

**Universidad de Costa Rica
Facultad de Ingeniería
Escuela de Ingeniería Eléctrica**

**ALGORITMO DE ODOMETRÍA VISUAL
ESTEREOSCÓPICO BASADO EN LA
MINIMIZACIÓN DEL ERROR
CUADRÁTICO MEDIO DE
CORRESPONDENCIAS TÍPICAS POR
DESCOMPOSICIÓN EN VALORES
SINGULARES**

**Por:
SANTIAGO JESÚS MORA ZÚÑIGA**

IPCV-LAB

Ciudad Universitaria Rodrigo Facio

Febrero de 2021

**ALGORITMO DE ODOMETRÍA VISUAL
ESTEREOSCÓPICO BASADO EN LA
MINIMIZACIÓN DEL ERROR
CUADRÁTICO MEDIO DE
CORRESPONDENCIAS TÍPICAS POR
DESCOMPOSICIÓN EN VALORES
SINGULARES**

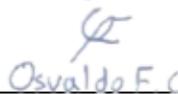
Por:

SANTIAGO JESÚS MORA ZÚÑIGA

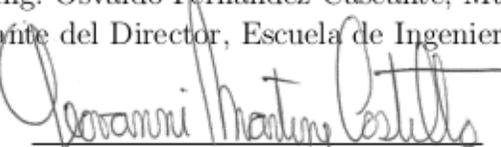
Sometido a la Escuela de Ingeniería Eléctrica de la Facultad de
Ingeniería de la Universidad de Costa Rica como requisito parcial para
optar por el grado de:

LICENCIATURA EN INGENIERÍA ELÉCTRICA

Aprobado por el Tribunal:

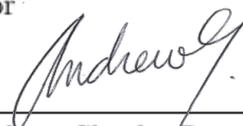

Osvaldo F. C.

Ing. Osvaldo Fernández Cascante, Mtr.
Representante del Director, Escuela de Ingeniería Eléctrica


Dr. Ing. Giovanni Martínez Castillo
Director, Comité Asesor



Ing. Peter De Ford González, MSc.
Miembro, Comité Asesor



Ing. Andrew Sheehy/Protti, Lic.
Miembro, Comité Asesor



Ing. Nelson Porrás Moreno, Lic.
Miembro del Tribunal

Dedicatoria

A Dios, el Eterno Padre, por brindarme salud, fortaleza y sabiduría, por ser la luz incondicional que ha guiado mi camino y por haberme permitido llegar a este momento tan importante no solo de vida, también de mi formación profesional.

A mis padres y abuelos, por ser los pilares fundamentales en mi aprendizaje y desarrollo a lo largo de toda mi vida y por su gran amor, cariño, comprensión, dedicación y apoyo incondicional en los buenos y malos momentos

Al resto de mi familia, por apoyarme y creer en mi en todo momento.

Reconocimientos

Me gustaría transmitir mi absoluta gratitud a mi profesor tutor, el Dr.-Ing. Geovanni Martínez, coordinador del Laboratorio de Investigación en Procesamiento Digital de Imágenes y Visión por Computador (IPCV-LAB), por haberme guiado a través de la mayor parte de este trabajo, por su amable pero rigurosa supervisión y retroalimentación que me dieron constantemente la motivación para desempeñar mi máxima capacidad, por la transmisión de sus valiosos conocimientos y sabiduría que adquirió a lo largo de toda su carrera como investigador, que me hicieron crecer no solo como profesional, también como persona. Le doy gracias desde lo más profundo de mi ser por su gran disposición, ayuda y consejos, por su amor a los estudiantes y sobre todo por su amor al conocimiento y a la investigación. Tuve mucha suerte de haberlo tenido como profesor tutor, su papel como tal supera cualquier expectativa que alguien se pueda imaginar.

También quiero agradecer a toda mi familia y amigos cercanos, por su tremendo aliento, apoyo y paciencia a lo largo de este trabajo y porque de una u otra forma lograron aportar un granito de arena para hacer posible la culminación de esta etapa de mi carrera profesional.

Índice general

Dedicatoria	iii
Reconocimientos	iv
Índice de figuras	viii
Índice de cuadros	xiii
Nomenclatura	xiv
Resumen	xxix
1 Introducción	1
1.1 Introducción al tema	1
1.2 Justificación y Propuesta	6
1.3 Metodología	7
1.4 Objetivos	9
1.4.1 Objetivo general	9
1.4.2 Objetivos específicos	9
2 Desarrollo teórico	10
2.1 Conceptos, métodos y modelos básicos en visión por computador	10
2.1.1 Modelo de iluminación	10
2.1.2 Modelo de color	11
2.1.3 Modelo de posición de un objeto rígido	14
2.1.4 Modelo de orientación de un objeto rígido	15
2.1.5 Modelo de pose de un objeto rígido	18
2.1.6 Modelo de movimiento de un objeto rígido	19
2.1.7 Odometría de ruedas	23
2.1.8 Unidad de medición inercial	26
2.1.9 Modelo de pinhole de una cámara	28
2.2 Estereoscopía	33
2.2.1 Geometría de lo sistemas estereoscópicos	34
2.2.2 Modelo canónico de una cámara estereoscópica	35
2.2.3 Estimación de profundidad por estereoscopía	37
2.2.4 Obtención de una nube de puntos por estereoscopía	42

2.2.5	Obtención de una nube de puntos usando el Sistema Operativo Robótico (ROS)	43
3	Algoritmo de odometría visual estereoscópico	45
3.1	Definición y algoritmo general	45
3.2	Obtención de las nubes de puntos e imágenes	51
3.3	Establecimiento de correspondencias temporales	56
3.4	Estimación inicial de movimiento	71
3.4.1	Solución basada en la descomposición en valores singulares	77
3.4.2	Búsqueda de correspondencias típicas mediante RANSAC	81
3.5	Implementación del algoritmo	85
4	Pruebas experimentales	91
4.1	Obtención de los datos de entrada de prueba	91
4.1.1	Equipo utilizado	92
4.1.2	Entorno controlado de pruebas	95
4.1.3	Imagen de intensidad izquierda y nube de puntos de prueba	99
4.2	Planteamiento de las pruebas	100
4.2.1	Prueba I: establecimiento de correspondencias temporales	101
4.2.2	Prueba II: estimación de movimiento	112
4.2.3	Prueba III: algoritmo completo excluyendo la parte de ROS	121
5	Resultados de las pruebas y análisis	132
5.1	Resultados de la prueba I	132
5.2	Resultados de la prueba II	140
5.3	Resultados de la prueba III	149
6	Conclusiones y recomendaciones	159
6.1	Conclusiones	159
6.2	Recomendaciones	165
	Bibliografía	170
A	Funcionamiento de ROS	177
A.1	Sistema de archivos	177
A.2	Computación a nivel gráfico	179
B	Algoritmo de RANSAC	182
B.1	Algoritmo	183
B.2	Parámetros	183

B.2.1	Umbral de distancia t para determinar la compatibilidad	
	dato/modelo	183
B.2.2	Numero de iteraciones N	183
B.2.3	Número mínimo T de puntos en el subconjunto de decisión	184

Índice de figuras

2.1	Modelo de iluminación difusa e invariante (Martínez, 2019b).	12
2.2	Calculo de la imagen de intensidad a partir de una imagen RGB (Martínez, 2019b).	13
2.3	Posición de cualquier punto de un objeto cuando ambos sistemas de coordenadas están alineados (Martínez, 2019b).	15
2.4	Método de los ángulos fijos X-Y-Z para describir la orientación de un objeto respecto a una referencia (Martínez, 2019b).	16
2.5	Posición de cualquier punto de un objeto cuando el origen de su sistema de coordenadas coincide con el origen del sistema de referencia (Martínez, 2019b).	18
2.6	Posición de cualquier punto de un objeto cuando su sistema de coordenadas no está alineado y su origen no coincide con el sistema de referencia (Martínez, 2019b).	19
2.7	Posición de cualquier punto de un objeto después de una traslación y rotación, es decir, después de algún movimiento (Martínez, 2019b).	21
2.8	Diagrama de las partes y el funcionamiento de un codificador óptico (Martínez, 2019b).	24
2.9	Geometría detallada de la odometría de ruedas (Martínez, 2019b).	25
2.10	Esquemático de una unidad de medición inercial respecto al sistema de referencia (Martínez, 2019b).	27
2.11	Acelerómetro tridimensional compuesto por tres sensores montados ortogonalmente.	27
2.12	Giroscopio tridimensional compuesto por tres sensores montados ortogonalmente.	28
2.13	Modelo de pinhole de una cámara (Martínez, 2019b).	29
2.14	Vista del plano YZ del modelo de pinhole de una cámara.	31
2.15	Relación entre el sistema de coordenadas del plano de la cámara, el sistema de coordenadas de la imagen y los sensores (Pajares y de la Escalera, 2016).	32
2.16	Representación de la geometría epipolar que relaciona dos imágenes de una misma escena (Martínez, 2019b).	35
2.17	Modelo estéreo canónico (Martínez, 2019b; Pajares y de la Escalera, 2016).	36
2.18	Establecimiento de correspondencias estereoscópicas (Bradski y Kaehler, 2008).	39

2.19	Triangulación estereoscópica de un punto de la escena (Martínez, 2019b).	41
2.20	Nube de puntos respecto al sistema de coordenadas de la cámara izquierda.	43
2.21	Implementación de nodos de ROS para obtener una nube de puntos.	44
3.1	Esquema general de un algoritmo de odometría visual estereoscópico.	48
3.2	Esquema general de un algoritmo de odometría visual estereoscópico, donde las nubes de puntos son proporcionadas por un proceso externo.	50
3.3	Protocolo para obtener las nubes de puntos e imágenes antes y después del movimiento, proporcionadas a través de tópicos de ROS mediante un proceso externo.	52
3.4	Numeración de cada uno de los píxeles de la imagen izquierda antes del movimiento.	53
3.5	J-ésimo píxel de la imagen de intensidad izquierda antes del movimiento.	53
3.6	Numeración de cada uno de los puntos característicos en la imagen de izquierda antes del movimiento.	54
3.7	I-ésimo punto característico en la imagen de intensidad izquierda antes del movimiento.	54
3.8	Subconjuntos de los puntos característicos y las nubes de puntos antes y después del movimiento, así como las correspondencias temporales entre estos.	58
3.9	Punto característico elegido de forma aleatoria en la imagen izquierda antes del movimiento.	59
3.10	Ventana de correlación en la imagen izquierda antes del movimiento, con centro en el punto característico elegido aleatoriamente.	60
3.11	Ventana de búsqueda en la imagen izquierda después del movimiento, con centro en las mismas coordenadas del punto característico elegido aleatoriamente en la imagen izquierda antes del movimiento.	63
3.12	Puntos característicos dentro de la ventana de búsqueda y su número de aparición.	64
3.13	Ventana de correlación en la imagen izquierda después del movimiento, con centro en un punto característico que se encuentra dentro de la ventana de búsqueda.	65
3.14	Comparación entre la ventana de correlación en la imagen izquierda antes del movimiento y las ventanas de correlación en la imagen izquierda después del movimiento.	66
3.15	Vector que se forma entre el centro de las ventanas de correlación, si estas se encuentran en una misma imagen ficticia.	68

3.16	Relación que existe entre un píxel de la ventana V_c y el píxel equivalente en la ventana V_c, pc_n	68
3.17	Ventana de correlación en la imagen después del movimiento con el coeficiente de similitud máximo.	69
3.18	Establecimiento de una correspondencia temporal entre un punto característico elegido aleatoriamente en la imagen izquierda antes del movimiento y un punto característico elegido cuidadosamente en la imagen izquierda después del movimiento.	72
3.19	Coincidencia ideal entre los puntos tridimensionales \mathbf{Q}_{s,t_k} transformados y los puntos tridimensionales $\mathbf{Q}_{s,t_{k+1}}$, en el instante de tiempo t_{k+1} después del movimiento.	74
3.20	Coincidencia con ruido entre los puntos tridimensionales \mathbf{Q}_{s,t_k} transformados y los puntos tridimensionales $\mathbf{Q}_{s,t_{k+1}}$, en el instante de tiempo t_{k+1} después del movimiento.	75
3.21	Coincidencia con ruido entre los puntos tridimensionales \mathbf{Q}_{s,t_k} transformados y los puntos tridimensionales $\mathbf{Q}_{s,t_{k+1}}$, en el instante de tiempo t_{k+1} después del movimiento.	76
3.22	Correspondencias temporales 2D típicas y atípicas entre un punto bidimensional en la imagen izquierda antes del movimiento y un punto bidimensional en la imagen izquierda después del movimiento, así como las respectivas correspondencias temporales 3D típicas y atípicas entre sus posiciones tridimensionales.	82
3.23	Clasificación de correspondencias temporales en típicas o atípicas de acuerdo a la distancia entre los puntos característicos \mathbf{q}'_{ls,t_k} y $\mathbf{q}_{ls,t_{k+1}}$	84
3.24	Diagrama general del algoritmo completo donde se muestra la comunicación entre todos los nodos de ROS, así como las salidas y entradas del algoritmo o nodo principal.	90
4.1	Cámara estereoscópica Bumblebee 2 de la marca Point Grey.	92
4.2	Plataforma de desarrollo robótico Husky A200 de la marca Clearpath Robotics.	94
4.3	Espacio controlado de pruebas, ubicado dentro del laboratorio IPCV-LAB de la Universidad de Costa Rica.	95
4.4	Superficie utilizada dentro del campo de visión de la cámara estereoscópica atada rígidamente a la plataforma robótica dentro del entorno controlado.	97
4.5	Foco proyectores LED usados para iluminar de forma difusa el entorno controlado.	97
4.6	Inclinación horizontal de la cámara estereoscópica atada rígidamente a la plataforma robótica dentro del entorno controlado.	98

4.7	Inclinación vertical de la cámara estereoscópica atada rígidamente a la plataforma robótica dentro del entorno controlado.	98
4.8	Altura con respecto al suelo de la cámara estereoscópica atada rígidamente a la plataforma robótica dentro del entorno controlado.	98
4.9	Imagen de intensidad izquierda de prueba.	99
4.10	Imagen después del movimiento obtenida a partir de una transformación euclidiana de la imagen antes del movimiento, llevada a cabo por la función <i>warpaffine</i>	104
4.11	Obtención del límite de traslación para que en el peor caso posible no se pierda más de un 40% de la imagen antes del movimiento transformada.	105
4.12	Matriz de confusión para la clasificación de las correspondencias.	109
4.13	Obtención del límite de traslación tridimensional de la nube de puntos de prueba a partir del límite de traslación bidimensional de la imagen izquierda prueba.	114
5.1	Gráficas de la media y la varianza del error de traslación, contra la varianza del error de posición introducido a la segunda nube de puntos en la segunda prueba.	141
	(a) Media del error de traslación en X_l (en azul), media del error de traslación en Y_l (en anaranjado) y media del error de traslación en Z_l (en amarillo).	141
	(b) Varianza del error de traslación en X_l (en azul), varianza del error de traslación en Y_l (en anaranjado) y varianza del error de traslación en Z_l (en amarillo).	141
5.2	Gráficas de la media y la varianza del error de rotación, contra la varianza del error de posición introducido a la segunda nube de puntos en la segunda prueba.	142
	(a) Media del error de rotación en X_l (en azul), media del error de rotación en Y_l (en anaranjado) y media del error de rotación en Z_l (en amarillo).	142
	(b) Varianza del error de rotación en X_l (en azul), varianza del error de rotación en Y_l (en anaranjado) y varianza del error de rotación en Z_l (en amarillo).	142
5.3	Gráficas de la media y la varianza del error de coincidencia entre las nubes de puntos, contra la varianza del error de posición introducido a la segunda nube de puntos en la segunda prueba.	143
	(a) Media del error de coincidencia.	143
	(b) Varianza del error de coincidencia.	143
5.4	Gráficas de la media y la varianza del error de traslación, contra la varianza del ruido introducido a los valores de intensidad de la segunda imagen izquierda en la tercera prueba.	150

(a)	Media del error de traslación en X_l (en azul) y media del error de traslación en Y_l (en anaranjado).	150
(b)	Varianza del error de traslación en X_l (en azul) y varianza del error de traslación en Y_l (en anaranjado).	150
5.5	Gráficas de la media y la varianza del error de rotación, contra la varianza del ruido introducido a los valores de intensidad de la segunda imagen izquierda en la tercera prueba.	151
(a)	Media del error de rotación en Z_l	151
(b)	Varianza del error de rotación en Z_l	151
5.6	Gráficas de la media y la varianza del error de coincidencia entre las nubes de puntos, contra la varianza del ruido introducido a los valores de intensidad de la segunda imagen izquierda en la tercera prueba.	152
(a)	Media del error de coincidencia.	152
(b)	Varianza del error de coincidencia.	152
A.1	Esquema del nivel de sistema de archivos de ROS.	178
A.2	Esquema de la jerarquización de ROS.	179
A.3	Comunicación entre nodos a través de un tópico.	180
A.4	Ejemplo del uso de tres nodos de ROS para obtener una nube de puntos a partir de una cámara estereoscópica.	181

Índice de cuadros

5.1	Resultados de los casos con un límite de rotación igual a 0 grados ($l_\theta = 0^\circ$).	132
5.2	Resultados de los casos con un límite de rotación igual a 5 grados ($l_\theta = 5^\circ$).	132
5.3	Resultados de los casos con un límite de rotación igual a 10 grados ($l_\theta = 10^\circ$).	133
5.4	Resultados de los casos con un límite de rotación igual a 15 grados ($l_\theta = 15^\circ$).	133
5.5	Resultados de los casos con un límite de rotación igual a 20 grados ($l_\theta = 20^\circ$).	133

Nomenclatura

a_q	Numero elegido aleatoriamente entre 1 y la cantidad de puntos tridimensionales que tiene la nube antes del movimiento $C_{N_{t_k}}$.
A	Vector dado por $(A_R, A_S, A_T)^T$ que describe aceleración lineal de una unidad de medición inercial, respecto a su sistema local.
B	Distancia de la línea base de un arreglo estereoscópico.
B	Vector multidimensional que representa el movimiento de un objeto respecto a un sistema de referencia. Depende de las componentes de la traslación y de los ángulos de rotación.
cpn_{max}	Coefficiente de similitud máximo asociado a la ventana de correlación $V_{c,pcn_{max}}$.
cpn_{min}	Coefficiente de similitud mínimo que debe de tener una ventana de correlación $V_{c,pcn}$ para poder establecer una correspondencia temporal.
cpn_{pcn}	Coefficiente de similitud asociado a cada una de las ventanas de correlación $V_{c,pcn}$.
$C_{N_{t_k}}$	Cantidad de puntos tridimensionales que tiene la nube de puntos \mathbf{N}_{t_k} antes del movimiento del rover.
$C_{N_{t_{k+1}}}$	Cantidad de puntos tridimensionales que tiene la nube de puntos $\mathbf{N}_{t_{k+1}}$ después del movimiento del rover.
C_q	Tamaño de los subconjuntos formados por los puntos \mathbf{q}_{l_s,t_k} y \mathbf{Q}_{l_s,t_k} antes del movimiento del rover y de los subconjuntos formados por los puntos $\mathbf{q}_{l_s,t_{k+1}}$ y $\mathbf{Q}_{l_s,t_{k+1}}$ después del movimiento.
C	Centro óptico de una cámara monocular.

\mathbf{C}_{img}	Vector dado por $(C_c, C_r)^T$ que describe la posición bidimensional del origen del sistema de coordenadas del plano de una cámara, respecto al de la imagen.
\mathbf{C}_l	Centro óptico de la cámara izquierda de un arreglo estereoscópico.
\mathbf{C}_{l,t_k}	Centro óptico de la cámara izquierda de un arreglo estereoscópico antes del movimiento del rover.
$\mathbf{C}_{l,t_{k+1}}$	Centro óptico de la cámara izquierda de un arreglo estereoscópico después del movimiento del rover.
\mathbf{C}_r	Centro óptico de la cámara derecha de un arreglo estereoscópico.
$d_{baseline}$	Distancia entre las ruedas de un robot móvil.
d_c	Distancia recorrida por el centro del eje de un robot móvil.
d_l	Distancia recorrida por la rueda izquierda de un robot móvil.
d_{qs}	Distancia en píxeles entre los puntos bidimensionales $\mathbf{q}_{l_s,t_{k+1}}$ y $\mathbf{q}'_{l_s,t_{k+1}}$, en la imagen izquierda después del movimiento.
$d_{q,max}$	Distancia máxima entre los puntos bidimensionales $\mathbf{q}_{l_s,t_{k+1}}$ y $\mathbf{q}'_{l_s,t_{k+1}}$, que se utiliza en el algoritmo de RANSAC para clasificar las correspondencias temporales.
d_r	Distancia recorrida por la rueda derecha de un robot móvil.
d_u	Distancia entre sensores a lo largo del eje u del plano de la cámara.
d_v	Distancia entre sensores a lo largo del eje v del plano de la cámara.
\mathbf{D}	Matriz central de la descomposición en valores singulares de la matriz de covarianza $\Sigma_{Q_{t_k-t_{k+1}}}$.
e^2	Error cuadrático medio que es igual a la media del cuadrado de los vectores de error residual \mathbf{e}_s .
e_w^2	Error cuadrático sopesado que es igual a la suma de los cuadrados sopesados de los vectores de error residual \mathbf{e}_s .

\mathbf{e}_l	Epipolo izquierdo de un arreglo estereoscópico.
\mathbf{e}_r	Epipolo derecho de un arreglo estereoscópico.
\mathbf{e}_s	Vector de error residual entre los puntos tridimensionales $\mathbf{Q}_{ls,t_{k+1}}$ y $\mathbf{Q}'_{ls,t_{k+1}}$, después del movimiento del rover.
f	Distancia focal de una cámara monocular o estereoscópica.
<i>FPGA</i>	<i>Field Programmable Gate Array</i> (Arreglos de Compuertas Lógicas Programables en Sitio).
<i>GUID</i>	<i>Globally Unique Identifier</i> (Identificador Único Global).
<i>GPS</i>	<i>Global Positioning System</i> (Sistema de Posicionamiento Global).
\mathbf{G}	Vector dado por $(G_X, G_Y, G_Z)^T$ que describe la posición tridimensional de un objeto respecto a un sistema de referencia.
\mathbf{G}'	Vector dado por $(G'_X, G'_Y, G'_Z)^T$ que describe la posición tridimensional de un objeto después del movimiento, respecto a un sistema de referencia.
\mathbf{G}_r^l	Vector dado por $(B, 0, 0)^T$ que describe la posición tridimensional de la cámara derecha de un arreglo estereoscópico respecto a la cámara izquierda.
h	Largo de una imagen capturada por una cámara monocular o estereoscópica.
h_b	Largo de la ventana de búsqueda V_b en la imagen izquierda después del movimiento.
h_c	Largo de una ventana de correlación V_c o $V_{c,pcn}$ en la imagen izquierda antes o después del movimiento, respectivamente.
\mathbf{h}	Proyección del punto \mathbf{H} sobre el plano de una cámara monocular, dada por el vector bidimensional $(h_u, h_v)^T$.
\mathbf{h}_l	Proyección del punto \mathbf{H} sobre el plano de la cámara izquierda de un arreglo estereoscópico, dada por el vector bidimensional $(h_{u_l}, h_{v_l})^T$.

