda y línea vista derecha, que van desde el origen del sistema de coordenadas de la cámara izquierda y de la cámara derecha respectivamente, pasando por el respectivo punto característico en el plano de la cámara y se extienden hasta el infinito. Al elegir el sistema de coordenadas de la cámara izquierda como el sistema de coordenadas de referencia, se puede referir cada una de las líneas vistas derechas a este, a partir de la ecuación de pose, siempre y cuando se conozca la posición y orientación del sistema de coordenadas de la cámara derecha respecto al de la cámara izquierda. Se puede obtener la posición tridimensional respecto al sistema de coordenadas de la cámara izquierda de cada uno de los puntos característicos, a partir de la intersección de la línea vista izquierda con la línea vista derecha referenciada al sistema de coordenadas de la cámara izquierda.

Con el uso de la plataforma de programación ROS se puede obtener la nube de puntos a partir de una cámara estereoscópica, esto mediante dos nodos de ROS llamados camera1394stereo y *stereo_image_proc*. El primero se encarga de obtener la imagen izquierda y la imagen derecha calibradas y sin distorsión, y el segundo toma esas imágenes e implementa el respectivo procedimiento para obtener la nube de puntos.

A partir de la nube de puntos, es decir, a partir del conjunto de puntos de los cuales se co-

noce sus posiciones tridimensionales respecto al sistema de coordenadas de referencia (cámara izquierda) y usando el método iterativo conocido como RANSAC, junto con un método que se basa en el corte de líneas equidistantes del campo de profundidad, es posible generar una red de triángulos que estima la forma tridimensional de la porción de la superficie que está siendo capturada por la cámara.

A partir de uno de los vértices que conforman la red de triángulos, específicamente el primer vértice ubicado en la esquina inferior izquierda, se puede establecer un sistema de coordenadas local sobre la red de triángulos, con origen en ese vértice y que está alineado con el sistema de coordenadas de la cámara izquierda. De esta forma, se puede describir la posición y orientación de la red con respecto al sistema de coordenadas de la cámara izquierda, haciendo uso de la ecuación de pose y de la matriz de rotación.

Se logró comprobar la funcionalidad del algoritmo al someterlo a varias pruebas que consisten en variar la forma irregular de la superficie que está siendo capturada por la cámara izquierda, mediante distintos objetos de diferentes formas y tamaños que se colocan sobre la superficie del espacio destinado para realizar dichas pruebas experimentales. Los resultados indicaron que en general el algoritmo es capaz de dar una aproximación de la forma irregular de la superficie que está siendo capturada por la cámara mediante una red rígida de triángulos.

Cuando se tiene un único objeto sobre la superficie que está siendo capturada por la cámara, a excepción de objetos altos o con formas muy irregulares, la red de triángulos generada por el algoritmo aproxima la forma irregular de la superficie capturada por la cámara izquierda. En este se muestra la forma y tamaño estimado del objeto que sobresale de la superficie plana. Si se tiene más de un objeto sobre la superficie que está siendo capturada por la cámara, la red de triángulos es capaz de dar una aproximación de la forma irregular de la superficie, pero surge el problema del faltante de columnas. Sucede porque hay columnas de la imagen cuya porción de la nube de puntos a la que están asociadas es prácticamente nula o muy pequeña, es decir, para la mayoría de los píxeles de esas columnas se desconoce su respectiva posición tridimensional, y entonces el algoritmo desecha este tipo de columnas. La cantidad de puntos tridimensionales obtenidos varía según la forma que tenga la superficie capturada por la cámara, es decir, dependiendo de la superficie se encuentran más o menos puntos característicos entre un caso y otro. Entre más objetos se tengan sobre la superficie, más difícil será encontrar puntos característicos.

Si se tienen dos objetos pegados, uno al lado del otro o uno detrás del otro, sobre la superficie, la respectiva red de triángulos da una aproximación de lo que está pasando en la realidad, es decir, se puede determinar fácilmente que hay un objeto que sobresale de la superficie plana que está pegado a otro objeto que también sobresale.

Cuando se tiene un objeto alto o un objeto muy irregular sobre la superficie, la red de triángulos obtenida da una mala aproximación de la forma general que tiene el objeto, solamente aproxima la silueta superior de este. Se generan líneas que van desde el suelo hasta la silueta superior, ignorando el resto del objeto. Lo anterior se debe a que la búsqueda de los puntos característicos se hace demasiado difícil y por lo tanto se obtienen muy pocas posiciones tridimensionales en el objeto, en este caso solo se obtienen puntos en la parte de arriba.