\mathbf{h}_r	Proyección del punto \mathbf{H} sobre el plano de la cámara derecha de un arreglo estereoscópico, dada por el vector bidimensional $(h_{u_r}, h_{v_r})^T$.
\mathbf{h}'	Proyección del punto \mathbf{H} sobre la imagen de una cámara monocular, dada por el vector bidimensional $(h_c, h_r)^T$.
\mathbf{h}'_l	Proyección del punto \mathbf{H} sobre la imagen izquierda de un arreglo estereoscópico, dada por el vector bidimensional $(h'_{c_l}, h'_{r_l})^T$.
\mathbf{h}'_{la_q, t_k}	Punto característico elegido aleatoriamente dentro del conjunto de puntos característicos \mathbf{h}'_{l_i} , en la imagen izquierda antes del movimiento, dado por el vector bidimensional $(c_{la_q}, r_{la_q})^T$. Se elige de acuerdo al valor de a_q .
\mathbf{h}'_{l_i}	Conjunto de puntos característicos, dados por el vector bidimensional $(h'_{c_{l_i}}, h'_{r_{l_i}})^T$, que representan la proyección de los puntos H_i sobre la imagen izquierda de un arreglo estereoscópico. También se pueden denotar como $\mathbf{h}'_{l_i}(c_{l_i}, r_{l_i})$.
\mathbf{h}'_{l_i, t_k}	Conjunto de puntos característicos, dados por el vector bidimensional $(c_{l_i}, r_{l_i})^T$, que representan la proyección de los puntos tridimensionales H_{i, t_k} sobre la imagen izquierda de un arreglo estereoscópico antes del movimiento del rover.
$\mathbf{h}'_{lm, t_{k+1}}$	Conjunto de puntos característicos, dados por el vector bidimensional $(c_{lm}, r_{lm})^T$, que representan la proyección de los puntos tridimensionales $H_{m, t_{k+1}}$ sobre la imagen izquierda de un arreglo estereoscópico después del movimiento del rover.
$\mathbf{h}'_{lpc_n, t_{k+1}}$	Punto característico dentro de la ventana de búsqueda V_b que pertenece al conjunto de puntos característicos $\mathbf{h}'_{lm, t_{k+1}}$ en la imagen izquierda después del movimiento. Esta dado por el vector bidimensional $(c_{lpc_n}, r_{lpc_n})^T$.
$\mathbf{h}'_{lpc_{nmax}, t_{k+1}}$	Punto característico ubicado en el centro de la ventana de correlación $V_{c, pc_{nmax}}$ en la imagen izquierda después del movimiento. Esta dado por el vector bidimensional $(c_{lpc_{nmax}}, r_{lpc_{nmax}})^T$.
\mathbf{h}'_r	Proyección del punto \mathbf{H} sobre la imagen derecha de un arreglo estereoscópico, dada por el vector bidimensional $(h'_{c_r}, h'_{r_r})^T$.

\mathbf{h}'_{ri}	Conjunto de puntos característicos, dados por el vector $(h'_{c_{ri}}, h'_{r_{ri}})^T$, que representan la proyección de los puntos H_i sobre la imagen derecha de un arreglo estereoscópico. También se pueden denotar como $\mathbf{h}'_{ri}(c_{ri}, r_{ri})$.
\mathbf{h}^*	Proyección del punto \mathbf{H} sobre el plano de la cámara pero con unidades de píxeles, dada por el vector bidimensional $(h_u^*, h_v^*)^T$.
\mathbf{H}	Posición tridimensional de un punto de un objeto, respecto a sus sistema de coordenadas local (puede ser respecto a una cámara monocular $(X, Y, Z)^T$ o a la cámara izquierda de un arreglo estereoscópico $(X_l, Y_l, Z_l)^T$). Esta dada por el vector $(H_X, H_Y, H_Z)^T$ o $(H_{X_l}, H_{Y_l}, H_{Z_l})^T$.
\mathbf{H}_{a_q, t_k}	Posición tridimensional del punto característico \mathbf{h}'_{la_q, t_k} elegido aleatoriamente en la imagen izquierda antes del movimiento, dada por el vector $(X_{a_q}, Y_{a_q}, Z_{a_q})^T$.
\mathbf{H}_i	Conjunto de punto tridimensionales, dados por el vector $(H_{X_{li}}, H_{Y_{li}}, H_{Z_{li}})^T$, que representan las posiciones tridimensionales de los puntos característicos \mathbf{h}'_{li} , respecto a la cámara izquierda de un arreglo estereoscópico. También se pueden representar como $\mathbf{H}_i(X_{li}, Y_{li}, Z_{li})$.
\mathbf{H}_{i, t_k}	Conjunto de punto tridimensionales, dados por el vector $(X_{li}, Y_{li}, Z_{li})^T$, que representan las posiciones tridimensionales de los puntos característicos \mathbf{h}'_{li, t_k} , respecto a la cámara izquierda de un arreglo estereoscópico antes del movimiento del rover.
$\mathbf{H}_{m, t_{k+1}}$	Conjunto de punto tridimensionales, dados por el vector $(X_{lm}, Y_{lm}, Z_{lm})^T$, que representan las posiciones tridimensionales de los puntos característicos $\mathbf{h}'_{lm, t_{k+1}}$, respecto a la cámara izquierda de un arreglo estereoscópico después del movimiento del rover.
$\mathbf{H}_{pc_{nmax}, t_{k+1}}$	Posición tridimensional del punto característico $\mathbf{h}'_{lpc_{nmax}, t_{k+1}}$ en la imagen izquierda después del movimiento, dada por el vector $(X_{pc_{nmax}}, Y_{pc_{nmax}}, Z_{pc_{nmax}})^T$.
\mathbf{H}^*	Posición tridimensional de un punto de un objeto, respecto al sistema de coordenadas local del objeto, dada por el vector $(H_X^*, H_Y^*, H_Z^*)^T$.

\mathbf{H}'	Posición tridimensional de un punto de un objeto después del movimiento, respecto a un sistema de referencia, dada por el vector $(H'_X, H'_Y, H'_Z)^T$.
i	I-ésimo punto característico \mathbf{h}'_{li,t_k} o punto tridimensional \mathbf{H}_{i,t_k} , antes del movimiento del rover.
I_{lj,t_k}	Valor de intensidad de un píxel p_{lj,t_k} en la imagen izquierda antes del movimiento.
$I_{ln,t_{k+1}}$	Valor de intensidad de un píxel p_{ln,t_k} en la imagen izquierda después del movimiento.
I_{p_{vc1},t_k}	Valor de intensidad del primer píxel de la ventana de correlación V_c .
$I_{p_{vc1,pcn},t_{k+1}}$	Valor de intensidad del primer píxel de la ventana de correlación $V_{c,pcn}$.
I_{p_{vcf},t_k}	Valor de intensidad del último píxel de la ventana de correlación V_c .
$I_{p_{vcf,pcn},t_{k+1}}$	Valor de intensidad del último píxel de la ventana de correlación $V_{c,pcn}$.
<i>IMU</i>	<i>Inertial Measurement Unit</i> (Unidad de Medición Inercial).
<i>IPCV – LAB</i>	<i>Image Processing and Computer Vision Research Laboratory</i> (Laboratorio de Investigación en Procesamiento de Imágenes y Visión por Computador).
j	J-ésimo píxel p_{lj,t_k} de la imagen izquierda antes del movimiento del rover.
l	Línea vista.
l_c	Línea base de un arreglo estereoscópico.
l_{el}	Línea epipolar izquierda de un arreglo estereoscópico
l_{er}	Línea epipolar derecha de un arreglo estereoscópico
l_l	Línea vista izquierda de un arreglo estereoscópico.
l_r	Línea vista derecha de un arreglo estereoscópico.

m	M-ésimo punto característico $\mathbf{h}'_{lm,t_{k+1}}$ o punto tridimensional $\mathbf{H}_{m,t_{k+1}}$, después del movimiento del rover.
MER	<i>Mars Exploration Rover</i> (Vehículo de Exploración de Marte).
MSL	<i>Mars Science Laboratory</i> (Laboratorio de Ciencias de Marte).
n	N-ésimo píxel $\mathbf{p}_{ln,t_{k+1}}$ de la imagen izquierda después del movimiento del rover.
n_c	Numero de la correspondencia establecida entre un punto \mathbf{h}'_{la_q,t_k} en la imagen izquierda antes del movimiento y un punto $\mathbf{h}'_{lpc_{n_{max}},t_{k+1}}$ en la imagen izquierda después del movimiento.
n_{pc}	Numero de aparición de un punto característico $\mathbf{h}'_{lpc_n,t_{k+1}}$ dentro de la ventana de búsqueda V_b .
$NASA$	<i>National Aeronautics and Space Administration</i> (Administración Nacional de la Aeronáutica y del Espacio).
N_q	Número máximo de iteraciones del algoritmo de RANSAC.
N_u	Numero de píxeles por sensor a lo largo del eje u del plano de la cámara.
N_v	Numero de píxeles por sensor a lo largo del eje v del plano de la cámara.
\mathbf{N}	Nube de puntos tridimensionales formada por los puntos \mathbf{H}_i .
\mathbf{N}_{t_k}	Nube de puntos tridimensionales formada por los puntos \mathbf{H}_{i,t_k} antes del movimiento del rover.
$\mathbf{N}_{t_{k+1}}$	Nube de puntos tridimensionales formada por los puntos $\mathbf{H}_{i,t_{k+1}}$ después del movimiento del rover.
\mathbf{O}	Vector dado por $(\mathfrak{R}_X, \mathfrak{R}_Y, \mathfrak{R}_Z)^T$ que describe la orientación tridimensional de un objeto respecto a un sistema de referencia.

\mathbf{O}_A	Vector dado por $(\mathfrak{R}_{AR}, \mathfrak{R}_{AS}, \mathfrak{R}_{AT})^T$ que describe la orientación de una unidad de medición inercial, proporcionado por un acelerómetro 3D, respecto a un sistema de referencia.
\mathbf{O}_W	Vector dado por $(\mathfrak{R}_{WR}, \mathfrak{R}_{WS}, \mathfrak{R}_{WT})^T$ que describe la orientación de una unidad de medición inercial, proporcionado por un giroscopio 3D, respecto a un sistema de referencia.
\mathbf{O}'	Vector dado por $(\mathfrak{R}'_X, \mathfrak{R}'_Y, \mathfrak{R}'_Z)^T$ que describe la orientación tridimensional de un objeto después del movimiento, respecto a un sistema de referencia.
p_i	Numero del píxel j asociado a un punto característico \mathbf{h}'_{li,t_k} en la imagen izquierda antes del movimiento.
p_m	Numero del píxel n asociado a un punto característico $\mathbf{h}'_{ln,t_{k+1}}$ en la imagen izquierda después del movimiento.
p_{n_q}	Cantidad de correspondencias típicas que se encontraron en la iteración número n_q del algoritmo de RANSAC.
$p_{v_{b1}}$	Numero del primer píxel de la ventana de búsqueda V_b .
$p_{v_{bf}}$	Numero del ultimo píxel de la ventana de búsqueda V_b .
$p_{v_{c1}}$	Numero del primer píxel de la ventana de correlación V_c .
$p_{v_{cf}}$	Numero del ultimo píxel de la ventana de correlación V_c .
$p_{v_{c1},pcn}$	Numero del primer píxel de cada una de las ventanas de correlación $V_{c,pcn}$.
$p_{v_{cf},pcn}$	Numero del ultimo píxel de cada una de las ventanas de correlación $V_{c,pcn}$.
pc_j	Numero del punto característico i asociado a un píxel pl_{j,t_k} en la imagen izquierda antes del movimiento.
pc_n	Numero del punto característico m asociado a un píxel $pl_{n,t_{k+1}}$ en la imagen izquierda después del movimiento.
\mathbf{pl}_j	Conjunto de píxeles que conforman la imagen izquierda de un arreglo estereoscópico, dado por el vector $(p_{cl_j}, p_{rl_j})^T$. También se pueden denotar como $\mathbf{pl}_j(c_{lj}, r_{lj})$.

$\mathbf{P}l_{j,t_k}$	Conjunto de píxeles, dados por el vector (c_{lj}, r_{lj}) , que conforman la imagen izquierda de un arreglo estereoscópico antes del movimiento del rover.
$\mathbf{P}l_{n,t_{k+1}}$	Conjunto de píxeles, dados por el vector (c_{ln}, r_{ln}) , que conforman la imagen izquierda de un arreglo estereoscópico después del movimiento del rover.
$\mathbf{P}lp_{v_{b1},t_{k+1}}$	Primer píxel de la ventana de búsqueda V_b , dado por el vector bidimensional $(c_{lp_{v_{b1}}}, r_{lp_{v_{b1}}})^T$.
$\mathbf{P}lp_{v_{c1},t_k}$	Primer píxel de la ventana de correlación V_c , dado por el vector bidimensional $(c_{lp_{v_{c1}}}, r_{lp_{v_{c1}}})^T$.
$\mathbf{P}lp_{v_{c1},pc_n,t_{k+1}}$	Primer píxel de cada una de las ventanas de correlación V_{c,pc_n} , dado por el vector bidimensional $(c_{lp_{v_{c1},pc_n}}, r_{lp_{v_{c1},pc_n}})^T$.
$\mathbf{P}lp_{v_{bf},t_{k+1}}$	Ultimo píxel de la ventana de correlación V_b , dado por el vector bidimensional $(c_{lp_{v_{bf}}}, r_{lp_{v_{bf}}})^T$.
$\mathbf{P}lp_{v_{cf},t_k}$	Ultimo píxel de la ventana de correlación V_c , dado por el vector bidimensional $(c_{lp_{v_{cf}}}, r_{lp_{v_{cf}}})^T$.
$\mathbf{P}lp_{v_{cf},pc_n,t_{k+1}}$	Ultimo píxel de cada una de las ventanas de correlación V_{c,pc_n} , dado por el vector bidimensional $(c_{lp_{v_{cf},pc_n}}, r_{lp_{v_{cf},pc_n}})^T$.
\mathbf{P}	Vector multidimensional que representa la pose de un objeto respecto a un sistema de referencia. Depende de las componentes de la posición y de los ángulos de orientación.
\mathbf{P}_s, t_k	Conjunto de puntos tridimensionales que se obtienen al sumar los puntos \mathbf{Q}_s, t_k con su centroide $\mu_{\mathbf{Q}_s, t_k}$.
\mathbf{P}_s, t_{k+1}	Conjunto de puntos tridimensionales que se obtiene al sumar los puntos \mathbf{Q}_s, t_{k+1} con su centroide $\mu_{\mathbf{Q}_s, t_{k+1}}$.
\mathbf{P}'	Vector multidimensional que representa la pose de un objeto después del movimiento, respecto a un sistema de referencia.
$\mathbf{q}l_{s,t_k}$	Subconjunto de puntos, del conjunto formado por los puntos característicos \mathbf{h}'_{li,t_k} antes del movimiento del rover, dados por el vector bidimensional $(c_{ls}, r_{ls})^T$.

$\mathbf{q}_{ls,t_{k+1}}$	Subconjunto de puntos del conjunto formado por los puntos característicos $\mathbf{h}'_{lm,t_{k+1}}$ después del movimiento del rover, dados por el vector bidimensional $(c_{ls}, r_{ls})^T$.
$\mathbf{q}'_{ls,t_{k+1}}$	Subconjunto de puntos que se obtienen al proyectar los puntos $\mathbf{Q}'_{ls,t_{k+1}}$ sobre la imagen izquierda después del movimiento, dados por el vector bidimensional $(c'_{ls}, r'_{ls})^T$.
\mathbf{Q}_{ls,t_k}	Subconjunto de puntos del conjunto formado por los puntos tridimensionales \mathbf{H}_{i,t_k} antes del movimiento del rover, dados por el vector $(X_{ls}, Y_{ls}, Z_{ls})^T$. Representan las posiciones tridimensionales de los puntos \mathbf{q}_{ls,t_k} .
$\mathbf{Q}_{ls,t_{k+1}}$	Subconjunto de puntos del conjunto formado por los puntos tridimensionales $\mathbf{H}_{m,t_{k+1}}$ antes del movimiento del rover, dados por el vector $(X_{ls}, Y_{ls}, Z_{ls})^T$. Representan las posiciones tridimensionales de los puntos $\mathbf{q}_{ls,t_{k+1}}$.
$\mathbf{Q}'_{ls,t_{k+1}}$	Subconjunto de puntos, que se obtienen al transformar los \mathbf{Q}_{ls,t_k} (rotarlos y trasladarlos), dados por el vector tridimensional $(X'_{ls}, Y'_{ls}, Z'_{ls})^T$.
r	Radio de las ruedas de un robot móvil.
r_c	Radio entre el origen del sistema de referencia y el centro del eje de un robot móvil.
r_l	Radio entre el origen del sistema de referencia y la rueda izquierda de un robot móvil.
r_r	Radio entre el origen del sistema de referencia y la rueda derecha de un robot móvil.
<i>RANSAC</i>	<i>Random Sample Consensus</i> (Consenso de Muestra Aleatoria).
<i>RGB</i>	Modelo de color basado en los colores primarios rojo (<i>Red</i>), verde (<i>Green</i>) y azul (<i>Blue</i>).
<i>ROS</i>	<i>Robot Operating System</i> (Sistema Operativo Robótico).
R	Matriz que describe la orientación de un objeto respecto a un sistema de referencia.
R'	Matriz que describe la orientación de un objeto después del movimiento, respecto a un sistema de referencia.

\mathbf{R}_r^l	Matriz que describe la orientación de la cámara derecha de un arreglo estereoscópico, respecto a la cámara izquierda.
s	S-ésimo punto característico \mathbf{q}_{l_s, t_k} o tridimensional Q_{l_s, t_k} antes del movimiento del rover, así como el s-ésimo punto característico $\mathbf{q}_{l_s, t_{k+1}}$ o tridimensional $Q_{l_s, t_{k+1}}$ después del movimiento.
<i>SLAM</i>	<i>Simultaneous Localization and Mapping</i> (Localización y Mapeo Simultáneos).
\mathbf{S}	Matriz diagonal que se utiliza para calcular la matriz de rotación $\Delta \mathbf{R}'$.
t_k	Instante de tiempo antes del movimiento de un objeto (puede ser un rover o astromóvil).
t_{k+1}	Instante de tiempo después del movimiento de un objeto (puede ser un rover o astromóvil).
U	Componente de crominancia o de proyección azul.
\mathbf{U}	Matriz izquierda de la descomposición en valores singulares de la matriz de covarianza $\Sigma_{Q_{t_k - t_{k+1}}}$.
V_b	Ventana de búsqueda en la imagen izquierda después del movimiento del rover, con centro en (c_{la_q}, r_{la_q}) .
V_c	Ventana de correlación en la imagen izquierda antes del movimiento del rover, con centro en el punto \mathbf{h}'_{la_q, t_k} .
V_{c, pc_n}	Ventana de correlación en la imagen izquierda después del movimiento del rover, con centro en el punto $\mathbf{h}'_{lpc_n, t_{k+1}}$.
$V_{c, pc_{nmax}}$	Ventana de correlación con el coeficiente de similitud máximo, en la imagen izquierda después del movimiento del rover, con centro en el punto $\mathbf{h}'_{lpc_{nmax}, t_{k+1}}$.
\mathbf{V}	Matriz derecha de la descomposición en valores singulares de la matriz de covarianza $\Sigma_{Q_{t_k - t_{k+1}}}$.
\mathbf{V}_{cpn}	Vector de coeficientes de similitud asociados a cada una de las ventanas de correlación V_{c, pc_n} .
w	Ancho de una imagen capturada por una cámara monocular o estereoscópica.

w_b	Ancho de una ventana de búsqueda V_b en la imagen izquierda después del movimiento.
w_c	Ancho de una ventana de correlación V_c o V_{c,pc_n} en la imagen izquierda antes o después del movimiento, respectivamente.
w_s	Factor de peso escalar que refleja la calidad de los pares de puntos tridimensionales \mathbf{Q}_{ls,t_k} y $\mathbf{Q}_{ls,t_{k+1}}$.
W	Vector dado por $(W_R, W_S, W_T)^T$ que describe la velocidad angular de una unidad de medición inercial, respecto a su sistema local.
YUV	Modelo de color basado en una componente de luminancia monocroma o de intensidad (Y) y dos componentes de crominancia (U,V).
$2D$	Bidimensional.
$3D$	Tridimensional.
$(c, r)^T$	Sistema de coordenadas de una imagen de una cámara monocular.
$(c_l, r_l)^T$	Sistema de coordenadas de la imagen izquierda de un arreglo estereoscópico.
$(c'_l, r'_l)^T$	Vector que se forma entre el centro de la ventana de correlación V_c y el centro de la ventana de V_{c,pc_n} en una imagen ficticia.
$(c_r, r_r)^T$	Sistema de coordenadas de la imagen derecha de un arreglo estereoscópico.
$(R, S, T)^T$	Sistema de coordenadas local de una unidad de medición inercial.
$(u, v)^T$	Sistema de coordenadas del plano de una cámara.
$(u_l, v_l)^T$	Sistema de coordenadas del plano de la cámara izquierda de un arreglo estereoscópico.
$(u_r, v_r)^T$	Sistema de coordenadas del plano de la cámara derecha de un arreglo estereoscópico.
$(U, V, W)^T$	Sistema de coordenadas local de un objeto.

$(X, Y, Z)^T$	Sistema de coordenadas de referencia.
$(X_l, Y_l, Z_l)^T$	Sistema de la cámara izquierda de un arreglo estereoscópico.
$(X_r, Y_r, Z_r)^T$	Sistema de la cámara derecha de un arreglo estereoscópico.
$[c]$	Cámara.
$[img]$	Imagen.
$[img_l]$	Imagen izquierda de un arreglo estereoscópico.
$[img_{l,t_k}]$	Imagen izquierda de un arreglo estereoscópico antes del movimiento del rover.
$[img_r]$	Imagen derecha de un arreglo estereoscópico.
$[img_{r,t_{k+1}}]$	Imagen derecha de un arreglo estereoscópico después del movimiento del rover.
$[O]$	Objeto.
$[O_{t_k}]$	Objeto antes del movimiento.
$[O_{t_{k+1}}]$	Objeto después del movimiento.
$[pc]$	Plano de una cámara monocular.
$[pc_l]$	Plano de la cámara izquierda de un arreglo estereoscópico.
$[pc_r]$	Plano de la cámara derecha de un arreglo estereoscópico.
$[ref]$	Sistema de referencia.
π	Plano epipolar de un arreglo estereoscópico.
ϕ	Ángulo formado por la distancia recorrida por las ruedas de un robot móvil.
α	Orientación de un robot móvil antes del movimiento.
α'	Orientación de un robot móvil después del movimiento.
μ_{v_c}	Media de los valores de intensidad de los píxeles dentro de la ventana de correlación V_c .

μ_{v_c, pc_n}	Media de los valores de intensidad de los píxeles dentro cada una de las ventanas de correlación V_{c, pc_n} .
$\mu_{\mathbf{Q}_{s, t_k}}$	Centroide del conjunto formado por los puntos tridimensionales \mathbf{Q}_{ls, t_k} .
$\mu_{\mathbf{Q}_{s, t_{k+1}}}$	Centroide del conjunto formado por los puntos tridimensionales $\mathbf{Q}_{ls, t_{k+1}}$.
$\mu_{\mathbf{Q}'_{s, t_{k+1}}}$	Centroide del conjunto formado por los puntos tridimensionales $\mathbf{Q}'_{ls, t_{k+1}}$.
$\sigma_{v_c}^2$	Varianza de los valores de intensidad de los píxeles dentro de la ventana de correlación V_c .
σ_{v_c, pc_n}^2	Varianza de los valores de intensidad de los píxeles dentro de cada una de las ventanas de correlación V_{c, pc_n} .
$\Sigma_{\mathbf{Q}_{s, t_k}}$	Matriz de covarianza del vector de ruido que afecta a los puntos tridimensionales \mathbf{Q}_{ls, t_k} .
$\Sigma_{\mathbf{Q}_{s, t_{k+1}}}$	Matriz de covarianza del vector de ruido que afecta a los puntos tridimensionales $\mathbf{Q}_{ls, t_{k+1}}$.
$\Sigma_{\mathbf{Q}_{t_k - t_{k+1}}}$	Matriz de covarianza de los conjuntos formados por los puntos tridimensionales \mathbf{Q}_{ls, t_k} y $\mathbf{Q}_{ls, t_{k+1}}$.
δ	Ángulo de rotación de una rueda.
δ_l	Ángulo de rotación de la rueda izquierda de un robot móvil.
δ_r	Ángulo de rotación de la rueda derecha de un robot móvil.
$\Delta \mathbf{R}$	Matriz que describe la rotación o el cambio de orientación de un objeto (puede ser un rover) respecto a un sistema de referencia.
$\Delta \mathbf{R}'$	Matriz que describe la rotación de un rover y que se obtiene al minimizar el error cuadrático medio e^2 .
$\Delta \mathbf{R}'_{pn_q, max}$	Matriz que describe la rotación de un rover con mayor exactitud y que se obtiene a partir del algoritmo de RAN-SAC.

$\Delta \mathbf{T}$	Vector dado por $(\Delta T_X, \Delta T_Y, \Delta T_Z)^T$ que describe la traslación tridimensional de un objeto (puede ser un rover) respecto a un sistema de referencia.
$\Delta \mathbf{T}'$	Vector que describe la traslación de un rover y que se obtiene al minimizar el error cuadrático medio e^2 .
$\Delta \mathbf{T}'_{pq,max}$	Vector que describe la traslación de un rover con mayor exactitud y que se obtiene a partir del algoritmo de RAN-SAC.
$\Delta \mathbf{W}$	Vector dado por $(\Delta W_X, \Delta W_Y, \Delta W_Z)^T$ que describe los ángulos de rotación de un objeto respecto a un sistema de referencia.

Resumen

En el presente trabajo se implementó y se probó la primera parte del algoritmo de odometría visual estereoscópica propuesto por Larry Matthies en 1989, sin considerar el error de posición en los puntos característicos, el cual proporciona una estimación inicial del movimiento que experimenta un rover entre dos instantes de tiempo consecutivos, a partir de la imagen izquierda de los pares de imágenes capturados en ambos instantes por una cámara estereoscópica atada rigidamente al rover y de las nubes de puntos tridimensionales que se obtienen a partir de la triangulación de correspondencias estereoscópicas entre los pares de imágenes para cada instante, las cuales son proporcionadas a través de un algoritmo externo basado en ROS. Específicamente, el algoritmo encuentra aquella traslación y rotación tridimensional con la que se tiene que mover un subconjunto de la nube de puntos antes del movimiento, elegido aleatoriamente, para que este coincida con un subconjunto de la nube de puntos después del movimiento, donde la coincidencia se logra minimizando la media de los cuadrados de las distancias de correspondencias temporales clasificadas como típicas entre el primer subconjunto transformado (rotado y trasladado) y el segundo subconjunto, a partir de un método basado en la descomposición en valores singulares de una matriz 3×3 . Además, las correspondencias temporales se establecen de acuerdo a un método basado en la correlación pseudo normalizada y la clasificación de estas en típicas y atípicas se lleva a cabo mediante la implementación de un proceso iterativo conocido como RANSAC. Dicho algoritmo se implementó en el lenguaje de programación C, haciendo uso de la plataforma de programación conocida como ROS-Indigo, bajo el sistema operativo Ubuntu 14.04 LTS. Para probar el algoritmo implementado se llevaron a cabo un total de tres pruebas que utilizaron como entrada una imagen de intensidad izquierda y una nube de puntos que se obtuvieron en un entorno controlado del mundo real y que se consideraron como los datos de entrada de prueba antes del movimiento. Las pruebas sometieron a los distintos procesos del algoritmo a distintas circunstancias y situaciones generadas de forma controlada y sintética, gracias a que la imagen de intensidad izquierda y la nube de puntos después del movimiento, se obtuvieron manipulando los datos de entrada de prueba antes del movimiento, mediante transformaciones y ecuaciones matemáticas conocidas. Los resultados de las pruebas demostraron que el algoritmo implementado y por lo tanto, los distintos procesos que lo componen, tienen un funcionamiento esperado, es decir, que cumplen con las expectativas y que no están por debajo ni por encima de estas y además, que tienen una buena precisión y exactitud bajo ciertas condiciones.

1 Introducción

1.1. Introducción al tema

En los últimos 20 años, los rovers (vehículos de exploración espacial, también conocidos como astromóviles), como el *Spirit* y *Opportunity* (Arvidson et al., 2011; Squyres et al., 2004) de la Misión *Mars Exploration Rover* (MER) y el *Curiosity* (Vasavada et al., 2014; Grotzinger et al., 2012) de la Misión *Mars Science Laboratory* (MSL), ambas de la NASA, han probado ser instrumentos considerablemente duraderos para la exploración de Marte, debido a su gran capacidad para navegar y efectuar actividades de manera semiautónoma (Maimone et al., 2006; Martínez, 2019a, 2018b,a, 2017; Martinez, 2014; Martínez, 2013), así como para sobrevivir más allá de cualquier pronóstico (Townsend et al., 2014), lo que les ha facilitado la posibilidad de observar más de cerca cualquier objeto interesante e importante que se encuentre en su trayecto y ensanchar mucho más el territorio explorado (Biesiadecki et al., 2005; Leger et al., 2005). Las tareas que debe de efectuar el rover durante el día generalmente se transmiten solo una vez por día marciano¹ a través de una cadena preprogramada de comandos, que los ingenieros y científicos en la base de la Tierra envían cada mañana (Mishkin et al., 2006). Se espera que el rover navegue de forma precisa y segura a lo largo de un trayecto establecido, se posicione respecto a un objetivo determinado, despliegue sus distintos instrumentos para recopilar y extraer datos científicos valiosos y los devuelva de forma segura a la Tierra (Maimone et al., 2006), donde cualquier error en la navegación podría ser fatal e implicar la pérdida de un día entero de exploración, atrapar al rover en un terreno resbaladizo y peligroso o estropear el hardware (Biesiadecki et al., 2005; Leger et al., 2005). Los datos que se reciben en la base terrestre se utilizarán para la investigación, inspección y observación científica y para planear las tareas que se realizarán en el próximo sol (Mishkin et al., 2006).

Para una navegación autónoma, segura y precisa, el rover tiene que conocer su posición y orientación exactas en el momento de la ejecución de los comandos de movimiento (Maimone et al., 2007). La posición del rover se obtiene integrando su traslación a lo largo del tiempo, la cual se estima a partir de lecturas de la distancia recorrida por sus ruedas mediante un método conocido como odometría de ruedas (Ali et al., 2005). La orientación también se obtiene integrando la rotación del rover a lo largo del tiempo, la cual se obtiene

¹A un día marciano también se le conoce como sol y es aproximadamente 40 minutos más largo que un día en la Tierra

directamente de los giroscopios de la unidad de medición inercial (IMU, del inglés *Inertial Measurement Unit*) que se encuentra a bordo del rover (Ali et al., 2005). Al principio del movimiento del rover, se supone que la posición inicial es conocida y la orientación inicial se estima a partir de las mediciones proporcionadas por la IMU y la ubicación del sol, la cual se obtiene mediante un sensor solar (Eisenman et al., 2002).

El uso de odometría de ruedas tiene varios inconvenientes, el más común es que el error acumulado con la distancia recorrida depende directamente del tipo y la geometría del terreno sobre el cual el astromóvil ha estado navegando, siendo despreciable en terrenos nivelados con alta fricción (Leger et al., 2005; Ali et al., 2005; Li et al., 2006), donde el deslizamiento de las ruedas es mínimo a causa de la buena tracción, y grande en pendientes pronunciadas y en terrenos arenosos, pantanosos y pedregosos (Biesiadecki et al., 2005; Maimone et al., 2007; Lindemann y Voorhees, 2005), donde las ruedas deslizan considerablemente producto de la pérdida de tracción o cuando se topan con una roca (Li et al., 2006). Esto llega a limitar bastante la navegación autónoma de los astromóviles en escenarios resbaladizos (Helmick et al., 2005), porque la determinación de la posición utilizando odometría de ruedas no sería lo suficientemente precisa para compensar el deslizamiento, a fin de garantizar que el rover se mueva por el trayecto deseado (Maimone et al., 2007). Además, un deslizamiento excesivo de las ruedas podría generar problemas graves como el atascamiento del astromóvil en terrenos blandos (Maimone et al., 2007; Li et al., 2008).

Para mejorar la precisión y seguridad de la navegación autónoma de los astromóviles en terrenos resbaladizos, después de moverse una pequeña distancia, se le ordena al rover, que corrija cualquier error de pose producto del deslizamiento de las ruedas, utilizando para ello un segundo estimado de su pose, el cual se obtiene mediante un algoritmo de odometría visual estereoscópico basado en puntos característicos (*feature based stereo visual odometry algorithm*, en inglés) (Maimone et al., 2006, 2007; Biesiadecki et al., 2007). Dicho algoritmo es capaz de determinar la posición y orientación del astromóvil a partir de la estimación de su movimiento, mediante el rastreo de puntos característicos a lo largo de una secuencia de pares de imágenes capturados por una cámara estereoscópica montada en el rover. Una cámara estereoscópica es un arreglo calibrado de dos cámaras monoculares. Las imágenes capturadas por una cámara estereoscópica en un instante de tiempo dado son denominadas frecuentemente imagen derecha e imagen izquierda.

El algoritmo de odometría visual estereoscópico mencionado anteriormente fue propuesto por Moravec (1980) y mejorado posteriormente por Matthies (1989). El algoritmo de Larry Matthies tiene dos partes: parte a) de estimación inicial y parte b) de estimación precisa de movimiento.

En la parte a), se hace una estimación inicial de los parámetros de mo-

vimiento del astromóvil, y consiste en cuatro pasos. Primero, antes de que el rover se mueva, se captura un par de imágenes con la cámara estereoscópica. Posteriormente, un conjunto de puntos característicos bidimensionales (2D) se eligen cuidadosamente y de manera uniforme en la imagen izquierda (Maimone et al., 2007; Moravec, 1980). Un punto se considera como un punto característico si este se puede encontrar sin ambigüedades en ambas imágenes capturadas por una cámara estereoscópica, donde los ubicados en esquinas, con alto contraste en direcciones ortogonales, son los mejores (Shi y Tomasi, 1994). Luego, en el segundo paso, se calculan las posiciones tridimensionales (3D) de los puntos característicos 2D seleccionados. Para ello, se establecen las correspondencias estereoscópicas entre el conjunto de puntos característicos que se seleccionaron en la imagen izquierda y el conjunto de puntos o píxeles que conforman la imagen derecha. Un punto en la imagen derecha corresponderá a un punto característico seleccionado en la imagen izquierda, siempre y cuando cumpla con la restricción epipolar (Hartley y Zisserman, 2004), así como su valor de intensidad y los valores de intensidad vecinos sean los más similares a los valores de intensidad del punto característico y sus vecinos en la imagen izquierda, donde la similitud se mide haciendo uso de la correlación pseudo normalizada (Moravec, 1980; Hubbard, 2007). De las correspondencias estereoscópicas establecidas, se determinan las posiciones tridimensionales de los puntos característicos, haciendo uso de triangulación estereoscópica (Hartley y Zisserman, 2004). A dicho conjunto de posiciones 3D se le denomina frecuentemente como la nube de puntos 3D antes del movimiento. En el tercer paso, después de que el rover se mueva una pequeña distancia, se calcula una segunda nube de puntos 3D, conocida como nube de puntos 3D después del movimiento. Para ello, se captura un segundo par de imágenes. Luego, la nube de puntos 3D antes del movimiento, se mueve con el estimado de la traslación proporcionada por la odometría de ruedas (Maimone et al., 2007; Moravec, 1980). Posteriormente, sus puntos 3D se proyectan en la imagen izquierda del segundo par de imágenes y las posiciones de las proyecciones son seguidamente refinadas haciendo uso nuevamente de la correlación pseudo normalizada (Moravec, 1980; Hubbard, 2007). Luego, se establecen las correspondencias estereoscópicas entre las posiciones refinadas determinadas anteriormente y los puntos en la imagen derecha del segundo par de imágenes, haciendo uso nuevamente de la correlación pseudo normalizada como una medida de la similitud. De las correspondencias estereoscópicas establecidas entre la imagen izquierda y la imagen derecha del segundo par de imágenes, se determinan las posiciones 3D de los puntos que conforma la segunda nube de puntos, haciendo uso de triangulación estereoscópica (Hartley y Zisserman, 2004). Además, como la segunda nube de puntos se obtuvo a partir de la proyección refinada de la primera nube de puntos en la imagen izquierda después del movimiento, implica que se realizó indirectamente un establecimiento de correspondencias

temporales 2D entre las proyecciones refinadas de la primera nube de puntos en la imagen izquierda después del movimiento y los puntos característicos en la imagen izquierda después del movimiento. También implica que se realizó indirectamente un establecimiento de correspondencias temporales 3D entre la primera nube de puntos y la segunda nube de puntos, ya que estas son las posiciones tridimensionales de las correspondencias temporales 2D. Una correspondencia temporal se da cuando un punto en un instante de tiempo antes del movimiento corresponde a un punto en un instante de tiempo después del movimiento. En el cuarto paso, se procede a estimar el movimiento del astromóvil, encontrando aquella traslación y rotación con la que se debe mover la primera nube de puntos 3D para que ésta coincida con la segunda nube de puntos 3D, donde la coincidencia se logra minimizando la suma de los cuadrados sopesados de las distancias de correspondencias temporales 3D clasificadas como típicas entre un subconjunto de la primera nube de puntos transformada (traslada y rotada) y un subconjunto de la segunda nube de puntos, haciendo uso del método de descomposición en valores singulares (Umeyama, 1991; Arun et al., 1987, 2007), donde los pesos asociados a cada correspondencia se derivan de los determinantes de las matrices de covarianza de los errores de posición de los puntos tridimensionales, que a su vez se derivan de las matrices de covarianza de los errores de posición de los respectivos puntos característicos (Maimone et al., 2007; Arun et al., 2007; Matthies, 1989; Matthies y Shafer, 1987), mientras que las correspondencias típicas se encuentran haciendo uso de un algoritmo conocido en la literatura científica como RANSAC (Fischler y Bolles, 1981).

En la parte b) del algoritmo de Larry Matthies (Matthies, 1989; Matthies y Shafer, 1987), se mejora la precisión de los parámetros de movimiento inicialmente estimados en la parte a). Para ello, se maximiza una función de probabilidad condicional alrededor de los parámetros de movimiento inicialmente estimados en la parte a), la cual depende del error residual entre correspondencias temporales 3D clasificadas como típicas y de las matrices de covarianza de dichos errores residuales, las cuales se obtienen a partir de las matrices de covarianza de los errores de posición de los puntos tridimensionales (Maimone et al., 2007; Arun et al., 2007; Matthies, 1989; Matthies y Shafer, 1987).

El algoritmo de Larry Matthies (Matthies, 1989; Matthies y Shafer, 1987), fue implementado en tiempo real e incorporado en los rovers *Spirit* y *Opportunity* de la Misión MER (Maimone et al., 2007). Después de evaluar su desempeño en los rovers *Spirit* y *Opportunity* en Marte, se le realizaron cambios en (Johnson et al., 2008) para mejorar su robustez y reducir el tiempo de procesamiento. Esta última versión del algoritmo de odometría visual estereoscópico está siendo utilizada actualmente por el astromóvil *Curiosity* (Heverly et al., 2013; Maimone, 2013) en Marte. Una implementación aún más rápida, con la ayuda de un acelerador de hardware basado en FPGA, fue incorporada

en el nuevo astromóvil de exploración marciana de la NASA, conocido como *Perseverance*, el cual fue enviado a Marte en julio del 2020 (Howard et al., 2013).

En la literatura profesional se puede encontrar otros algoritmos similares, como los descritos en (Scaramuzza, 2011; Howard, 2008; Comport et al., 2007; Nister et al., 2006; Lacroix et al., 1999; Zhang et al., 1998), algunos de los cuales han sido adaptados para operar con una cámara monocular (Nister et al., 2006) o una cámara omnidireccional (Stuerzl et al., 2010; Corke et al., 2004; Vassallo et al., 2002; Gluckman y Nayar, 1998). En (Aqel et al., 2016; Yousif et al., 2015; Fraubdirfer y Scaramuzza, 2012; Scaramuzza y Fraundorfer, 2011) se pueden encontrar tutoriales útiles sobre odometría visual. En los últimos años, una rama evolutiva de la odometría visual, llamada SLAM (del inglés *Simultaneous Localization and Mapping*), ha empezado a emerger con fuerza en la comunidad científica (Strasdat et al., 2010a,b).

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, de la Universidad de Costa Rica, se propuso un algoritmo de odometría visual monocular basado en diferencias de intensidad, como una alternativa a los algoritmos de odometría visual estereoscópica (Martínez, 2019a, 2018b,a, 2017; Martinez, 2014; Martínez, 2013), que evita tener que establecer correspondencias estereoscópicas y temporales entre puntos característicos y sus posiciones tridimensionales, para la estimación de movimiento, tareas que se sabe que son muy difíciles, consumen mucho tiempo de procesamiento (Scaramuzza, 2011) y son propensas a errores debido a grandes movimientos, oclusiones o ambigüedades, que afectan en gran medida la estimación del movimiento 3D (Olson et al., 2003). Con este algoritmo es posible estimar el movimiento 3D del astromóvil mediante la maximización de la función de probabilidad condicional de las diferencias de intensidad medidas en puntos de observación entre dos imágenes sucesivas. Las imágenes son tomadas por una sola cámara de video antes y después del movimiento del astromóvil. Los puntos de observación son puntos de la imagen cuyos gradientes de intensidad lineal son altos. En (Engel et al., 2018), se ofrece otra alternativa que también evalúa diferencias de intensidad, no para estimar directamente movimiento, sino para estimar las posiciones y orientaciones del astromóvil sobre una ventana de imágenes consecutivas y recientes, junto con los valores intrínsecos de la cámara y los valores inversos de la profundidad correspondientes, haciendo uso de un algoritmo de ajuste en bloques, conocido en inglés como *bundle adjustment* (Triggs et al., 1999).

1.2. Justificación y Propuesta

A pesar de que en (Martinez, 2014; Martínez, 2013), el algoritmo de odometría visual monocular del IPCV-LAB fue probado exhaustivamente con datos sintéticos, así como en (Martínez, 2019a, 2018b) y en (Martínez, 2018a, 2017) fue probado en un astromóvil real en interiores y exteriores, respectivamente, todavía hace falta realizar experimentos que permitan comparar su desempeño con el desempeño del algoritmo de odometría visual estereoscópico de Larry Matthies (Matthies, 1989; Matthies y Shafer, 1987), con el fin de definir las fortalezas y debilidades de ambos algoritmos bajo las mismas condiciones experimentales.

Para lograr la meta anterior, y debido a su complejidad, ésta se subdividió en diferentes etapas, las cuales serán llevadas a cabo por diferentes miembros del IPCV-LAB, mediante trabajos finales de graduación y proyectos de investigación. En una primer etapa, se implementará y probará la parte a) del algoritmo de odometría visual estereoscópico de Larry Matthies (Matthies, 1989; Matthies y Shafer, 1987), sin considerar el error de posición en los puntos característicos, que se refiere a la estimación inicial del movimiento del astromóvil, minimizando la media de los cuadrados de las distancias de correspondencias temporales 3D clasificadas como típicas entre un subconjunto de la primera nube de puntos transformada (traslada y rotada) y un subconjunto de la segunda nube de puntos, haciendo uso del método de descomposición en valores singulares, que equivale a minimizar la suma de los cuadrados sopesados de dichas distancias, pero considerando los pesos como iguales y constantes. Luego, en una segunda etapa, se modelará el error de posición de los puntos característicos y se incorporará en el algoritmo de la primera etapa. Posteriormente, en una tercera etapa, se implementará y probará la parte b) del algoritmo de Larry Matthies, que se refiere a la estimación precisa del movimiento del astromóvil, maximizando la función de probabilidad condicional de las distancias de correspondencias temporales 3D clasificadas como típicas, alrededor los parámetros de movimiento inicialmente estimados por el algoritmo concluido en la segunda etapa. Finalmente, una vez concluida la implementación del algoritmo de odometría visual estereoscópico de Larry Matthies (Matthies, 1989; Matthies y Shafer, 1987), en una cuarta etapa, se hará la experimentación en interiores y exteriores, y realizará la evaluación del desempeño de dicho algoritmo y el desempeño del algoritmo de odometría visual monocular del IPCV-LAB, ambos bajo las mismas condiciones experimentales.