7.2. Recomendaciones

Para mejorar y optimizar el funcionamiento del algoritmo y para mejorar las pruebas experimentales realizadas, se proponen las siguientes recomendaciones:

- Desarrollar, implementar y probar un método que permita calcular el error de estimación de forma, entre la forma descrita por la red de triángulos generada por el algoritmo y la forma real de la superficie, tomando como forma real la forma descrita por el campo de profundidad generado por la cámara estereoscópica (la nube de puntos). Este método es necesario para poder medir la precisión y exactitud con la que el algoritmo aproxima y estima la forma irregular de la superficie capturada por la cámara con una red rígida de triángulos y sirve como punto de comparación numeraria entre una prueba y otra.

- Modificar los parámetros de control del algoritmo para corregir el problema del faltante de columnas en la red de triángulos generada por el algoritmo, que se presenta cuando se tiene más de un objeto sobre la superficie plana que está siendo capturada por la cámara. Específicamente, se debe modificar el valor del parámetro que define el número de columnas de la red de triángulos y el parámetro que define el porcentaje mínimo de posiciones tridimensionales que debe de tener una columna para ser considerada, estos se deben de aumentar independientemente hasta que el problema se haya corregido.

- Modificar y mejorar el procedimiento matemático en el que se basa el algoritmo, para que este sea capaz de estimar y aproximar lo mejor posible la forma de cualquier objeto, especialmente objetos altos o irregulares, que se encuentre sobre la superficie plana que está siendo capturada por la cámara, es decir, mejorar el funcionamiento del algoritmo cuando se tienen ese tipo de objetos. Para ello, se pueden dividir las columnas en subcolumnas y obtener una línea de mejor ajuste por cada una de ellas.

- Plantear una serie de pruebas que permitan determinar cuáles son las condiciones óptimas para obtener la mayor cantidad de puntos característicos y, por lo tanto, para obtener la mayor cantidad de posiciones tridimensionales. Para ello, se puede variar el tipo de suelo y la iluminación.

Apéndice A

Funcionamiento de ROS

El Sistema Operativo Robótico (ROS, por sus siglas en inglés) es un marco o plataforma flexible, es decir, es un entorno de trabajo que se utiliza para el desarrollo de software en robótica. Los principales objetivos de ROS y la razón de por qué lo convierte en una gran opción para la implementación de aplicaciones en robótica, se enumeran a continuación:

- Lograr una comunicación basada en una red de pares.
- Aceptar diferentes lenguajes de programación.
- Brindar diferentes herramientas y bibliotecas de uso abierto.

La principal característica de ROS es que las tareas o procesos se llevan a cabo mediante programas que se conocen con el nombre de nodos. Además, ROS permite la conexión de varios de estos nodos y que estos se puedan comunicar entre si utilizando mensajes que se pasan a través de tópicos o temas específicos. Los nodos tienen la capacidad de suscribirse a los tópicos para poder recibir mensajes o publicar a través de los tópicos para poder enviar mensajes. La interacción entre dos nodos a través de un tópico se puede observar en la figura A.1, es decir, la forma en cómo se comunican los nodos en ROS.

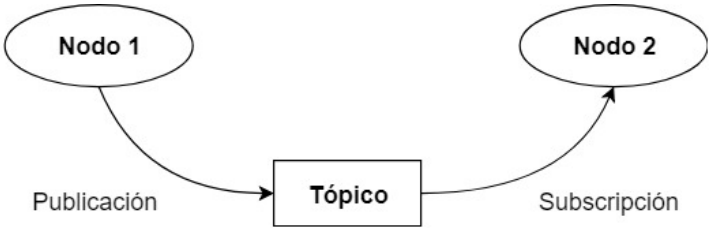


Figura A.1: Comunicación entre dos nodos a través de un tópico en ROS.

La comunicación es posible gracias a un *maestro*, este se encarga del registro de los nodos. Si no hubiera un *maestro* presente, los nodos no podrían encontrar a otros nodos y, por lo tanto, no podrían intercambiar mensajes entre sí. En la figura A.2, se muestra como dos nodos deben

de registrarse a través del *maestro* para que estos se puedan comunicar entre si a través de un tópico.

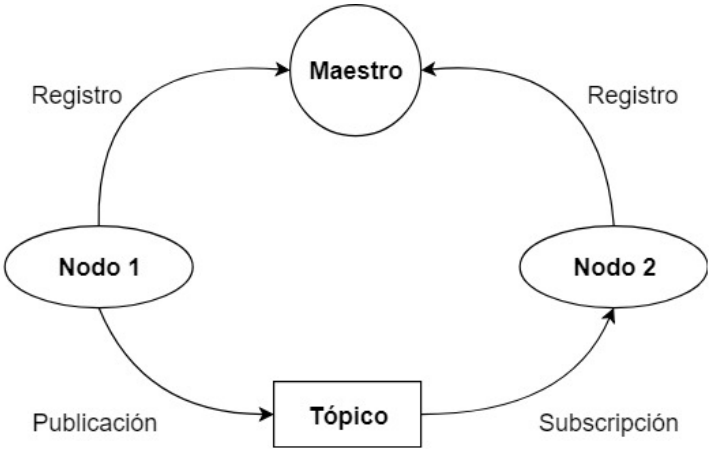


Figura A.2: Función del maestro en ROS.