En este Proyecto Final de Graduación de Licenciatura, se llevó a cabo únicamente la primera etapa mencionada anteriormente, cuyo objetivo general es la implementación y prueba de la versión de la parte a) del algoritmo de odometría visual estereoscópico de Larry Matthies (Matthies, 1989; Matthies

y Shafer, 1987), en la cual no se considera el error de posición que afecta a los puntos característicos.

1.3. Metodología

La ejecución del proyecto se dividió en seis fases, las cuales se describen a continuación.

I Fase: Búsqueda bibliográfica, propuesta y justificación

Primeramente, se realizó un estudio de la evolución de los algoritmos de odometría visual, basado principalmente en el artículo de Martínez (2018a), que sirvió de introducción al tema y para definir el estado del arte de la odometría visual, así como para formular y justificar este Proyecto Final de Graduación de Licenciatura.

Posteriormente, se realizó un estudio de las bases de la odometría visual estereoscópica, como los conceptos, métodos y modelos básicos en visión por computador y en estereoscopía, que son utilizados en cualquier algoritmo de odometría visual estereoscópico y que sirvió para el desarrollo teórico de este proyecto.

Por último, se estudió con más detalle el algoritmo de odometría visual estereoscópico propuesto por Larry Matthies (Matthies, 1989; Matthies y Shafer, 1987), específicamente su parte a), que involucra la necesidad de estudiar con profundidad la correlación pseudo normalizada para establecer correspondencias, el algoritmo de RANSAC para excluir correspondencias atípicas o para encontrar correspondencias típicas y el procedimiento basado en la descomposición en valores singulares para la minimización del error cuadrático medio de las distancias entre correspondencias típicas.

II Fase: Estudio de lenguaje de programación C y sistemas operativos

Se estudió el lenguaje de programación C, así como los sistemas operativos Ubuntu 14.04 LTS y *Robot Operating System* (ROS), en su versión Indigo.

III Fase: Implementación

La parte a) del algoritmo de odometría visual estereoscópico propuesto por Larry Matthies (Matthies, 1989; Matthies y Shafer, 1987), sin contemplar el error de posición de los puntos característicos, se implementó en ROS-Indigo, bajo el sistema operativo Ubuntu 14.04 LTS, mediante un único nodo llamado *motion_estimation_node*, escrito en el lenguaje de programación C, que es capaz de proporcionar un estimado inicial de los parámetros que describen

el movimiento de un astromóvil, entre dos instantes de tiempo consecutivos, minimizando la media de los cuadrados de las distancias de correspondencias temporales 3D clasificadas como típicas entre un subconjunto de la primera nube de puntos transformada (traslada y rotada) y un subconjunto de la segunda nube de puntos, haciendo uso del procedimiento basado en la de descomposición en valores singulares, donde las nubes de puntos son proporcionadas y recibidas por medio del tópico *points2* del paquete de ROS llamado *stereo_imag_proc*, el cual las genera por triangulación a partir de dos pares consecutivos de imágenes, capturados antes y después del movimiento, por una cámara estereoscópica Bumblebee modelo BB2-03S2M-60, de la marca Point Grey, la cual estará inclinada hacia abajo y atada rígidamente al costado izquierdo de una plataforma de desarrollo robótico real, modelo Husky A200, de la marca Clearpath Robotics; el establecimiento de las correspondencias entre las nubes antes y después del movimiento se realiza haciendo uso de la correlación pseudo normalizada, mientras que las correspondencias consideradas como típicas serán seleccionadas haciendo uso del algoritmo de RANSAC. Finalmente, los parámetros estimados se almacenan en un archivo de texto de la siguiente manera: primero el vector de traslación, seguido por la matriz de rotación.

IV Fase: Pruebas experimentales y análisis de resultados

Se probó el correcto funcionamiento del algoritmo implementado, a partir de datos sintéticos, utilizando parámetros de movimiento conocidos y se evaluó su precisión cuando se introduce ruido Gaussiano a las posiciones tridimensionales de la nube de puntos y a los píxeles de la imagen de intensidad izquierda, después del movimiento.

Posteriormente, se realizó una evaluación crítica de los resultados y de los datos obtenidos.

V Fase: Escritura del informe del proyecto y manual de usuario

Se escribió el informe del proyecto con las conclusiones de las pruebas y se distribuyó entre los miembros del comité asesor para su evaluación. Asimismo, se escribió un manual de usuario del algoritmo implementado para que miembros del IPCV-LAB puedan aprender a usarlo con facilidad.

VI Fase: Defensa del proyecto

Se hizo una presentación oral del proyecto ante el comité asesor del mismo, y posteriormente se respondieron todas las preguntas que los miembros del comité decidieron formular.

1.4. Objetivos

1.4.1 Objetivo general

Implementar y probar la parte a) del algoritmo de odometría visual estereoscópico propuesto por Larry Matthies (Matthies, 1989; Matthies y Shafer, 1987), excluyendo el error de posición presente en los puntos característicos.

1.4.2 Objetivos específicos

- Realizar una búsqueda bibliográfica, formular la propuesta y justificar la misma.
- Estudiar el lenguaje de programación C y los sistemas operativos Ubuntu 14.04 LTS y ROS-Indigo.
- Implementar en un nodo ROS llamado *motion_estimacion_node*, la parte a) del algoritmo de odometría visual estereoscópico propuesto por Larry Matthies (Matthies, 1989; Matthies y Shafer, 1987) sin considerar el error de posición, haciendo uso del lenguaje de programación C, bajo los sistemas operativos Ubuntu 14.04 LTS y ROS-Indigo.
- Realizar pruebas experimentales y analizar los resultados obtenidos.
- Escribir el informe del proyecto, así como el manual de usuario del algoritmo implementado.
- Realizar la defensa del proyecto.

2 Desarrollo teórico

2.1. Conceptos, métodos y modelos básicos en visión por computador

En visión por computador se desarrollan nuevos algoritmos capaces de estimar y deducir de forma automática la estructura y propiedades de objetos dentro de un mundo tridimensional y posiblemente dinámico (forma, color, posición, orientación, movimiento, entre otros), a partir del análisis de una o más imágenes digitales y bidimensionales de ese mundo, las cuales son capturadas por una o más cámaras colocadas alrededor de los mismos. Las personas también se pueden considerar como un objeto dentro del contexto anterior y en ese caso interesa desarrollar nuevos algoritmos que sean capaces de estimar la mímica y gestos de las personas (Martínez, 2019b).

La visión por computador no se centra solamente en la estructura y propiedades de los objetos, el medio que los rodea también es de interés y es por eso que se desarrollan algoritmos capaces de construir mapas del entorno que rodea a los objetos y localizar a los mismos dentro de los mapas, a partir de imágenes digitales capturadas por cámaras atadas rígidamente a estos. Dichos algoritmos son frecuentemente conocidos como algoritmos de localización y mapeo. Son especialmente útiles en robótica autónoma. La detección y el reconocimiento de objetos también son temas de estudio en la visión por computador (Martínez, 2019b).

El propósito de esta sección es introducir los conceptos, métodos y modelos básicos más utilizados por los algoritmos de visión por computador, como, por ejemplo, los algoritmos de odometría visual estereoscópica que se utilizan para estimar el movimiento de un robot.

2.1.1 Modelo de iluminación

Un sistema de iluminación bien diseñado ilumina la escena de tal forma que las imágenes que se capturen faciliten el posterior proceso sobre las mismas, manteniendo e incluso mejorando la información y los datos necesarios para la detección y extracción de los objetos y las propiedades de valor (Pajares y de la Escalera, 2016).

El éxito de un algoritmo de visión radica en la iluminación que se utilice. Si se utiliza una iluminación inadecuada pueden surgir varios problemas de contrastes, brillos y sombras que complican el algoritmo y en el peor de los

casos, ocasionan que este no pueda llegar a una solución válida. Para incrementar la eficiencia de un algoritmo de visión se debe de utilizar el modelo y la técnica de iluminación apropiada, que facilite la obtención de imágenes correctas para ser procesadas. Una imagen correcta es una imagen donde los píxeles que describen al objeto de interés tienen propiedades de iluminación similares y difieren de los demás píxeles que describen el resto de la escena. Además, una imagen correcta no presenta zonas saturadas o sombras que alteran la información y provocan falsas detecciones de bordes y esto solo es posible con una iluminación idónea. También, una mala iluminación puede llegar a generar ruido en los píxeles de la imagen (Pajares y de la Escalera, 2016).

Existen muchas técnicas y modelos de iluminación en visión por computador. La técnica de iluminación más utilizada es la difusa, que consiste en la iluminación de la escena de forma indirecta para conseguir una luz suave. Esta técnica reduce los brillos y las sombras, pero requiere de un sistema de iluminación de gran tamaño. Se utiliza un modelo de iluminación sin parámetros para describir dicha técnica, es el más simple de todos, pero el más utilizado. En dicho modelo se considera que la iluminación difusa es invariante en el espacio y en el tiempo. Esto quiere decir que cada punto que se encuentra sobre la superficie de un objeto es iluminado con la misma intensidad, sin importar el ángulo entre el vector de posición de la fuente de iluminación (\mathbf{L}_0) y el vector normal a la superficie del objeto (\mathbf{N}), como se aprecia en la figura 2.1 (Martínez, 2019b). Es entonces un modelo bastante sencillo que se basa en la ausencia de sombras y cambios temporales y locales de iluminación.

2.1.2 Modelo de color

El color es una característica muy importante que le permite a los seres humanos identificar y clasificar los objetos que los rodean. Es una percepción visual que se genera en el cerebro producto de las ondas electromagnéticas que se reflejan en los objetos y que luego son captadas por los ojos. Todo aspecto visual que se pueda cuantificar de un objeto es debido a la luminosidad y al color. Los bordes que limitan un objeto se derivan de la facultad del ojo para diferenciar entre zonas de luminosidad y color diferentes (Szeliski, 2011).

En el campo de visión por computador se utilizan los llamados espacios o modelos de color, que son un conjunto de fórmulas matemáticas que describen los colores mediante su descomposición en tres componentes (luminancias) para describirlos de forma cualitativa y cuantitativa (Pajares y de la Escalera, 2016). Los modelos de color más utilizados son el RGB y el YUV.

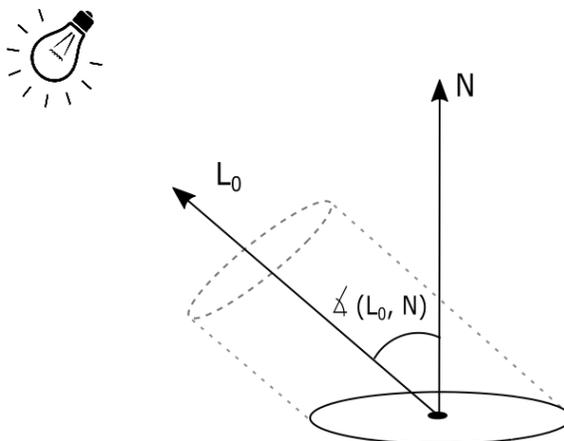


Figura 2.1: Modelo de iluminación difusa e invariante (Martínez, 2019b).

Modelo RGB

El modelo RGB consiste en la variación de las cantidades de los colores primarios (rojo, verde y azul) para obtener múltiples colores C mediante la suma aritmética (Martínez, 2019b):

$$C = R + G + B \quad (2.1)$$

El color resultante es trivariante debido a que los tres colores primarios son totalmente independientes entre sí, de tal forma que el color de la luz reflejada en un punto sobre la superficie de un objeto se puede describir usando las tres componentes RGB (también conocidas como componentes espectrales) que representan las cantidades adecuadamente dosificadas de cada uno de los tres colores primarios respectivos, que al mezclarse reproducen el color de la luz reflejada en ese punto (Martínez, 2019b).

Modelo YUV

Este modelo se basa en la obtención de la componente de luminancia (luminosidad o brillo) y las componentes de crominancia (color).

A partir de los colores primarios y del nivel blanco de referencia, se puede obtener la expresión fundamental para la componente de luminancia monocroma, también conocida como la componente de intensidad (Martínez, 2019b; Pajares y de la Escalera, 2016):

$$Y = 0,3R + 0,59G + 0,11B \quad (2.2)$$

donde R , G y B son las componentes espectrales de la cámara de vídeo, de un píxel de la imagen o de un punto de un objeto.

La ecuación 2.2 es muy utilizada en visión por computador para obtener la imagen de intensidad de una imagen RGB capturada por una cámara de vídeo tricolor (Martínez, 2019b). Como se observa en la figura 2.2, la imagen de intensidad se obtiene al aplicar dicha ecuación a cada uno de los píxeles de la imagen RGB , de tal forma que los nuevos píxeles representen el valor de la componente de intensidad o de luminancia monocroma, al cambiar el valor de las respectivas componentes RGB por el valor de la intensidad Y ($R = G = B = Y$). En múltiples algoritmos de visión se obtienen mejores resultados al procesar la imagen de intensidad en lugar de la imagen RGB .

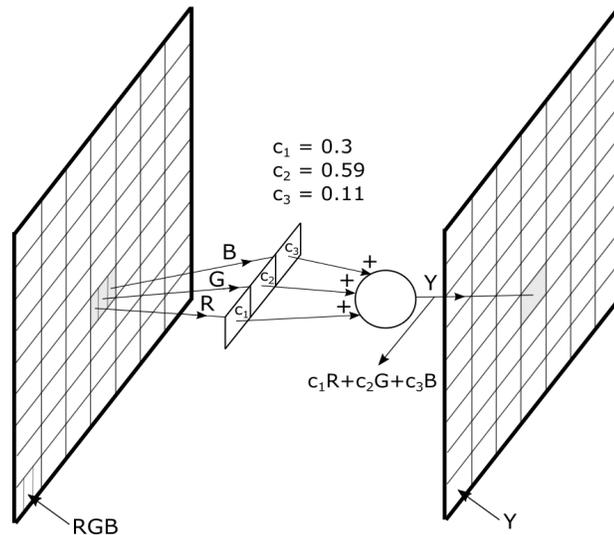


Figura 2.2: Cálculo de la imagen de intensidad a partir de una imagen RGB (Martínez, 2019b).

Respecto a las componentes de crominancia, se obtienen al restar las componentes RGB y la luminancia Y , para un total de tres componentes: $R - Y$, $G - Y$ y $B - Y$. Con la componente de luminancia se tiene un total de cuatro componentes, como el color es definido por el modelo RGB , solo tres son necesarias. Debido a que la componente $G - Y$ tiene una baja relación y menores coeficientes respecto a las otras, no se toma en cuenta en el modelo (Martínez, 2019b; Pajares y de la Escalera, 2016). Finalmente, las ecuaciones de la crominancia son las que aparecen en la ecuación 2.3:

$$\begin{aligned} U &= 0,493(B - Y) \\ V &= 0,877(R - Y) \end{aligned} \quad (2.3)$$

2.1.3 Modelo de posición de un objeto rígido

En visión por computador es sumamente importante el estudio de la cinemática de los objetos rígidos, porque constantemente se está tratando con la pose de los cuerpos en el espacio (posición y orientación) y su movimiento (traslación rotación). Los cuerpos de interés van desde el brazo de una persona hasta una cámara atada rígidamente al costado de un robot.

Para estudiar la cinemática de un objeto rígido, primero se establecen dos sistemas de coordenadas cartesianos espaciales, los cuales se representan mediante tres rectas perpendiculares entre sí, llamadas ejes, que se cortan en un mismo punto, llamado origen. El primero, es un sistema de coordenadas local que se une o se conecta al objeto de una manera conocida, es decir, su origen se ubica en un punto conocido del objeto (puede ser su centro de masa, una esquina, etc.) y se conoce la orientación de los ejes respecto a este. Dicho sistema se mueve junto con el objeto, pero manteniendo siempre el mismo origen, es decir, desde el punto de vista del objeto este sistema de coordenadas es fijo. El segundo, es un sistema de coordenadas de referencia que se ubica en cualquier lugar del espacio que rodea al objeto, pero con un origen y una orientación de los ejes conocidas. Dicho sistema es fijo en el espacio (Tsai, 1999).

Entonces, el sistema de coordenadas de referencia se puede representar con las rectas $(X, Y, Z)^T$ y de igual manera, el sistema de coordenadas local del objeto se puede representar con otras tres rectas $(U, V, W)^T$, tal y como se aprecia en la figura 2.3.

La posición del objeto con respecto al sistema de coordenadas de referencia se representa con el vector tridimensional de posición \mathbf{G} , compuesto por las coordenadas $(G_X, G_Y, G_Z)^T$ (Martínez, 2019b):

$$\mathbf{G} = \begin{pmatrix} G_X \\ G_Y \\ G_Z \end{pmatrix} \quad (2.4)$$

Este vector se extiende desde el origen del sistema de coordenadas de referencia, hasta otro punto extremo que corresponde al origen del sistema de coordenadas local del objeto, como se observa en la figura 2.3.

Por otro lado, en la figura 2.3 se tiene una posición definida por el vector \mathbf{H}^* , que representa la posición de un punto del objeto respecto a su sistema de coordenadas local. Se quiere conocer la posición de dicho punto, pero con

respecto al sistema de coordenadas de referencia, cuando este sistema se encuentra alineado con el sistema de coordenadas local, es decir, tienen la misma orientación y están separados solamente por el vector de posición \mathbf{G} . Entonces, como los vectores \mathbf{H}^* y \mathbf{G} están definidos sobre sistemas con la misma orientación, se puede calcular la posición del punto del objeto respecto a la referencia mediante una simple suma vectorial (Martínez, 2019b; Craig, 1999):

$$\mathbf{H} = \mathbf{H}^* + \mathbf{G} \quad (2.5)$$

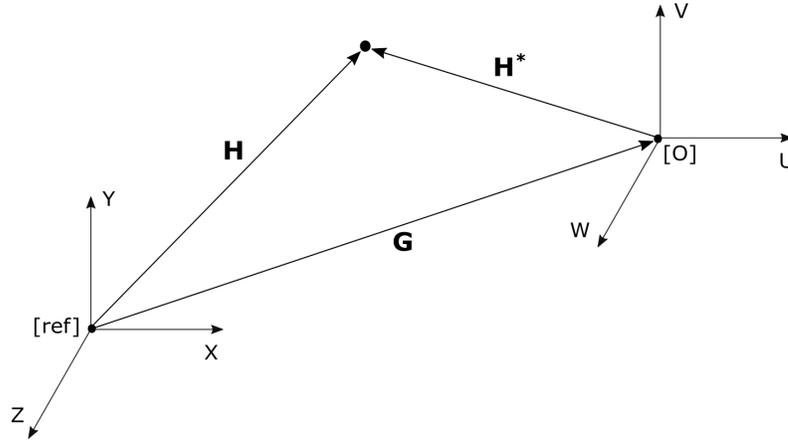


Figura 2.3: Posición de cualquier punto de un objeto cuando ambos sistemas de coordenadas están alineados (Martínez, 2019b).

2.1.4 Modelo de orientación de un objeto rígido

Frecuentemente, es necesario no solo representar la posición de un objeto en el espacio respecto a un sistema de referencia, sino también describir su orientación en el espacio.

En la figura 2.4, se muestra el método más usado para describir la orientación de un objeto. Se comienza con el sistema de coordenadas local $(U, V, W)^T$ coincidente con el sistema de coordenadas de referencia $(X, Y, Z)^T$, es decir, comparten el mismo origen. Primero se gira el sistema local alrededor del eje X en un ángulo \mathfrak{R}_X , luego alrededor del eje Y en un ángulo \mathfrak{R}_Y y finalmente, alrededor del eje Z en un ángulo \mathfrak{R}_Z . Cada uno de los tres giros se dan alrededor de un eje del sistema de referencia fijo. Esta forma de describir la orientación se conoce como ángulos fijos X-Y-Z. La palabra "fijo" se refiere a que la orientación se especifica sobre el sistema de referencia (Martínez, 2019b; Craig, 1999).

Para representar la orientación de un objeto, descrita por el método de los ángulos fijos X-Y-Z, se utiliza una matriz 3x3 que se denota como \mathbf{R} y se le conoce como la matriz de orientación (Martínez, 2019b):

$$\mathbf{R} = \begin{pmatrix} r_1 & r_2 & r_3 \\ r_4 & r_5 & r_6 \\ r_7 & r_8 & r_9 \end{pmatrix} \quad (2.6)$$

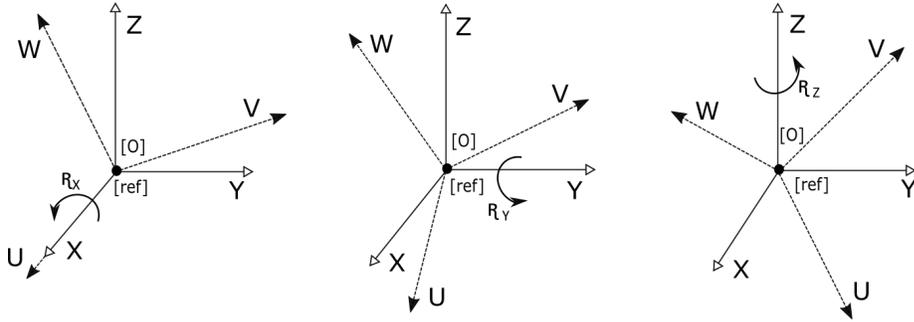


Figura 2.4: Método de los ángulos fijos X-Y-Z para describir la orientación de un objeto respecto a una referencia (Martínez, 2019b).