A manera de ejemplo, considérese la sección 4.4 en la cual se menciona que se usaron tres nodos de ROS para poder obtener la nube de puntos a partir de las imágenes capturadas por una cámara estereoscópica, en la figura A.3 se muestra de una manera más específica y detallada la comunicación entre estos nodos. El nodo llamado $camera1394stereo$ toma las imágenes de la cámara y las publica como un mensaje a través del tópico llamado $/image_raw$. Luego, se corre el nodo $camera_calibration$, el cual se suscribe a $/image_raw$ y procede a realizar la calibración de la cámara, calcula los respectivos parámetros. Por último, se corre el nodo $stereo_image_proc$ que toma la información del tópico $/image_raw$ (se subscribe) y utiliza los parámetros de calibración para luego publicar a través del tópico $/image_rect$ las imágenes calibradas y además, publica la nube de puntos través del tópico llamado $/points2$.

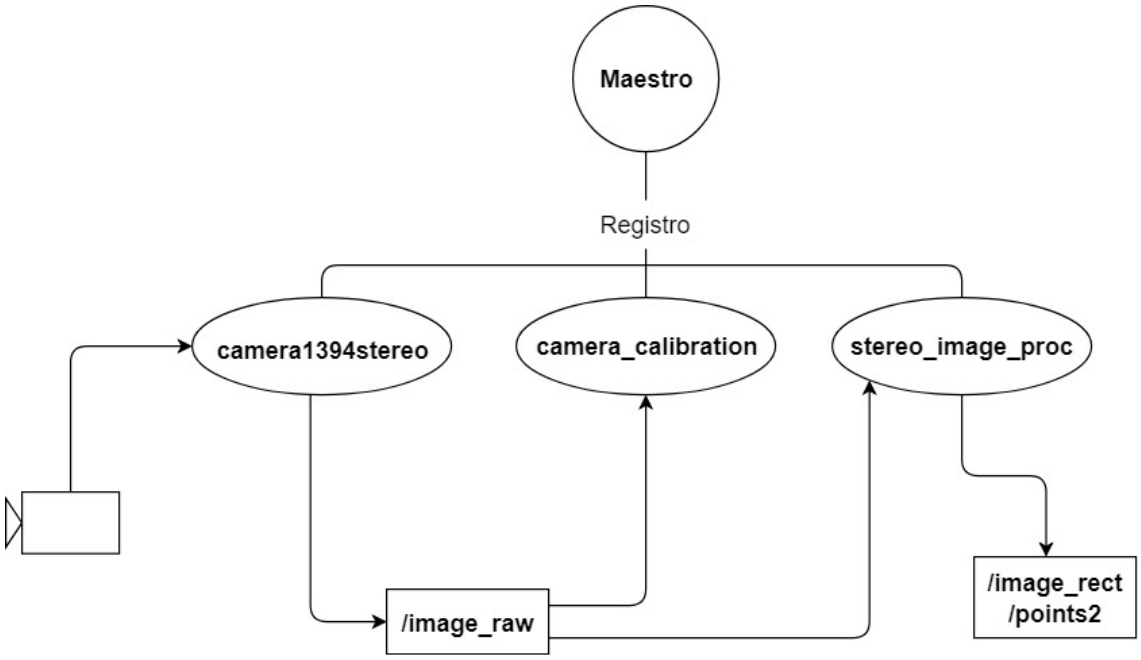


Figura A.3: Ejemplo del uso de tres nodos de ROS para obtener una nube de puntos a partir de una cámara estereoscópica.

Apéndice B

Algoritmo de RANSAC

El algoritmo *RandomSampleConsensus* (RANSAC) fue publicado por primera vez por Fischler y Bolles en SRI International en 1981 [1]. Ellos propusieron este algoritmo para resolver el problema de determinación de ubicación (LDP), donde el objetivo principal era determinar los puntos en el espacio que se proyectan en una imagen en un conjunto de puntos de referencia con ubicaciones desconocidas.

Este algoritmo es un estimador robusto general, consiste en un método iterativo que permite estimar los parámetros de un modelo matemático a partir de un conjunto de datos observados que contienen valores atípicos (en inglés se les conoce como outliers), cuando no se le debe de dar importancia a los valores atípicos en los valores de las estimaciones; es por esto que también se considera como un método de detección de valores atípicos.