Cada una de las columnas de la matriz de orientación de la expresión 2.6 representan los vectores unitarios de los ejes principales del sistema de coordenadas local, respecto al sistema de referencia. Además, las entradas de dicha matriz se pueden obtener con las ecuaciones de la 2.8 a la 2.16, las cuales son función de las entradas del vector de orientación \mathbf{O} , ecuación 2.7, que corresponden a los ángulos de orientación respecto a cada uno de los ejes del sistema de coordenadas de referencia (Martínez, 2019b).

$$\mathbf{O} = \begin{pmatrix} \mathfrak{R}_X \\ \mathfrak{R}_Y \\ \mathfrak{R}_Z \end{pmatrix} \quad (2.7)$$

$$r_1 = \cos(\mathfrak{R}_Y) \cdot \cos(\mathfrak{R}_Z) \quad (2.8)$$

$$r_2 = \sin(\mathfrak{R}_X) \cdot \sin(\mathfrak{R}_Y) \cdot \cos(\mathfrak{R}_Z) - \cos(\mathfrak{R}_X) \cdot \sin(\mathfrak{R}_Y) \quad (2.9)$$

$$r_3 = \cos(\mathfrak{R}_X) \cdot \sin(\mathfrak{R}_Y) \cdot \cos(\mathfrak{R}_Z) + \sin(\mathfrak{R}_X) \cdot \sin(\mathfrak{R}_Z) \quad (2.10)$$

$$r_4 = \cos(\mathfrak{R}_Y) \cdot \sin(\mathfrak{R}_Z) \quad (2.11)$$

$$r_5 = \sin(\mathfrak{R}_X) \cdot \sin(\mathfrak{R}_Y) \cdot \sin(\mathfrak{R}_Z) + \cos(\mathfrak{R}_X) \cdot \cos(\mathfrak{R}_Z) \quad (2.12)$$

$$r_6 = \cos(\mathfrak{R}_X) \cdot \sin(\mathfrak{R}_Y) \cdot \sin(\mathfrak{R}_Z) - \sin(\mathfrak{R}_X) \cdot \cos(\mathfrak{R}_Z) \quad (2.13)$$

$$r_7 = -\sin(\mathfrak{R}_Y) \quad (2.14)$$

$$r_8 = \sin(\mathfrak{R}_X) \cdot \cos(\mathfrak{R}_Y) \quad (2.15)$$

$$r_9 = \cos(\mathfrak{R}_X) \cdot \cos(\mathfrak{R}_Y) \quad (2.16)$$

El problema inverso también es de interés, extraer los ángulos de orientación a partir de una matriz de orientación. Hay múltiples soluciones a este problema, debido a que se tienen nueve ecuaciones (las entradas de la matriz), pero tan solo tres incógnitas (los ángulos de orientación). La solución más utilizada es la del seno inverso, consiste en utilizar los valores y las ecuaciones de las entradas r_4 , r_7 y r_8 de la matriz de orientación, para obtener tres expresiones que permitan determinar el valor de los ángulos de orientación (Martínez, 2019b):

$$\mathfrak{R}_Y = \sin^{-1}(-r_7) \quad (2.17)$$

$$\mathfrak{R}_Z = \sin^{-1}\left(\frac{r_4}{\cos(\mathfrak{R}_Y)}\right) \quad (2.18)$$

$$\mathfrak{R}_X = \sin^{-1}\left(\frac{r_8}{\cos(\mathfrak{R}_Y)}\right) \quad (2.19)$$

Por otro lado, en la figura 2.5, se tiene una posición de un punto del objeto definida por el vector \mathbf{H}^* . Se quiere conocer la posición de dicho punto, pero con respecto al sistema de coordenadas de referencia, cuando el origen de este sistema coincide con el origen del sistema de coordenadas local. Esto solo es posible cuando se conoce la descripción de la orientación del sistema local respecto al de referencia, es decir, se conoce la matriz de orientación. Entonces, como las componentes de cualquier vector son simplemente las proyecciones de dicho vector en las direcciones unitarias de su sistema de coordenadas, se puede calcular la posición del punto del objeto respecto a la referencia mediante el producto de la matriz de orientación con el vector de posición \mathbf{H}^* (Martínez, 2019b):

$$\mathbf{H} = \mathbf{R} \cdot \mathbf{H}^* \quad (2.20)$$

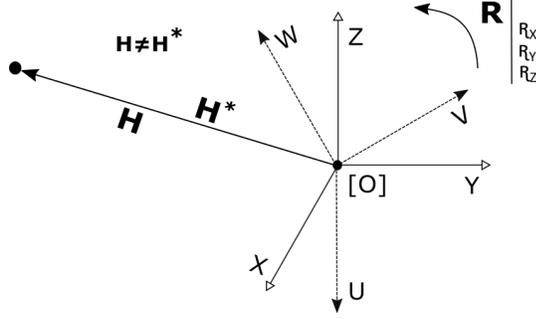


Figura 2.5: Posición de cualquier punto de un objeto cuando el origen de su sistema de coordenadas coincide con el origen del sistema de referencia (Martínez, 2019b).

2.1.5 Modelo de pose de un objeto rígido

La pose de un objeto rígido se puede describir por la posición del origen de su sistema de coordenadas local y la orientación de este respecto al sistema de coordenadas de referencia, es decir, la pose de un objeto se describe con el vector de posición \mathbf{G} (ecuación 2.4), y la matriz de orientación \mathbf{R} (ecuación 2.6) de su sistema local respecto a la referencia (Martínez, 2019b; Tsai, 1999). Entonces, la pose es la descripción general de cómo está ubicado un objeto rígido respecto a una referencia.

De forma compacta, la pose se puede representar mediante un vector multidimensional \mathbf{P} que depende de las componentes del vector de posición y de los ángulos de orientación (Martínez, 2019b):

$$\mathbf{P} = (G_X, G_Y, G_Z, \mathfrak{R}_X, \mathfrak{R}_Y, \mathfrak{R}_Z)^T \quad (2.21)$$

Por otro lado, ahora se considera el caso general de la figura 2.6, en el que se tiene la posición \mathbf{H}^* de un punto del objeto y se quiere conocer dicha posición respecto al sistema de coordenadas de referencia. En este caso, el origen del sistema local no coincide con el origen del sistema referencia, pero tiene un desplazamiento vectorial respecto a la referencia, que se representa con el vector de posición \mathbf{G} . Además, el sistema local no está alineado con el de referencia, tiene una cierta orientación respecto a este, que se describe con la matriz de orientación \mathbf{R} (Martínez, 2019b; Craig, 1999).

Entonces, primero se encuentra la posición de dicho punto respecto a un sistema de coordenadas intermedio o temporal que tiene la misma orientación

que el sistema de referencia y que su origen coincide con el del sistema local. Esto se hace mediante el uso de la ecuación 2.20, descrita en la sección anterior. Luego, simplemente se añade el vector de posición mediante una simple suma vectorial, como en la sección 2.1.3, para así obtener (Martínez, 2019b; Craig, 1999):

$$\mathbf{H} = \mathbf{R} \cdot \mathbf{H}^* + \mathbf{G} \quad (2.22)$$

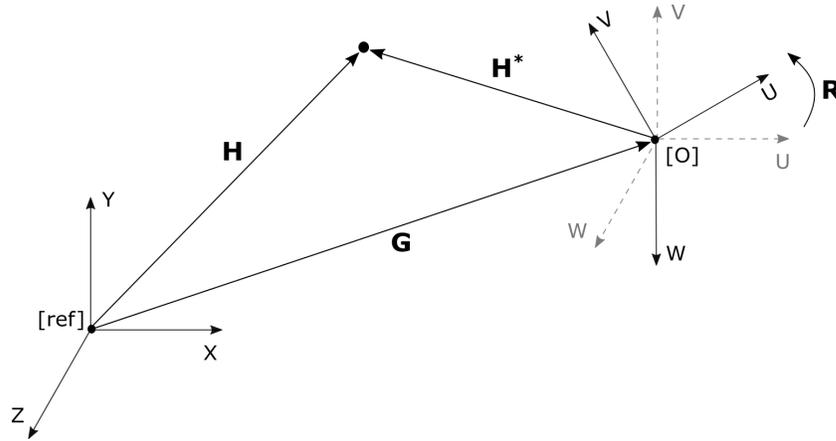


Figura 2.6: Posición de cualquier punto de un objeto cuando su sistema de coordenadas no está alineado y su origen no coincide con el sistema de referencia (Martínez, 2019b).

La ecuación 2.22 se le conoce como la ecuación de pose y permite calcular la posición de cualquier punto de un objeto respecto a un sistema de referencia cualquiera, si se tiene la posición de dicho punto respecto al sistema local del objeto y se conoce la pose del objeto (Martínez, 2019b).

2.1.6 Modelo de movimiento de un objeto rígido

El movimiento de un objeto rígido se puede describir mediante un cambio de orientación (rotación) de su sistema de coordenadas local $(U, V, W)^T$, seguidamente de un cambio de posición (traslación) del origen de dicho sistema, con respecto al sistema de referencia $(X, Y, Z)^T$ (Martínez, 2019b; Olgún, 2019), tal y como se observa en la figura 2.7. Entonces, el movimiento es la descripción del cambio de ubicación de un objeto rígido entre un instante t_k (antes del movimiento) y un instante t_{k+1} (después del movimiento), respecto a una referencia.

El modelo de orientación explicado en la sección 2.1.4 también es utilizado para describir de forma general la rotación de un objeto rígido respecto a una

referencia (Martínez, 2019b; Olguín, 2019), ya que dicho modelo lo que en realidad describe es el caso particular de un cambio de orientación o rotación de un sistema de coordenadas local respecto a un sistema de referencia, cuando ambos sistemas se encuentran totalmente alineados antes de la rotación. Entonces, se puede utilizar la matriz de la ecuación 2.6 para representar la rotación de un objeto, solo que ahora se le denomina como la matriz de rotación, en vez de orientación, y se denota diferente:

$$\Delta \mathbf{R} = \begin{pmatrix} r_1 & r_2 & r_3 \\ r_4 & r_5 & r_6 \\ r_7 & r_8 & r_9 \end{pmatrix} \quad (2.23)$$

Además, las entradas que conforman la matriz de rotación se pueden obtener con las ecuaciones de la 2.8 a la 2.16, solo que ahora son función de las entradas del vector de rotación $\Delta \mathbf{W}$, ecuación 2.24, que corresponden a los ángulos de rotación o la diferencia entre los ángulos de orientación antes y después del giro, respecto a cada uno de los ejes $(X, Y, Z)^T$.

$$\Delta \mathbf{W} = \begin{pmatrix} \Delta W_X \\ \Delta W_Y \\ \Delta W_Z \end{pmatrix} \quad (2.24)$$

Análogamente, el modelo de posición explicado en la sección 2.1.3 también es utilizado para describir la traslación de un objeto rígido respecto a una referencia (Martínez, 2019b; Olguín, 2019), ya que dicho modelo lo que en realidad describe es el caso particular de un cambio de posición o traslación del origen de un sistema de coordenadas local respecto a una referencia, cuando el origen de ambos sistemas coincide antes de la traslación. Entonces, se puede utilizar un vector tridimensional como el de la ecuación 2.4 para describir la traslación de un objeto, solo que ahora se le denomina como el vector de traslación, en vez de posición y se denota como $\Delta \mathbf{T}$, ecuación 2.25:

$$\Delta \mathbf{T} = \begin{pmatrix} \Delta T_X \\ \Delta T_Y \\ \Delta T_Z \end{pmatrix} \quad (2.25)$$

Este vector se extiende desde el origen del sistema de coordenadas del objeto antes del movimiento, hasta el origen del sistema de coordenadas del objeto después del movimiento.

Entonces, el movimiento se puede representar de forma compacta mediante un vector \mathbf{B} que depende de las componentes del vector de traslación y de los ángulos de rotación (Martínez, 2019b):

$$\mathbf{B} = (\Delta T_X, \Delta T_Y, \Delta T_Z, \Delta W_X, \Delta W_Y, \Delta W_Z)^T \quad (2.26)$$

Por otro lado, ahora se considera el caso general de la figura 2.7, se tiene un objeto con un sistema de coordenadas local $(U, V, W)^T$ en el instante t_k y se conoce su orientación y posición respecto a la referencia (\mathbf{R} y \mathbf{G} respectivamente), además, se conoce la posición de un punto del objeto respecto al sistema de coordenadas local en ese instante y respecto a la referencia (\mathbf{H}^* y \mathbf{H} respectivamente). Luego, el objeto se rota y se traslada respecto a la referencia ($\Delta\mathbf{R}$ y $\Delta\mathbf{T}$ son conocidos) y se obtiene una nueva ubicación del objeto en el instante t_{k+1} . Se quiere determinar la pose del objeto después del movimiento (la nueva orientación $\Delta\mathbf{R}'$ y la nueva posición $\Delta\mathbf{G}'$) y la posición del punto del objeto después del movimiento (\mathbf{H}') (Martínez, 2019b).

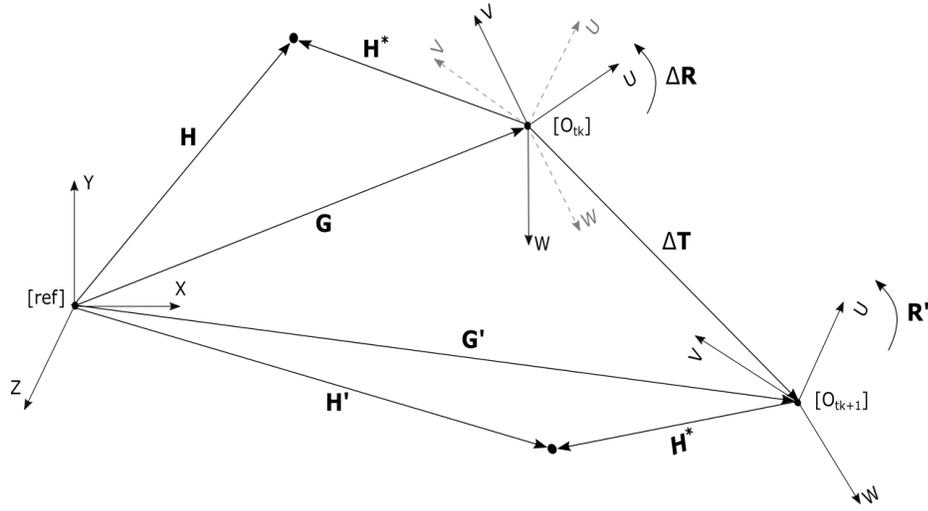


Figura 2.7: Posición de cualquier punto de un objeto después de una traslación y rotación, es decir, después de algún movimiento (Martínez, 2019b).

Primero se encuentra la nueva orientación del objeto. Recordemos que las entradas de la matriz de rotación dependen del respectivo vector de rotación, que es una resta vectorial entre el vector orientación antes del movimiento \mathbf{O} y el vector de orientación después del movimiento \mathbf{O}' , ecuación 2.27:

$$\Delta\mathbf{W} = \mathbf{O}' - \mathbf{O} = \begin{pmatrix} \mathfrak{R}'_X - \mathfrak{R}_X \\ \mathfrak{R}'_Y - \mathfrak{R}_Y \\ \mathfrak{R}'_Z - \mathfrak{R}_Z \end{pmatrix} \quad (2.27)$$

Y las entradas de la matriz de orientación antes del movimiento dependen del respectivo vector de orientación dado por la ecuación 2.7. Se puede demostrar matemáticamente que al multiplicar la matriz de rotación por la matriz de orientación antes del movimiento, se obtiene una nueva matriz cuyas entradas dependen solamente de las componentes del vector orientación

\mathbf{O}' después del movimiento (los ángulos de orientación en el instante t_{k+1}), es decir, se obtiene la nueva matriz de orientación en el instante t_{k+1} (Martínez, 2019b; Olguín, 2019):

$$\mathbf{R}' = \Delta\mathbf{R} \cdot \mathbf{R} \quad (2.28)$$

La nueva posición se puede encontrar fácilmente a partir de la definición del vector de traslación. Análogamente al vector de orientación, el vector de traslación es una resta vectorial entre el vector de posición \mathbf{G} antes del movimiento y el vector de posición \mathbf{G}' después del movimiento:

$$\Delta\mathbf{T} = \mathbf{G}' - \mathbf{G} = \begin{pmatrix} G'_X - G_X \\ G'_Y - G_Y \\ G'_Z - G_Z \end{pmatrix} \quad (2.29)$$

Si se despeja el vector \mathbf{G}' de la ecuación anterior, se obtiene una expresión para el nuevo vector de posición en el instante t_{k+1} (Martínez, 2019b):

$$\mathbf{G}' = \mathbf{G} + \Delta\mathbf{T} \quad (2.30)$$

Entonces, la pose del objeto después del movimiento, en el instante t_{k+1} y denotada como \mathbf{P}' , se describe con el vector posición \mathbf{G}' , dado por la ecuación 2.30 y la matriz de orientación \mathbf{R}' , dada por la ecuación 2.28, ambos referidos al sistema de referencia.

Finalmente, la nueva posición del punto del objeto respecto a la referencia (\mathbf{H}') se obtiene a partir de la ecuación 2.5, pero utilizando la nueva pose del objeto (Martínez, 2019b):

$$\mathbf{H}' = \mathbf{R}' \cdot \mathbf{H}^* + \mathbf{G}' \quad (2.31)$$

Y si se sustituye la ecuación 2.30 y 2.28 en la expresión anterior:

$$\mathbf{H}' = \Delta\mathbf{R} \cdot \mathbf{R} \cdot \mathbf{H}^* + \mathbf{G} + \Delta\mathbf{T} \quad (2.32)$$

A la ecuación 2.32 se le conoce como la ecuación de pose después del movimiento. Además, si se despeja la expresión $\mathbf{R} \cdot \mathbf{H}^*$ de la ecuación de pose 2.22 y se sustituye la expresión resultante en la ecuación de pose después del movimiento, dada por 2.5, se obtiene la siguiente ecuación equivalente:

$$\mathbf{H}' = \Delta\mathbf{R} \cdot \mathbf{R} \cdot (\mathbf{H} - \mathbf{G}) + \mathbf{G} + \Delta\mathbf{T} \quad (2.33)$$

A la ecuación 2.33 se le conoce como la ecuación de movimiento y permite calcular después del movimiento de un objeto, la posición de cualquier punto de este respecto a un sistema de referencia, si se conoce la posición de dicho punto antes del movimiento respecto a la referencia, la pose del objeto antes

del movimiento y el movimiento que este experimentó (traslación y rotación) (Martínez, 2019b).

2.1.7 Odometría de ruedas

La odometría de ruedas es uno de los métodos más maduros y más básicos que se puede utilizar para estimar la pose de un robot después del movimiento \mathbf{P}' , respecto a un sistema de referencia en el espacio $(X, Y, Z)^T$, siempre y cuando la pose inicial antes del movimiento, \mathbf{P} , sea conocida (Klancar et al., 2017).

Este método se basa principalmente en la obtención de la distancia recorrida por la rueda izquierda y la rueda derecha del robot (d_l y d_r), entre dos instantes de tiempo t_k y t_{k+1} (antes y después del movimiento), mediante el uso de codificadores ópticos ubicados en cada rueda (Martínez, 2019b; Klancar et al., 2017).

Un codificador óptico es un dispositivo electromecánico que utiliza una fuente de luz, detectores fotosensibles y rejillas ópticas para medir la rotación de la rueda. Sus partes y su funcionamiento se muestran en la figura 2.8. Como se puede observar en dicha figura, un láser emite un rayo de luz sobre un anillo alrededor del eje de la rueda. El anillo tiene rejillas ópticas, que son cortes perpendiculares a la rotación. De esta manera, cuando el eje y el anillo giran, el detector fotosensible detecta los pulsos de luz. El codificador cuenta cada uno de los pulsos entre los instantes t_k y t_{k+1} . Al final del movimiento, se hace un recuento de los pulsos que se traduce directamente en el ángulo de rotación de la rueda (δ), dado en radianes (Martínez, 2019b).

Entonces, mediante el uso de codificadores ópticos se puede obtener el ángulo de rotación de la rueda izquierda y la rueda derecha (δ_l y δ_r). Finalmente, la distancia recorrida por cada rueda se puede obtener a partir de la fórmula para calcular una longitud de un arco, si se conoce el radio de las ruedas (r) (Martínez, 2019b):

$$d_l = r\delta_l \quad (2.34)$$

$$d_r = r\delta_r \quad (2.35)$$

Una vez que se conoce la distancia recorrida por cada rueda entre el instante t_k y t_{k+1} , se puede utilizar el modelo cinemático de un robot móvil respecto al sistema de referencia, para encontrar su nueva pose después del movimiento. Considérese el modelo de la figura 2.9, que aproxima el movimiento del robot, durante un intervalo de tiempo pequeño, mediante un arco.

Donde:

- $d_{baseline}$ es la distancia entre ruedas (centro del eje).

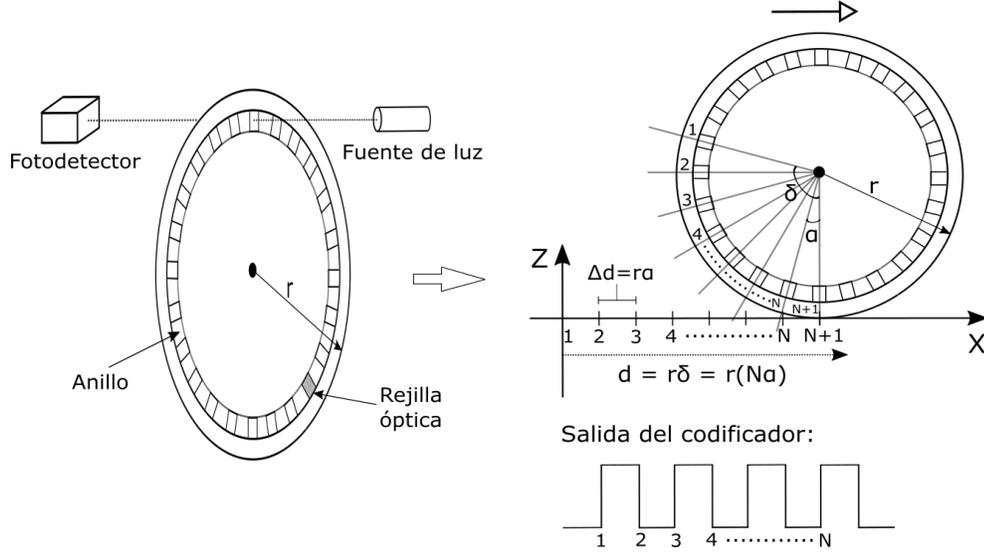


Figura 2.8: Diagrama de las partes y el funcionamiento de un codificador óptico (Martínez, 2019b).

- d_c es la distancia recorrida por el centro del eje.
- r_l radio de la circunferencia que se forma entre el origen del sistema de referencia y el centro de la rueda izquierda.
- r_r radio de la circunferencia que se forma entre el origen del sistema de referencia y el centro de la rueda derecha.
- r_c radio de la circunferencia que se forma entre el origen del sistema de referencia y el centro del eje.
- ϕ ángulo del interno de los arcos formados por la distancia recorrida por las ruedas.
- α es la orientación del centro del eje antes del movimiento.
- α' es la orientación del centro del eje después del movimiento.
- $\mathbf{H} = (H_X, H_Y)^T$ representa la posición del centro del eje antes del movimiento.
- $\mathbf{H}' = (H'_X, H'_Y)^T$ representa la posición del centro del eje después del movimiento.

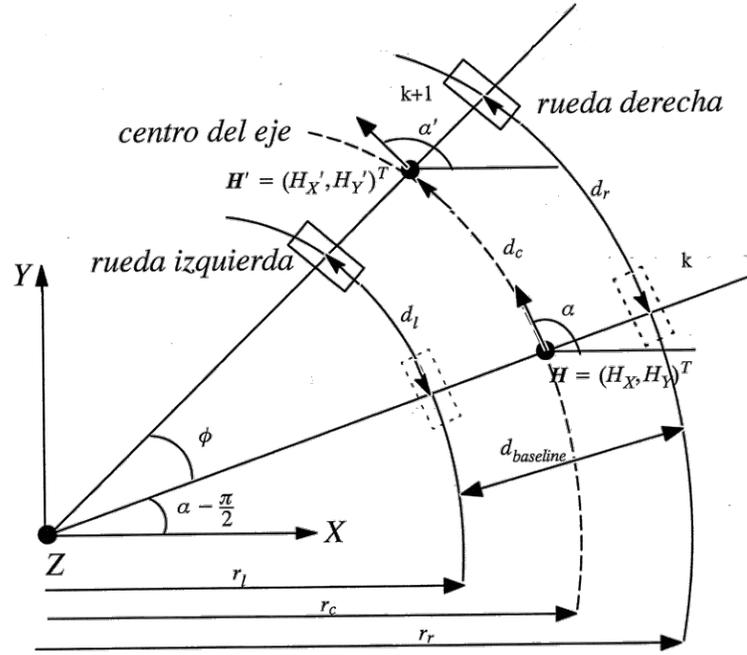


Figura 2.9: Geometría detallada de la odometría de ruedas (Martínez, 2019b).

Además, la pose del robot antes del movimiento, la cual es conocida, y la pose del robot después del movimiento, la cual se quiere estimar, están dadas por las siguientes ecuaciones, respectivamente (Martínez, 2019b):

$$\mathbf{P} = \begin{pmatrix} H_X \\ H_Y \\ \alpha \end{pmatrix} \quad (2.36)$$

$$\mathbf{P}' = \begin{pmatrix} H'_X \\ H'_Y \\ \alpha' \end{pmatrix} \quad (2.37)$$

Entonces, mediante el uso de geometría básica se obtienen las siguientes expresiones para d_c y ϕ , en términos de variables conocidas (Martínez, 2019b; Olson, 2003):

$$d_c = \frac{d_r + d_l}{2} \quad (2.38)$$

$$\phi = \frac{d_r - d_l}{d_{baseline}} \quad (2.39)$$

Luego, mediante el uso de trigonometría, geometría de ángulos y las ecuaciones anteriores, se puede obtener finalmente una expresión para la pose del robot después del movimiento, en el instante t_{k+1} (Martínez, 2019b; Olson, 2003):

$$\mathbf{P}' = \begin{pmatrix} H'_X \\ H'_Y \\ \alpha' \end{pmatrix} = \begin{pmatrix} H_X + d_c \cos(\alpha') \\ H_Y + d_c \sin(\alpha') \\ \alpha + \phi \end{pmatrix} \quad (2.40)$$

Si el robot se mueve en línea recta, la distancia recorrida por la rueda izquierda será igual a la distancia recorrida por la rueda derecha ($d_l=d_r$) y, por lo tanto, el ángulo ϕ de la ecuación 2.39 se vuelve cero, lo que implica que el ángulo α' es exactamente igual al ángulo α (Martínez, 2019b).

Por otro lado, el cálculo de la nueva orientación α' mediante el uso de odometría de ruedas es muy impreciso, por lo que frecuentemente se utiliza solamente para determinar la nueva posición del robot \mathbf{H}' , en lugar de su pose, mientras que la determinación de su nueva orientación se realiza empleando una unidad de medición inercial, conocida como IMU (Martínez, 2019b).

Además, la precisión de la odometría de ruedas para obtener la posición del robot, depende en gran medida del tipo de terreno. Se puede obtener una precisión de $<1\%$ de la distancia recorrida cuando el robot navega por superficies duras y uniformes, pero el deslizamiento por inclinaciones pronunciadas o suelos con piedras sueltas, puede degradar considerablemente la precisión (Gao, 2016).

2.1.8 Unidad de medición inercial

Una unidad de medición inercial (IMU, por sus siglas en inglés) es un dispositivo electrónico que se encuentra atado rígidamente en muchos objetos rígidos que experimentan movimiento constante y se encarga de proporcionar los tres ángulos de orientación $(\mathfrak{R}_X, \mathfrak{R}_Y, \mathfrak{R}_Z)^T$, el vector de orientación \mathbf{O} , que describen la orientación de su sistema de coordenadas denominado $(R, S, T)^T$, con respecto a un sistema de coordenadas de referencia denominado $(X, Y, Z)^T$, tal y como se observa en la figura 2.10, utilizando una combinación de acelerómetros 3D y giroscopios 3D (6 grados de libertad). La salida de la IMU usualmente es una matriz de orientación \mathbf{R} evaluada en los respectivos ángulos de orientación (Martínez, 2019b).

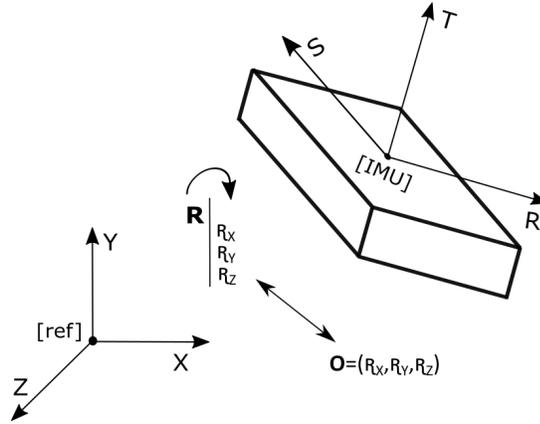


Figura 2.10: Esquemático de una unidad de medición inercial respecto al sistema de referencia (Martínez, 2019b).

Un acelerómetro 3D o de tres ejes es un arreglo de tres acelerómetros internos montados de forma ortogonal, es decir, un sensor montado a lo largo de cada uno de los ejes $(R, S, T)^T$, de tal forma que se sientan las tres componentes principales de la aceleración, como se observa en la figura 2.11. Entonces, el acelerómetro 3D lo que hace es medir el vector \mathbf{A} de aceleración lineal, que se descompone en tres componentes ortogonales $(A_R, A_S, A_T)^T$ y que representa el cambio de velocidad que experimenta el objeto a lo largo de cada uno de sus ejes. La aceleración usualmente se da en fuerzas G , una fuerza G en el planeta Tierra es equivalente a $9,8m/s^2$, pero esto varía con la elevación y es diferente en cada planeta debido a las variaciones en la atracción gravitacional. A partir del vector de aceleración se puede obtener el vector de orientación mediante relaciones trigonométricas, que se denota como \mathbf{O}_A y sus componentes como $(\mathfrak{R}_{AR}, \mathfrak{R}_{AS}, \mathfrak{R}_{AT})^T$ (Tran, 2017).

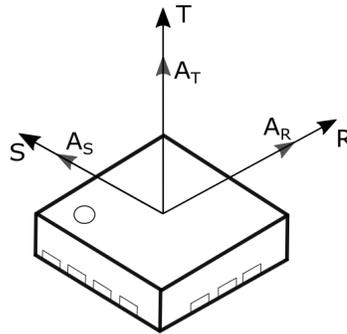


Figura 2.11: Acelerómetro tridimensional compuesto por tres sensores montados ortogonalmente.

Análogamente, un giroscopio 3D o de tres ejes es un arreglo de tres giroscopios internos montados de forma ortogonal, como se observa en la figura 2.12. Entonces, el giroscopio 3D lo que hace es medir el vector \mathbf{W} de velocidad angular, que se descompone en tres componentes ortogonales $(W_R, W_S, W_T)^T$ y que representa el cambio de orientación que experimenta el objeto alrededor de cada uno de sus ejes. La velocidad angular usualmente se da en radianes por segundo. A partir del vector de velocidad angular se puede obtener el vector de orientación mediante simple integración, que se denota como \mathbf{O}_W y sus componentes como $(\mathfrak{R}_{WR}, \mathfrak{R}_{WS}, \mathfrak{R}_{WT})$ (Tran, 2017).

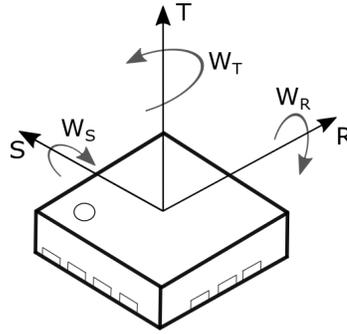


Figura 2.12: Giroscopio tridimensional compuesto por tres sensores montados ortogonalmente.

Lo que hace la IMU es combinar los vectores de orientación proporcionados por el acelerómetro 3D (\mathbf{O}_A) y el giroscopio 3D (\mathbf{O}_W), mediante un método de fusión de sensores, para obtener un vector de orientación más preciso (\mathbf{O}) respecto al sistema de coordenadas de referencia. Esta acción de combinar los resultados de ambos sensores, los vectores de orientación, se realiza mediante la implementación de un filtro de Kalman, que es una herramienta poderosa que combina información en presencia de incertidumbre. Finalmente, se obtiene la matriz de orientación, al evaluar sus entradas en el vector de orientación \mathbf{O} obtenido mediante la fusión de sensores, que es la salida final (Tran, 2017; Gao, 2016; Martínez, 2019b).

Cabe destacar que, para el caso de la sección anterior, que se describe en la figura 2.9, la IMU solamente calcularía el ángulo de orientación \mathfrak{R}_Z , que también se denota como α , ya que el robot se mueve sobre el plano XY . Entonces, la salida final de la IMU sería la matriz de orientación respecto al eje Z (Martínez, 2019b).

2.1.9 Modelo de pinhole de una cámara

Una imagen es en realidad una proyección de una escena del espacio 3D (el espacio de los objetos) sobre un subespacio bidimensional 2D. Para proyectar

un punto del espacio sobre una imagen, se necesita conocer la relación que existe entre las coordenadas de la imagen y el espacio 3D del mundo, mediante un modelo de la cámara, como el que se aprecia en la figura 2.13, el cual se conoce como el modelo de pinhole (Hartley y Zisserman, 2004; Martínez, 2019b).

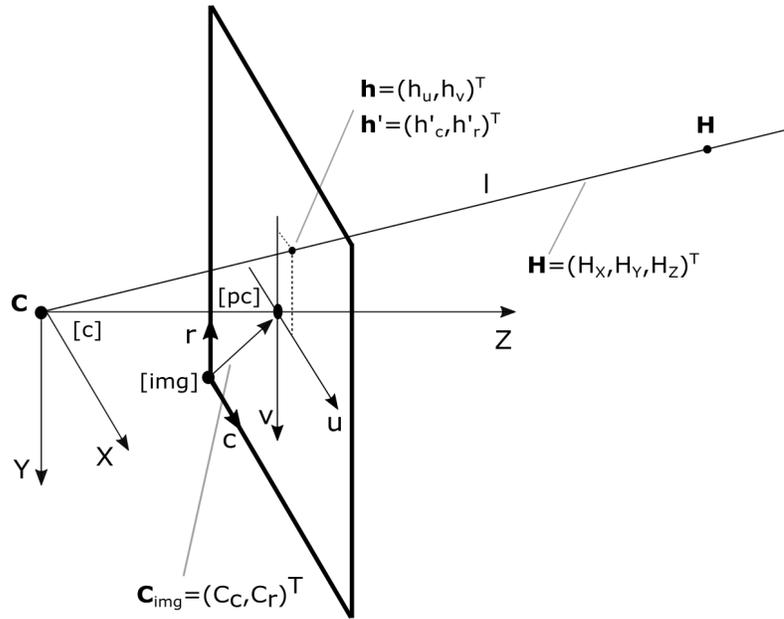


Figura 2.13: Modelo de pinhole de una cámara (Martínez, 2019b).

Donde:

- $[c]$ representa al objeto cámara en el espacio.
- f es la distancia focal de la cámara. Se da en milímetros.
- $(X, Y, Z)^T$ es el sistema de coordenadas de la cámara y también corresponde al sistema de coordenadas de referencia. El eje Z coincide con el eje óptico de la cámara.
- C es el centro óptico de la cámara, el cual coincide con el origen de su sistema de coordenadas.
- $H = (H_X, H_Y, H_Z)^T$ es un punto cualquiera de un objeto rígido ubicado en el espacio, pero que se encuentra en la escena captada por la cámara y se conoce su posición tridimensional respecto al sistema de coordenadas de esta. Las posiciones se dan en centímetros o metros.

- $[pc]$ representa al plano de la cámara, dado por la ecuación $Z = f$, el cual es paralelo al plano XY.
- $(u, v)^T$ es el sistema de coordenadas del plano de la cámara. Su origen corresponde a la intersección del eje Z (eje óptico) con el plano $Z = f$. La orientación de sus ejes coincide con la de los ejes X y Y.
- $\mathbf{h} = (h_u, h_v)^T$ es el vector de posición bidimensional del punto \mathbf{H} respecto al sistema de coordenadas del plano de la cámara. Es la proyección de \mathbf{H} sobre el plano de la cámara. Las unidades de este vector son milímetros.
- $[img]$ representa a la imagen, la cual se podría decir que es un rectángulo de largo w y ancho h , ubicado en el plano de la cámara, cuyo centro es el origen de $(u, v)^T$.
- $(c, r)^T$ es el sistema de coordenadas de la imagen. Su origen está ubicado en la esquina superior izquierda de la misma. El eje c está alineado con el eje u y el eje r está en la dirección contraria al eje v ($r = -v$).
- $\mathbf{h}' = (h'_c, h'_r)^T$ es el vector de posición bidimensional del punto \mathbf{H} respecto al sistema de coordenadas de la imagen. Es una transformación del punto \mathbf{h} del plano de la cámara a la imagen. También se podría decir que es la proyección de \mathbf{H} sobre la imagen. Las unidades de este vector son píxeles.
- l es una línea que inicia en el origen del sistema de coordenadas de la cámara, pasa por el punto P_i en el espacio y se extiende al infinito. A dicha línea se le conoce como la línea vista. Además, la intersección de esta línea con el plano de la cámara es el punto \mathbf{h} .
- $\mathbf{C}_{img} = (C_c, C_r)^T$ es un vector de posición bidimensional que representa la posición del sistema de coordenadas del plano de la cámara respecto a la imagen. Sus unidades son píxeles y las componentes están dadas por:

$$C_c = \frac{w}{2} \quad (2.41)$$

$$C_r = \frac{h}{2} \quad (2.42)$$

Dado el modelo de la figura 2.13, se quiere encontrar una expresión que permita calcular la proyección del punto \mathbf{H} sobre el plano de la cámara, en términos de variables conocidas, es decir, se quiere encontrar una expresión para \mathbf{h} . Para ello, se considera solamente el plano YZ de la figura 2.13, el cual se aprecia mejor en la figura 2.14 (Hartley y Zisserman, 2004).

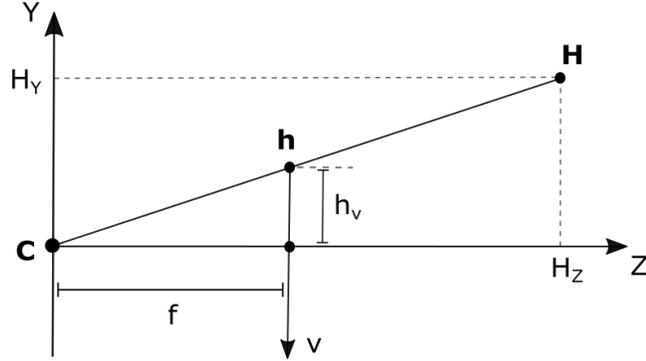


Figura 2.14: Vista del plano YZ del modelo de pinhole de una cámara.

En la figura anterior, se puede hacer uso del teorema de triángulos semejantes para así obtener una expresión para la componente h_v del vector \mathbf{h} (Martínez, 2019b; Hartley y Zisserman, 2004):

$$h_v = f \frac{H_Y}{H_Z} \quad (2.43)$$

Análogamente, si solamente se considera el plano XZ, se puede obtener una expresión para la componente h_u (Martínez, 2019b; Hartley y Zisserman, 2004):

$$h_u = f \frac{H_X}{H_Z} \quad (2.44)$$

Entonces, la proyección del punto \mathbf{H} sobre el plano de la cámara se puede calcular de la siguiente forma, usando las ecuaciones 2.44 y 2.43 (Martínez, 2019b; Hartley y Zisserman, 2004):

$$\mathbf{h} = \begin{pmatrix} h_u \\ h_v \end{pmatrix} = f \begin{pmatrix} \frac{H_X}{H_Z} \\ \frac{H_Y}{H_Z} \end{pmatrix} \quad (2.45)$$

Ahora se desea encontrar una expresión para calcular la posición del punto \mathbf{H} respecto al sistema de coordenadas de la imagen, es decir, se quiere encontrar una expresión para calcular \mathbf{h}' . Para ello, se considera solamente la imagen y el plano de la cámara de la figura 2.13, tal y como se observa en la figura 2.15 (Martínez, 2019b; Pajares y de la Escalera, 2016).

donde:

- N_u es el número de píxeles por sensor a lo largo del eje u .
- d_u es la distancia entre sensores en u , dada en milímetros.

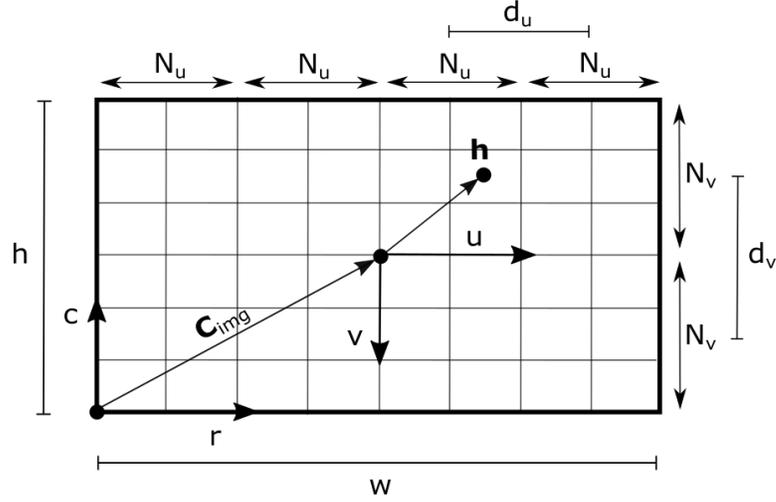


Figura 2.15: Relación entre el sistema de coordenadas del plano de la cámara, el sistema de coordenadas de la imagen y los sensores (Pajares y de la Escalera, 2016).

- N_v es el número de píxeles por sensor a lo largo del eje v .
- d_v es la distancia entre sensores en v , dada en milímetros.

Primero se realiza una conversión de milímetros a píxeles de las componentes del punto \mathbf{h} , mediante el uso de las variables anteriores (Martínez, 2019b):

$$h_u^* = N_u \left(\frac{\text{píxeles}}{\text{sensor}} \right) \cdot \frac{1}{d_u} \left(\frac{\text{sensor}}{\text{mm}} \right) \cdot h_u(\text{mm}) = N_u \frac{h_u}{d_u} (\text{píxeles}) \quad (2.46)$$

$$h_v^* = N_v \left(\frac{\text{píxeles}}{\text{sensor}} \right) \cdot \frac{1}{d_v} \left(\frac{\text{sensor}}{\text{mm}} \right) \cdot v_i(\text{mm}) = N_v \frac{h_v}{d_v} (\text{píxeles}) \quad (2.47)$$

Con las ecuaciones anteriores se puede obtener una expresión para el punto \mathbf{h} pero dado en píxeles, este nuevo punto se denota como \mathbf{h}^* . Además, en la mayoría de las cámaras el número de píxeles por sensor a lo largo del eje v es igual 1 ($N_v = 1$) (Martínez, 2019b; Pajares y de la Escalera, 2016). Entonces, teniendo en cuenta la consideración anterior y las ecuaciones 2.46 y 2.47, la expresión para \mathbf{h}^* es:

$$\mathbf{h}^* = \begin{pmatrix} h_u^* \\ h_v^* \end{pmatrix} = \begin{pmatrix} N_u \frac{h_u}{d_u} \\ \frac{h_v}{d_v} \end{pmatrix} \quad (2.48)$$

Ahora simplemente basta con aplicar la ecuación de posición, dada por la expresión 2.5 de la sección 2.1.3, al punto \mathbf{h}^* , dado por la expresión 2.48. Solamente hay que tener en cuenta que la posición \mathbf{G} del sistema de coordenadas del plano de la cámara respecto al de la imagen es \mathbf{C}_{img} y sus componentes están dadas por las expresiones 2.41 y 2.42. También hay que considerar la relación que existe entre los ejes ($c = u$ y $r = -v$). Entonces, teniendo en cuenta lo anterior y aplicando la ecuación de posición al punto \mathbf{h}^* , se obtiene finalmente una expresión para calcular la proyección del punto \mathbf{H} , pero sobre la imagen, en vez del plano de la cámara (Martínez, 2019b):

$$\mathbf{h}' = \begin{pmatrix} h'_c \\ h'_r \end{pmatrix} = \begin{pmatrix} h_u^* + C_c \\ C_r - h_v^* \end{pmatrix} = \begin{pmatrix} N_u \frac{h_u}{d_u} + \frac{w}{2} \\ \frac{h}{2} - \frac{h_v}{d_v} \end{pmatrix} \quad (2.49)$$

Mediante el uso de las ecuaciones 2.45 y 2.49 se puede proyectar cualquier punto del espacio que se encuentra en la escena vista por la cámara, a un punto en el plano de la cámara o a un píxel en la imagen, siempre y cuando dicho punto este referido al sistema de coordenadas de la cámara (Martínez, 2019b).

2.2. Estereoscopia

El método utilizado por los seres humanos para determinar la profundidad del entorno que los rodea mediante la visión binocular es la estereoscopia (Szeliski, 2011). 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 estos, y es a partir de esa disparidad que al mezclarlas se logra la percepción de la profundidad.