Dicho de otra forma, es un enfoque de estimación de parámetros general que fue diseñado para hacer frente a una gran proporción de valores atípicos dentro del conjunto de datos. A diferencia de muchas de las técnicas de estimación robustas comunes, como los estimadores M y los mínimos cuadrados que han sido adoptados por la comunidad de visión artificial, RANSAC se desarrolló y perfecciono desde dentro de la comunidad de visión artificial. Es una técnica de remuestreo que genera soluciones candidatas utilizando un número mínimo de observaciones necesarios para estimar los parámetros del modelo subyacente. Como lo mencionan Fischler y Bolles [1], a diferencia de las técnicas de muestreo convencionales que utilizan la mayor cantidad de datos posible para obtener una solución inicial y luego proceder a eliminar los valores atípicos, RANSAC utiliza el conjunto más pequeño posible y procede a ampliar este conjunto con puntos de datos consistentes.

B.1. Algoritmo

El principal objetivo es obtener un ajuste robusto de un modelo a un conjunto de datos S que contiene valores atípicos. De acuerdo con [7], el algoritmo básico se puede resumir de la siguiente manera:

1. Se selecciona aleatoriamente una muestra de s puntos de datos del conjunto S y se crea una instancia del modelo a partir de este subconjunto.

2. Se determina el conjunto de puntos de datos S_i que están dentro de un umbral de distancia t del modelo. El conjunto S_i es el conjunto de consenso de la muestra y define los valores típicos (inliers) de S.

3. Si el tamaño de S_i (el número de inliers) es mayor que algún umbral T, se re-estima el modelo usando todos los puntos de S_i y se termina el algoritmo.

4. Por el contrario, si el tamaño de S_i es menor que T, se selecciona un nuevo subconjunto y se repite lo anterior (máximo N veces).

5. Después de N iteraciones, se selecciona el mayor conjunto de consenso S_i, y el modelo es re-estimado utilizando todos los puntos en el subconjunto S_i

B.1.1. Umbral de distancia t para determinar la compatibilidad dato/modelo

B.1.2. Umbral de distancia t para determinar la compatibilidad dato/modelo

El umbral t se elige tal que la probabilidad de que un punto sea un $inlier$ es α. Para poder calcular el valor de t se requiere conoce la distribución de probabilidad de la distancia de un $inlier$ del modelo. En la práctica se asume que el error es gaussiano de media cero y varianza σ.

B.1.3. Numero de iteraciones N

El número de iteraciones N, se elige lo suficientemente alto como para garantizar que la probabilidad p (generalmente establecida en 0,99) de que al menos uno de los conjuntos de muestras aleatorios no incluya un valor atípico. Se supone que u representa la probabilidad de que cualquier punto de datos seleccionado sea un valor típico (inlier) y $v = 1-u$ representa la probabilidad de observar un valor atípico (outlier). Se requieren N iteraciones del número mínimo de puntos denotados como m, por lo tanto $1 - p$ esta dado por la expresión (B.1).

$$1 - p = (1 - u^m)^N \tag{B.1}$$

Con un poco de manipulación algebraica, se puede despejar el valor de N que está dado por la expresión (B.2).

$$N = \frac{log(1 - p)}{log(1 - (1 - v)^m)} \tag{B.2}$$

B.1.4. Número mínimo T de puntos en el subconjunto de decisión

El consenso aceptable establecido T, se elegí de tal forma que el algoritmo se termine si el tamaño del conjunto de consenso es similar al número de valores típicos (inliers) que se cree que hay en el conjunto de datos, dada la proporción supuesta de valores atípicos, es decir, para n puntos de datos T estaría dado por la expresión (B.3).

$$T = (1 - v) \cdot n \tag{B.3}$$

Bibliografía

[1] M.A. Fischler & R.C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[2] G. Bradski & A. Kaehler. *Learning OpenCV*, chapter 10, pages 316–319. O'Reilly Media, Inc., 1st. edition, 2008.

[3] G. Oviedo. *Diseño, Implementación y Prueba de un Nodo ROS para Visualizar Objetos Tridimensionales Basado en OpenGL.* Escuela de Ingeniería Eléctrica, Universidad de Costa Rica, Febrero 2013.

[4] J. Shi & C. Tomasi. Good features to track. *9th IEEE Conference on Computer Vision and Pattern Recognition*, Junio 1994.

[5] V. Purchart & T. Pasek & I. Kolingerova & P. Vanecek. Haptic visualization of material on tin-based surfaces. *Computer Vision and Graphics: International Conference.*, pages 220–227, 2012.

[6] Z. Zhang. A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(11):1330–1334, 2000.

[7] R. Hartley & A. Zisserman. *Multiple View Geometry in Computer Vision*, chapter 4, pages 117–123. University Press, Cambridge, 2nd. edition, 2004.