Muchos de los algoritmos en visión por computador utilizan técnicas de visión estereoscópica para tratar de imitar la capacidad que tienen los seres humanos para deducir la información geométrica (3D) de una escena (profundidad y relieve). Dichas técnicas se basan en el procesamiento y análisis de dos imágenes distintas para compensar la pérdida de información que se genera cuando se proyecta una escena tridimensional en una imagen que es 2D (Szeliski, 2011).

Para poder reconstruir una escena a partir de dos imágenes es necesario identificar las proyecciones de varios puntos de la escena en las dos imágenes, para luego, utilizar el modelo y la geometría del sistema estereoscópico para calcular las posiciones tridimensionales de dichos puntos en el espacio, a este proceso se le conoce como estimación de profundidad (Pajares y de la Escalera, 2016).

En la presente sección se analiza la conexión que existe entre dos imágenes de una misma escena y que corresponde al caso más simple del análisis multi-

imagen. Dicho análisis permite reconstruir una escena, recobrar la información tridimensional y es vital en aplicaciones de robots móviles, ingeniería inversa y reconstrucción de panoramas, entre otros.

2.2.1 Geometría de lo sistemas estereoscópicos

Dos imágenes tomadas desde distintas posiciones están relacionadas entre sí, debido a que estas son dos vistas de las proyecciones de una misma escena 3D. La geometría epipolar es la geometría proyectiva intrínseca que relaciona dichas vistas y es independiente de la estructura de la escena, solo depende de los parámetros internos de las cámaras y la pose relativa (Hartley y Zisserman, 2004).

A una de las imágenes se le conoce como imagen izquierda, ya que esta es capturada por una cámara que se encuentra a la izquierda de la escena y que tiene un centro óptico dado por \mathbf{C}_l . Análogamente, a la otra imagen se le conoce como imagen derecha y es capturada por una cámara que se encuentra a la derecha de la escena, con un centro óptico denotado como \mathbf{C}_r . En la figura 2.16 se muestran todas las entidades geométricas que intervienen entre esas dos imágenes, cuando se quiere encontrar la posición tridimensional de un punto de la escena, denotado como \mathbf{H} . El conjunto de estas entidades es la geometría epipolar y se describen a continuación (Martínez, 2019b; Hartley y Zisserman, 2004; Pajares y de la Escalera, 2016):

- Línea base: es un segmento 3D que une el centro óptico de la cámara izquierda con el de la cámara derecha, es decir, une el punto tridimensional \mathbf{C}_l con el punto \mathbf{C}_r . Se denota como l_c
- Plano epipolar: es el plano que contiene el triángulo conformado por el origen del sistema de coordenadas de la cámara izquierda (el centro óptico \mathbf{C}_l), el origen del sistema de coordenadas de la cámara derecha (el centro óptico \mathbf{C}_r) y la posición tridimensional \mathbf{H} de un punto de la escena en el espacio. Se denota como π .
- Epipolo: para ambas imágenes del sistema estéreo, es el punto de intersección de la línea base, l_c , con el plano de la cámara asociada a dicha imagen. Entonces, la intersección de la línea base con el plano de la cámara izquierda se denomina epipolo izquierdo y se denota como \mathbf{e}_l . Análogamente, la intersección con el plano de la cámara derecha es el epipolo derecho \mathbf{e}_r .
- Línea epipolar: para ambas cámaras, es la proyección de la respectiva línea vista sobre el plano de la respectiva cámara. Entonces, la proyección de cada uno de los puntos que conforman la línea vista izquierda, l_l ,

sobre el plano de la cámara izquierda, forman la línea epipolar izquierda y se denota como l_{er} . Análogamente, la línea epipolar derecha l_{el} , es la proyección de la línea vista derecha l_r sobre el plano de la cámara derecha. Cabe destacar que la línea epipolar se puede definir para ambas imágenes como la intersección del plano epipolar π con el respectivo plano de la cámara.

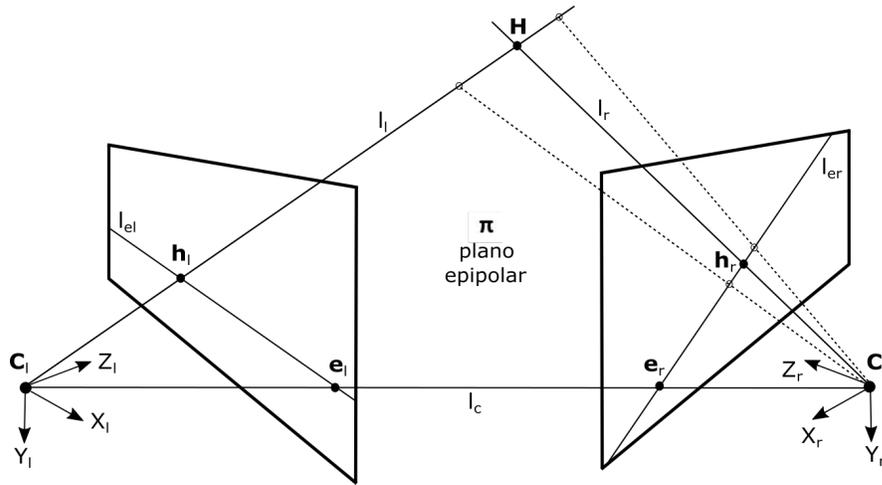


Figura 2.16: Representación de la geometría epipolar que relaciona dos imágenes de una misma escena (Martínez, 2019b).

Como se observa en la figura anterior, la proyección h_r del punto tridimensional H en el plano de la cámara derecha yace sobre la línea epipolar derecha l_{er} . Esto implica que una vez que se encuentre la proyección h_l del punto H en el plano de la cámara izquierda, la proyección h_r sobre el plano de la cámara derecha se debe de buscar únicamente a lo largo de la línea epipolar derecha. A lo anterior se le conoce como la restricción epipolar y únicamente es posible cuando las cámaras están calibradas: se conoce la posición de sus centros ópticos y su orientación respecto a un sistema de coordenadas de referencia, además de sus respectivas distancias focales (Martínez, 2019b).

2.2.2 Modelo canónico de una cámara estereoscópica

Un caso especial de un sistema estereoscópico es aquel que se encuentra formado por dos cámaras completamente idénticas y que han sido previamente rectificadas, esto quiere decir que sus ejes ópticos son paralelos ($Z_l \parallel Z_r$) y la línea base que une sus centros ópticos (C_l y C_r) es paralela a los planos de

ambas cámaras, es decir, a las filas de las imágenes. A este caso especial se le conoce como la configuración estéreo canónica y en la figura 2.17 se muestra la forma esquemática el modelo que la describe (Martínez, 2019b; Pajares y de la Escalera, 2016):

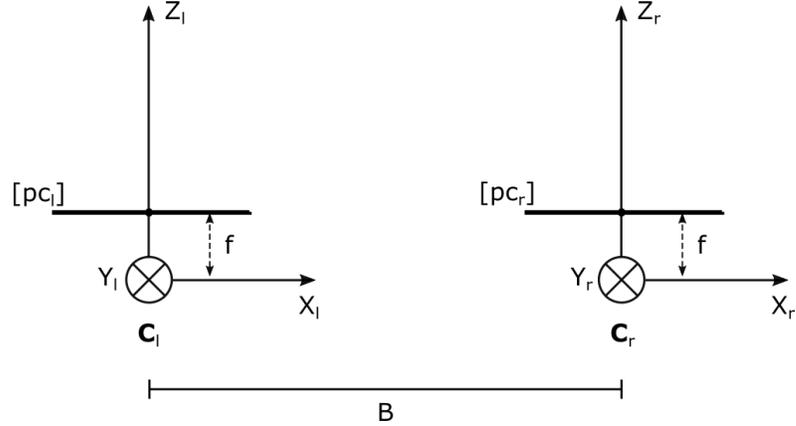


Figura 2.17: Modelo estéreo canónico (Martínez, 2019b; Pajares y de la Escalera, 2016).

donde:

- C_l es el centro óptico de la cámara izquierda.
- C_r es el centro óptico de 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 distancia de la línea base (l_c), es decir, la distancia entre los orígenes de los sistemas de coordenadas de las cámaras (los centros ópticos).
- f es la distancia focal. Como ambas cámaras son idénticas, la distancia focal es la misma para ambas ($f_r = f_l = f$).

En esta configuración los planos de ambas cámaras son coincidentes y, por lo tanto, las líneas epipolares no se encuentran inclinadas en la imagen, como en la figura 2.16, más bien son líneas que coinciden con las filas de las imágenes.

Además, los epipolos están en un punto del infinito correspondiente al eje u , ya que la línea base nunca se va a intersectar con alguno de los planos de ambas cámaras. Dichos epipolos se pueden representar de la forma $(1, 0, 0)^T$ (Bradski y Kaehler, 2008; Pajares y de la Escalera, 2016).

Por otro lado, la mayoría de las cámaras estereoscópicas son un arreglo de dos cámaras idénticas que han sido previamente rectificadas y, por lo tanto, se pueden modelar y describir mediante el esquema de la figura 2.17, es decir, tienen una configuración canónica. Este tipo de cámaras tratan de imitar completamente el comportamiento de los ojos humanos: un lente izquierdo (ojo izquierdo) y un lente derecho (ojo derecho) separados por una distancia fija (distancia de separación entre los ojos). De esta forma, se puede estimar la profundidad de los puntos de la escena (posiciones tridimensionales) de una manera muy simple, como lo hace el ser humano.

2.2.3 Estimación de profundidad por estereoscopía

La estimación de profundidad consiste en la determinación de la posición tridimensional $\mathbf{H} = (H_{X_l}, H_{Y_l}, H_{Z_l})^T$ de un punto de la escena elegido cuidadosamente, a partir de su proyección $\mathbf{h}'_l = (h'_{c_l}, h'_{r_l})^T$ en una imagen izquierda y su proyección $\mathbf{h}'_r = (h'_{c_r}, h'_{r_r})^T$ en una imagen derecha, capturadas por una cámara estereoscópica rectificada y que ha sido previamente calibrada, mediante una técnica llamada triangulación estereoscópica que se basa en el modelo estéreo canónico de la figura 2.17 (Martínez, 2019b; Pajares y de la Escalera, 2016).

Cabe destacar que cuando se desea estimar la profundidad, usualmente se trabaja con imágenes de intensidad y no con imágenes RGB. Existen cámaras estereoscópicas que ya dan la imagen de intensidad sin la necesidad de algún proceso adicional. Pero hay otras cámaras que solamente dan la imagen RGB y entonces se tiene que aplicar la ecuación 2.2. En este caso se supone que las imágenes izquierda y derecha son imágenes de intensidad.

Para calcular la posición tridimensional \mathbf{H} de un punto de la escena respecto a un sistema de coordenadas de referencia (en este caso es el sistema de coordenadas de la cámara izquierda), se debe de conocer de antemano la posición y orientación del sistema de coordenadas de la cámara derecha (Bradski y Kaehler, 2008). Para una cámara estereoscópica en configuración canónica, ambos sistemas de coordenadas se encuentran alineados. Por lo tanto, la matriz que describe la orientación del sistema de coordenadas de la cámara derecha con respecto a la referencia, es la matriz identidad:

$$\mathbf{R}_r^l = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (2.50)$$

Además, como los orígenes de ambas cámaras se encuentran separados por una distancia B que corresponde a la distancia de la línea base, el vector que describe la posición del sistema de coordenadas de la cámara derecha respecto a la izquierda esta dado por:

$$\mathbf{G}_r^l = \begin{pmatrix} B \\ 0 \\ 0 \end{pmatrix} \quad (2.51)$$

Una vez que se conoce la posición y orientación del sistema de coordenadas de la cámara derecha, se elige cuidadosamente un punto característico sobre la imagen izquierda, que representa la proyección \mathbf{h}'_l de un punto \mathbf{H} de la escena sobre dicha imagen (ver la figura 2.18) y del cual se quiere conocer su posición tridimensional (Martínez, 2019b).

Un punto característico es un píxel de la imagen que tiene una posición bien definida y que puede ser detectado de forma confiable y robusta. Pueden ser esquinas, un punto con un valor máximo o mínimo de intensidad, un punto al final de una línea, etc. La ventaja de estos puntos es que tienen suficiente información para ser encontrados fácilmente entre imágenes tomadas desde distintas posiciones y/o instantes de tiempo. Existen diferentes métodos para encontrar puntos característicos sobre una imagen, conocidos como operadores de interés, donde los más populares son: Forstner, Beyer, Shi-Tomasi y Harris. Estos se aplican a una imagen y se seleccionan aquellos píxeles con los valores de interés más altos. La mayoría de los operadores de interés se basan en la evaluación de gradientes lineales de intensidad para detectar esquinas en una imagen dada. (Shi y Tomasi, 1994; Bradski y Kaehler, 2008; Martínez, 2019b).

Una vez que se eligió el punto característico \mathbf{h}'_l , se debe determinar la proyección \mathbf{h}'_r del punto \mathbf{H} sobre la imagen derecha mediante un establecimiento de correspondencias estereoscópicas, tomando en cuenta la restricción epipolar. Un píxel de la imagen derecha corresponderá al punto característico dado por \mathbf{h}'_l , siempre y cuando este se encuentre sobre la respectiva línea epipolar en la imagen derecha (l_{er}) y sea el punto que más se asemeja. Para medir la similitud se usan varios tipos de métricas. La más usada se basa en una correlación normalizada entre las vecindades de los puntos de la línea epipolar derecha y la vecindad del punto característico en la imagen izquierda, tal y como se observa en la figura 2.18. A dicha vecindad se le conoce como la ventana de correlación y es un cuadrado de varios píxeles con centro en el punto que se está comparando. El punto más similar es aquel punto de la línea epipolar derecha cuyo valor de correlación es un máximo local (Hartley y Zisserman, 2004; Bradski y Kaehler, 2008; Szeliski, 2011) y, por lo tanto, se establece una correspondencia estereoscópica entre dicho punto en la imagen derecha (\mathbf{h}'_r) y el punto característico en la imagen izquierda (\mathbf{h}'_l). La correspondencia se representa como el par de puntos dado por $(\mathbf{h}'_l, \mathbf{h}'_r)$ y quiere decir que ambos

puntos bidimensionales representan la proyección de un mismo punto tridimensional en el espacio, \mathbf{H} , pero en imágenes diferentes, como se aprecia en la figura 2.18. Además, el píxel dado por \mathbf{h}'_r es también un punto característico, pero en la imagen derecha.

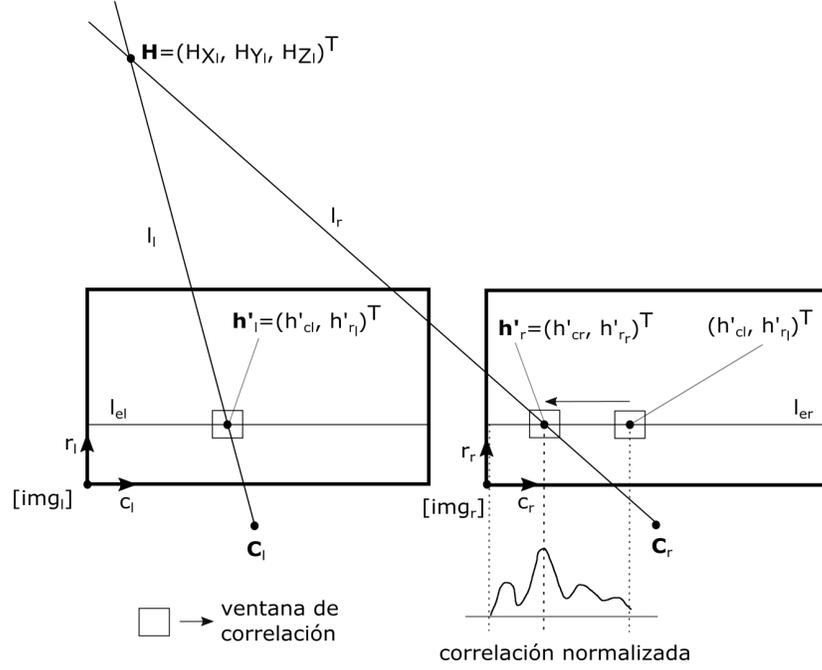


Figura 2.18: Establecimiento de correspondencias estereoscópicas (Bradski y Kaehler, 2008).

Una vez que se conocen las proyecciones del punto \mathbf{H} sobre la imagen derecha \mathbf{h}'_r y la imagen izquierda \mathbf{h}'_i , se debe de realizar una transformación del sistema de coordenadas de la imagen en píxeles $(c, r)^T$, al sistema de coordenadas del plano de la cámara en unidades métricas $(u, v)^T$, para obtener las proyecciones \mathbf{h}_r y \mathbf{h}_i sobre los respectivos planos de la cámara. Para ello, se puede hacer uso de la ecuación 2.49 en la cual se puede despejar las componentes h_u y h_v para obtener una nueva expresión que permite convertir un punto de la imagen a un punto en el plano de la cámara:

$$\mathbf{h} = \begin{pmatrix} h_u \\ h_v \end{pmatrix} = \begin{pmatrix} \frac{d_u}{N_u} \cdot (h'_c - \frac{w}{2}) \\ d_v \cdot (\frac{h}{2} - h'_r) \end{pmatrix} \quad (2.52)$$

La ecuación anterior se aplica a las proyecciones \mathbf{h}'_r y \mathbf{h}'_i :

$$\mathbf{h}_r = \begin{pmatrix} h_{u_r} \\ h_{v_r} \end{pmatrix} = \begin{pmatrix} \frac{d_u}{N_u} \cdot (h'_{c_r} - \frac{w}{2}) \\ d_v \cdot (\frac{h}{2} - h'_{r_r}) \end{pmatrix} \quad (2.53)$$

$$\mathbf{h}_l = \begin{pmatrix} h_{u_l} \\ h_{v_l} \end{pmatrix} = \begin{pmatrix} \frac{d_u}{N_u} \cdot (h'_{c_l} - \frac{w}{2}) \\ d_v \cdot (\frac{h}{2} - h'_{r_l}) \end{pmatrix} \quad (2.54)$$

La expresión 2.53 permite calcular la proyección \mathbf{h}_r respecto al plano de la cámara derecha a partir de la respectiva proyección \mathbf{h}'_r respecto a la imagen derecha. Análogamente, con la expresión 2.54 se puede calcular \mathbf{h}_l a partir de \mathbf{h}'_l .

Finalmente, se puede utilizar la técnica de triangulación estereoscópica para determinar las componentes del punto \mathbf{H} a partir de sus proyecciones en los planos de ambas cámaras (izquierda y derecha). Para ello, considérese el esquema de la figura 2.19, en esta se puede apreciar que los puntos característicos bidimensionales en la imagen izquierda y derecha que forman la correspondencia estereoscópica dada por $(\mathbf{h}_l, \mathbf{h}_r)$, solo difieren en el valor de la coordenada horizontal u . A esa diferencia se le llama disparidad y permite determinar las tres componentes de \mathbf{H} , siempre y cuando se conozca la separación entre cámaras B y la distancia focal f (Martínez, 2019b; Bradski y Kaehler, 2008; Pajares y de la Escalera, 2016).

Si las componentes del punto \mathbf{H} referidas a la cámara izquierda son $(H_{X_l}, H_{Y_l}, H_{Z_l})^T$, entonces, sus componentes referidas a la cámara derecha se pueden encontrar mediante la ecuación de pose dada por 2.22 y las expresiones 2.50 y 2.51, de tal forma que se obtiene $(H_{X_l} + B, H_{Y_l}, H_{Z_l})^T$ (Bradski y Kaehler, 2008; Pajares y de la Escalera, 2016). Teniendo esto en cuenta y usando la técnica de triángulos semejantes en la figura 2.19, se puede llegar a una expresión para h_{u_r} y otra para h_{u_l} (Martínez, 2019b; Bradski y Kaehler, 2008):

$$h_{u_r} = f \frac{H_{X_l}}{H_{Z_l}} \quad (2.55)$$

$$h_{u_l} = f \frac{H_{X_l} + B}{H_{Z_l}} \quad (2.56)$$

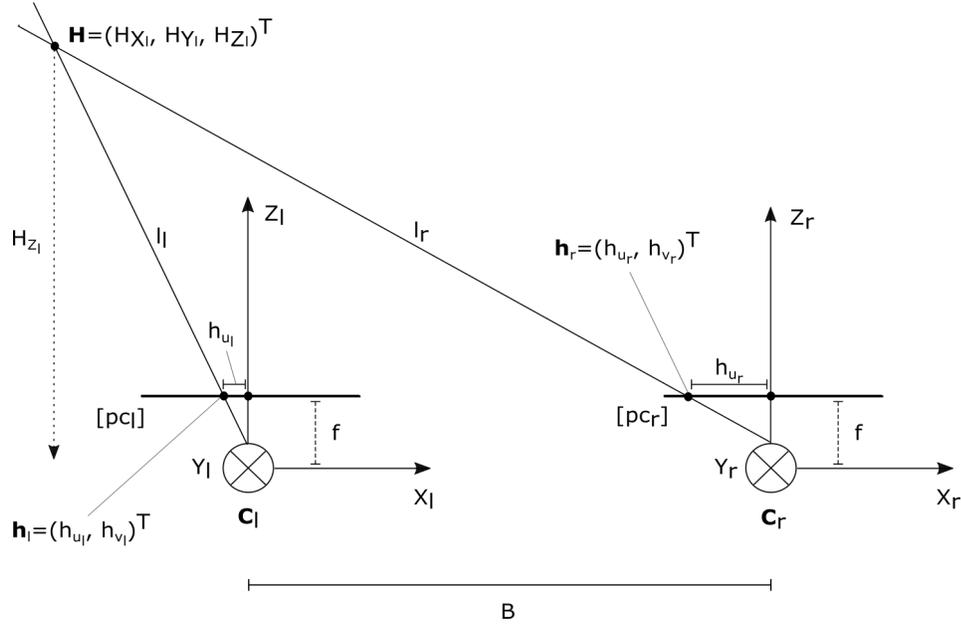


Figura 2.19: Triangulación estereoscópica de un punto de la escena (Martínez, 2019b).

Eliminado la componente H_{X_l} de las expresiones anteriores (Martínez, 2019b; Bradski y Kaehler, 2008):

$$H_{Z_l} = f \frac{B}{h_{u_l} - h_{u_r}} = f \frac{B}{d} \quad (2.57)$$

donde d es la disparidad.

La expresión 2.57 permite calcular la componente de la profundidad del punto \mathbf{H} . Las demás componentes, H_{X_l} y H_{Y_l} , se pueden obtener mediante el uso de la ecuación de proyección dada por 2.45, en la cual se puede despejar H_X y H_Y para obtener una nueva expresión que permite determinar las componentes X y Y de un punto 3D a partir de su proyección sobre el plano de la cámara:

$$\begin{pmatrix} H_X \\ H_Y \end{pmatrix} = \frac{1}{f} \begin{pmatrix} h_u \cdot H_Z \\ h_v \cdot H_Z \end{pmatrix} \quad (2.58)$$

La ecuación anterior se aplica a las componentes h_{u_l} y h_{v_l} de la proyección \mathbf{h}_l sobre el plano de la cámara izquierda:

$$\begin{pmatrix} H_{X_l} \\ H_{Y_l} \end{pmatrix} = \frac{1}{f} \begin{pmatrix} h_{u_l} \cdot H_{Z_l} \\ h_{v_l} \cdot H_{Z_l} \end{pmatrix} \quad (2.59)$$

Finalmente, la posición tridimensional de un punto de la escena se puede obtener al combinar la expresión 2.57 y la expresión 2.59 (Martínez, 2019b; Bradski y Kaehler, 2008):

$$\mathbf{H} = \begin{pmatrix} H_{X_l} \\ H_{Y_l} \\ H_{Z_l} \end{pmatrix} = \frac{B}{h_{u_l} - h_{u_r}} \begin{pmatrix} h_{u_l} \\ h_{v_l} \\ f \end{pmatrix} \quad (2.60)$$

que depende de las componentes las proyecciones de \mathbf{H} sobre el plano de la cámara derecha e izquierda, dadas por las expresiones 2.53 y 2.54, respectivamente.

2.2.4 Obtención de una nube de puntos por estereoscopia

Una nube de puntos tridimensionales o 3D es una estructura muy básica que se utiliza para representar una escena o una superficie en el espacio. Una nube de puntos se denota como \mathbf{N} y se puede definir de una manera muy sencilla como un conjunto de puntos tridimensionales, dados por $\mathbf{P}_i = (x_i, y_i, z_i)^T$ y referidos a un mismo sistema de coordenadas (Pajares y de la Escalera, 2016).

En la sección anterior se explicó como es que se puede adquirir la información de profundidad de un punto de la escena real, a partir de sus proyecciones en las imágenes capturadas por una cámara estereoscópica que esta rectificada y calibrada. Por lo tanto, para obtener una nube de puntos \mathbf{N} que representa la escena completa, basta con repetir todo el procedimiento anterior para un conjunto de correspondencias estereoscópicas dadas por $(\mathbf{h}'_{li} = (h'_{c_{li}}, h'_{r_{li}})^T, \mathbf{h}'_{ri} = (h'_{c_{ri}}, h'_{r_{ri}})^T)$ que se establecen entre los puntos característicos en la imagen izquierda y los píxeles en la imagen derecha, respectivamente. Además, cada uno de los pares $(\mathbf{h}'_{li}, \mathbf{h}'_{ri})$ representan la proyección en la imagen izquierda y derecha, respectivamente, de un punto tridimensional en el espacio, dado por $\mathbf{H}_i = (H_{X_{li}}, H_{Y_{li}}, H_{Z_{li}})^T$.

Cabe destacar que los puntos tridimensionales dados por $\mathbf{H}_i = (H_{X_{li}}, H_{Y_{li}}, H_{Z_{li}})^T$ se pueden representar de una forma simplificada como $\mathbf{H}_i(X_{li}, Y_{li}, Z_{li})$. Análogamente, los puntos bidimensionales dados por $\mathbf{h}'_{li} = (h'_{c_{li}}, h'_{r_{li}})^T$ y $\mathbf{h}'_{ri} = (h'_{c_{ri}}, h'_{r_{ri}})^T$ se puede representar como $\mathbf{h}'_{li}(c_{li}, r_{li})$ y $\mathbf{h}'_{ri}(c_{ri}, r_{ri})$.

Entonces, como se aprecia en la figura 2.20, la nube \mathbf{N} es simplemente un conjunto de puntos $\mathbf{H}_i(X_{li}, Y_{li}, Z_{li})$ de los cuales se conoce sus posiciones tridimensionales respecto al sistema de coordenadas de la cámara izquierda y, además, cada uno de estos puntos está asociado a un único punto característico $\mathbf{h}'_{li}(c_{li}, r_{li})$ en la imagen izquierda, que son sus proyecciones sobre dicha imagen.

En la figura 2.20 también se puede ver que hay puntos de la escena, del espacio 3D que está siendo captado por la cámara, de los cuales se desconoce su posición tridimensional, esto es debido a que no tienen un punto característico asociado, es decir, la proyección de dichos puntos sobre la imagen izquierda

es un simple píxel que no fue clasificado como punto característico y por lo tanto, no se puede realizar el procedimiento de estimación de profundidad.

En este caso, los píxeles que conforman la imagen izquierda, sean puntos característicos o no, se puede representar de una forma simplificada como $\mathbf{p}_{l_j}(c_{l_j}, r_{l_j})$ y por lo tanto, se podría decir que los puntos característicos \mathbf{h}'_{li} son un subconjunto del conjunto formado por los píxeles \mathbf{p}_{l_j} .

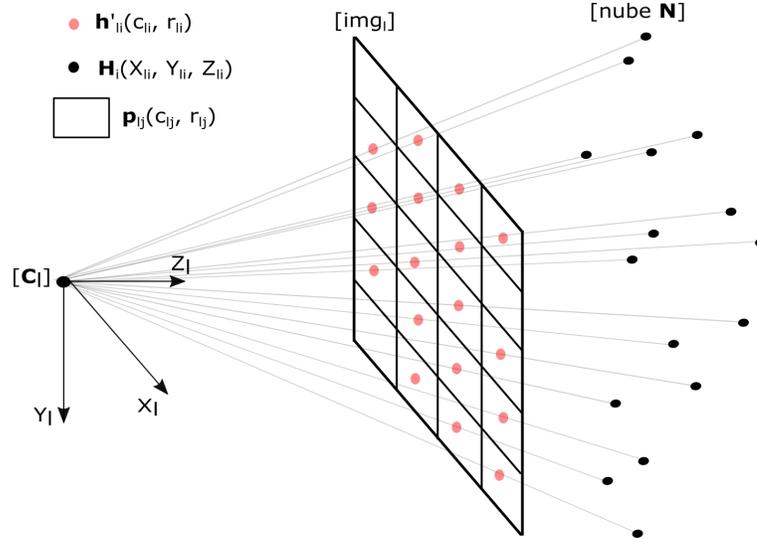


Figura 2.20: Nube de puntos respecto al sistema de coordenadas de la cámara izquierda.

2.2.5 Obtención de una nube de puntos usando el Sistema Operativo Robótico (ROS)

El Sistema Operativo Robótico (ROS¹) es un marco o plataforma flexible para escribir software para robots. Es una recopilación de bibliotecas, herramientas y convenciones cuyo fin es simplificar aquellas tareas que implican la creación de un comportamiento robótico complejo y robusto. La principal característica es que permite la comunicación e interacción entre procesos, conocidos como nodos (Koubaa, 2016).

Para obtener la representación más básica de una escena real, es decir, para obtener la nube de puntos \mathbf{N} a partir de una cámara estereoscópica en configuración canónica, se utilizan tres nodos de ROS de código abierto (Massot, 2013; Torres, 2020; Rabaud, 2016):

¹Un mayor desarrollo sobre el funcionamiento de ROS se puede encontrar en el apéndice A.

1. *camera1394stereo*: es el encargado de capturar o tomar las imágenes crudas de intensidad de la cámara izquierda y de la cámara derecha. En caso de que las cámaras solo puedan capturar imágenes RGB, este nodo se encarga también de calcular las respectivas imágenes de intensidad.
2. *camera_calibration*: toma las imágenes crudas de intensidad proporcionadas por el nodo anterior y les aplica el método de calibración de Zhang (Zhang et al., 1998), para determinar los parámetros intrínsecos y extrínsecos de las cámaras (izquierda y derecha) como distancias focales, orientación, posición y el factor de distorsión. Estos parámetros se guardan en una carpeta interna de ROS llamada *camera_info*. Específicamente, se guardan los parámetros de calibración de la cámara izquierda y de la cámara derecha con los siguientes nombres: “GUID”_left.yaml y “GUID”_right.yaml respectivamente. El GUID (del inglés Globally Unique Identifier) es un código de 128 bits único en todo el espacio y universo conocido por el hombre, existe un GUID para la cámara izquierda que es único y otro para la cámara derecha que también es único.
3. *stereo_image_proc*: toma los archivos que contienen los parámetros de calibración de la cámara izquierda y derecha, que fueron creados por el nodo anterior y guardados en la carpeta interna llamada *camera_info*. Utiliza la información dentro de dichos archivos para calibrar y rectificar las imágenes crudas de intensidad. Luego empieza a establecer correspondencias estereoscópicas entre los puntos característicos en la imagen de intensidad izquierda rectificada y calibrada y los píxeles en la imagen de intensidad derecha, también rectificada y calibrada. Posteriormente busca la posición tridimensional de todos los puntos característicos respecto al sistema de coordenadas de la cámara izquierda, mediante triangulación. Es de esta forma como finalmente se obtiene la nube de puntos N .

En la figura 2.21 se muestra un diagrama que describe del proceso anterior.

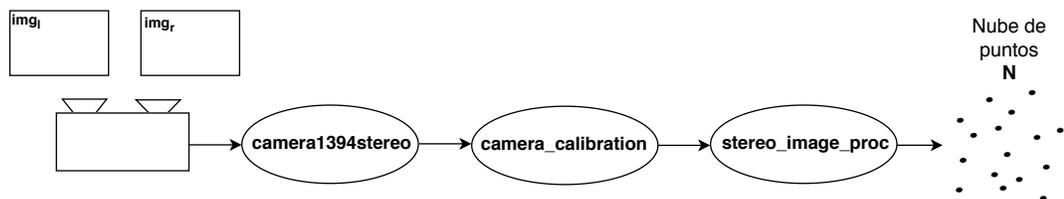


Figura 2.21: Implementación de nodos de ROS para obtener una nube de puntos.

3 Algoritmo de odometría visual estereoscópico

3.1. Definición y algoritmo general

La odometría visual estereoscópica es una técnica bastante madura que se utiliza para estimar el movimiento de un robot móvil en los seis grados de libertad que son posibles en un espacio 3D: tres componentes para la traslación $\Delta\mathbf{T}$ y tres componentes para la rotación $\Delta\mathbf{W}$, aunque usualmente esta última se representa con la matriz de rotación $\Delta\mathbf{R}$, en lugar de los ángulos que describen el cambio de orientación. Dicha estimación se realiza a partir de las imágenes de intensidad capturadas por una cámara estereoscópica, imagen izquierda y derecha, antes y después del movimiento del robot y que se encuentra atada rígidamente a un costado de este, con una inclinación hacia la superficie por la cual se mueve. Entonces, un algoritmo de odometría visual estereoscópico utiliza dos pares consecutivos de imágenes, obtenidos con una cámara estereoscópica en un instante de tiempo t_k antes del movimiento y en un instante t_{k+1} después del movimiento, para estimar el movimiento del robot entre esos dos instantes de tiempo consecutivos (Arun et al., 2007; Liu et al., 2018; Gao, 2016).

La técnica anterior está bien establecida en robots terrestres como un medio para incrementar la precisión de localización (se puede llegar a obtener hasta un $<1\%$ de la distancia recorrida) y tener en cuenta los errores que se generan con la odometría de ruedas a bordo, cuando las ruedas deslizan en la superficie. Además, la odometría visual estereoscópica ha demostrado tener una gran importancia y utilidad en los robots de exploración espacial (astro-móviles o rovers), no solamente porque mejora la precisión de localización, también aumenta la eficiencia de conducción y los vuelve más seguros (Gao, 2016).

Su importancia radica en que sirve para compensar los errores de la odometría clásica que en ocasiones es engañada por el deslizamiento de las ruedas, que genera un cálculo erróneo del movimiento del robot, debido a que el conteo de las revoluciones de las ruedas se ve alterado. También sirve para compensar la falta de información proporcionada por un GPS, que se puede generar cuando el robot está operando en un área donde no se pueden recibir transmisiones GPS o esté operando en un planeta diferente a la Tierra que no tiene equipado un conjunto de satélites, como Marte (Arun et al., 2007).

Un algoritmo de odometría visual estereoscópico sigue el esquema general que se muestra en la figura 3.1, el cual está compuesto por varias etapas o pasos generales (Matthies, 1989; Arun et al., 2007; Maimone et al., 2007; Gao, 2016):

1. Detección de puntos característicos: primero se captura un par de imágenes con la cámara estereoscópica, antes de que se mueva el robot. Luego, se selecciona sobre la imagen izquierda un conjunto de puntos característicos que se puedan encontrar fácilmente entre imágenes tomadas desde distintas posiciones (imagen izquierda y derecha antes del movimiento) y rastrear entre imágenes tomadas en distintos instantes de tiempo (imagen izquierda antes del movimiento e imagen izquierda después del movimiento). Para ello, se aplica a la imagen izquierda algún operador de interés (Forstner, Harris, Bayer, entre otros) sintonizado para la detección de esquinas y se seleccionan los píxeles con los valores de interés más altos.
2. Establecimiento de correspondencias estereoscópicas: se establecen las correspondencias estereoscópicas entre puntos característicos en la imagen izquierda y píxeles en la imagen derecha. Debido a que se supone que la cámara estereoscópica está bien calibrada, un punto en la imagen derecha corresponderá a un punto característico en la imagen izquierda, si este se encuentra a lo largo de la línea epipolar derecha y es el punto más similar. Se utiliza algún método basado en la correlación normalizada para determinar los puntos con la mayor similitud, que es aquel píxel de la imagen derecha cuyo valor de intensidad y los valores de intensidad vecinos sean lo más parecidos al valor de intensidad del punto característico en la imagen izquierda y sus vecinos. Posteriormente, a partir de las correspondencias estereoscópicas, se calculan las posiciones tridimensionales de los puntos característicos, mediante triangulación estereoscópica. De esta forma, se obtiene una nube de puntos antes del movimiento.
3. Establecimiento de correspondencias temporales: luego de que el robot se mueva una pequeña distancia, se captura un segundo par de imágenes con la cámara estereoscópica. Posteriormente, las posiciones tridimensionales que conforman la nube puntos antes del movimiento, se proyectan en la imagen izquierda del segundo par de imágenes, utilizando el movimiento aproximado proporcionado por la odometría de ruedas a bordo. Luego, usando nuevamente un método basado en la correlación normalizada, se refinan las posiciones de las proyecciones anteriores, que consiste en rastrear los puntos característicos de la imagen izquierda antes del movimiento, dentro de una ventana de búsqueda en la imagen izquierda

después del movimiento, con centro en la respectiva proyección. Dicho rastreo es un establecimiento de correspondencias temporales 2D entre los puntos característicos en la imagen izquierda antes del movimiento y los píxeles alrededor de las posiciones proyectadas en la imagen izquierda después del movimiento. Seguidamente, se realiza un establecimiento de correspondencias estereoscópicas sobre el segundo par de imágenes, exactamente como en el paso anterior. Por último, de las correspondencias establecidas en el segundo par de imágenes, se calcula la segunda nube de puntos después del movimiento, mediante la triangulación estereoscópica. Cabe destacar que como se realizó un establecimiento de correspondencias temporales 2D entre puntos de la imagen izquierda antes y después del movimiento, implica que indirectamente se conocen las correspondencias temporales 3D entre la primera y la segunda nube de puntos, ya que estas son las posiciones tridimensionales de las correspondencias temporales 2D.

4. Estimación inicial: se encuentra la traslación y rotación con la que se debe de mover la primera nube de puntos para que esta coincida con la segunda nube de puntos, mediante el uso de algún método que minimice la suma de los cuadrados sopesados de las distancias de las correspondencias temporales 3D clasificadas como típicas entre un subconjunto de la nube de puntos ante del movimiento transformada y un subconjunto de la nube de puntos después del movimiento. Los factores pesos escalares asociados a cada distancia representan la calidad de los puntos tridimensionales de cada correspondencia y usualmente se obtienen a partir del modelo 3D del error de posición de los puntos tridimensionales. Además, las correspondencias temporales típicas se encuentran al aplicar algún método iterativo (RANSAC, MSAC, MLESAC o ASRC) a la estimación anterior, para varios subconjuntos de correspondencias. La ventaja de esta estimación es que es simple, rápida y robusta. La desventaja es que al considerar indirectamente el modelo 3D del error de posición en una variable escalar que son los factores de peso, sus resultados son inferiores a aquellos métodos en los que si se considera directamente el modelo 3D del error de posición en una matriz 3D.
5. Estimación precisa: se mejora la precisión de los parámetros de movimiento estimados en el paso anterior, mediante la maximización de alguna función que considere e incorpore directamente el modelo 3D del error de posición de los puntos tridimensionales en una matriz 3D, en lugar de considerarlo e incorporarlo indirectamente en el calculo de un peso escalar que depende de este, como en el paso anterior. Además, dicha función también depende del error residual de las correspondencias

temporales 3D clasificadas como típicas y de los parámetros inicialmente estimados en el paso anterior, los cuales se consideran como valores iniciales.

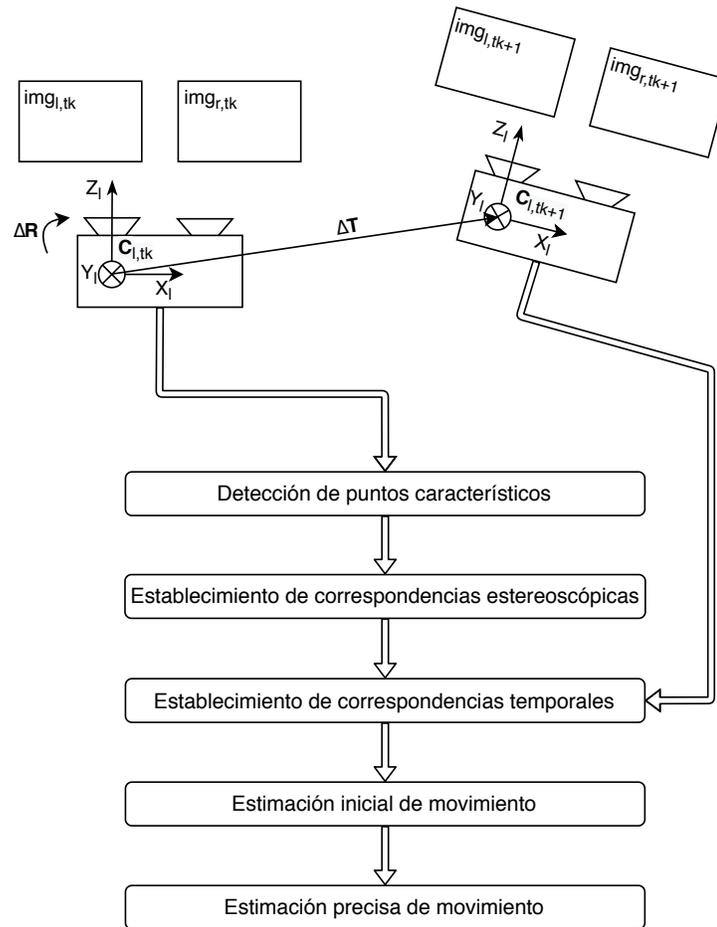


Figura 3.1: Esquema general de un algoritmo de odometría visual estereoscópico.

Dichas etapas son exactamente las mismas para cualquier algoritmo de odometría visual estereoscópico, lo que cambia son los métodos, modelos y técnicas matemáticas utilizadas para llevarlas a cabo. Por ejemplo, el algoritmo de odometría visual estereoscópico de (Matthies, 1989), que fue implementado en tiempo real en los rovers *Spirit* y *Opportunity* de la Misión MER, lleva a cabo los pasos anteriores de la siguiente forma:

1. Detección de puntos característicos: emplea el operador de interés de Forstner para la detección de esquinas (puntos característicos).
2. Establecimiento de correspondencias estereoscópicas: utiliza la pseudo-correlación normalizada para establecer las correspondencias estereoscópicas en el primer par de imágenes.
3. Establecimiento de correspondencias temporales: vuelve a utilizar la pseudo-correlación normalizada para establecer las correspondencias temporales 2D entre los puntos característicos en la imagen izquierda antes del movimiento y los píxeles alrededor de las posiciones proyectadas en la imagen izquierda después del movimiento (rastrear puntos característicos). También utiliza la pseudo-correlación normalizada para establecer las correspondencias estereoscópicas en el segundo par de imágenes.
4. Estimación inicial: minimiza la suma de los cuadrados sopesados de las distancias de correspondencias temporales 3D clasificadas como típicas entre subconjunto transformado (rotado y trasladado) de la nube antes del movimiento y el respectivo subconjunto de la nube después del movimiento, haciendo uso de un método basado en la descomposición en valores singulares de una matriz 3x3. Además, los pesos escalares se derivan de los determinantes de las matrices de covarianza 3D de un modelo Gaussiano 3D de los errores posición de los puntos tridimensionales y las correspondencias temporales típicas se encuentran mediante el método de RANSAC.
5. Estimación precisa: se maximiza una función de probabilidad condicional alrededor de los parámetros de movimiento inicialmente estimados en la parte a), la cual depende del error residual entre correspondencias temporales 3D clasificadas como típicas y de las matrices 3D de covarianza de dichos errores residuales, las cuales se obtienen directamente a partir de las matrices de covarianza 3D de un modelo Gaussiano 3D de los errores de posición de los puntos tridimensionales que conforman cada correspondencia.

Por otro lado, si la nube de puntos \mathbf{N}_{t_k} antes del movimiento y la nube de puntos $\mathbf{N}_{t_{k+1}}$ después del movimiento son proporcionadas por algún algoritmo o proceso externo, junto con las respectivas imágenes capturadas en cada instante, como el algoritmo de la figura 2.21 de la sección 2.2.5, que utiliza varios nodos de ROS para obtener nubes de puntos a partir de un par de imágenes capturadas por una cámara estereoscópica, entonces los pasos generales del algoritmo de odometría visual estereoscópico se reducen al esquema de la figura 3.2 y se convierte en un proceso más simple. De hecho, el paso de establecimiento de correspondencias temporales solo se reduce a eso, no se tiene

que encontrar la segunda nube de puntos y tampoco se tienen que establecer las correspondencias estereoscópicas en el segundo par de imágenes.

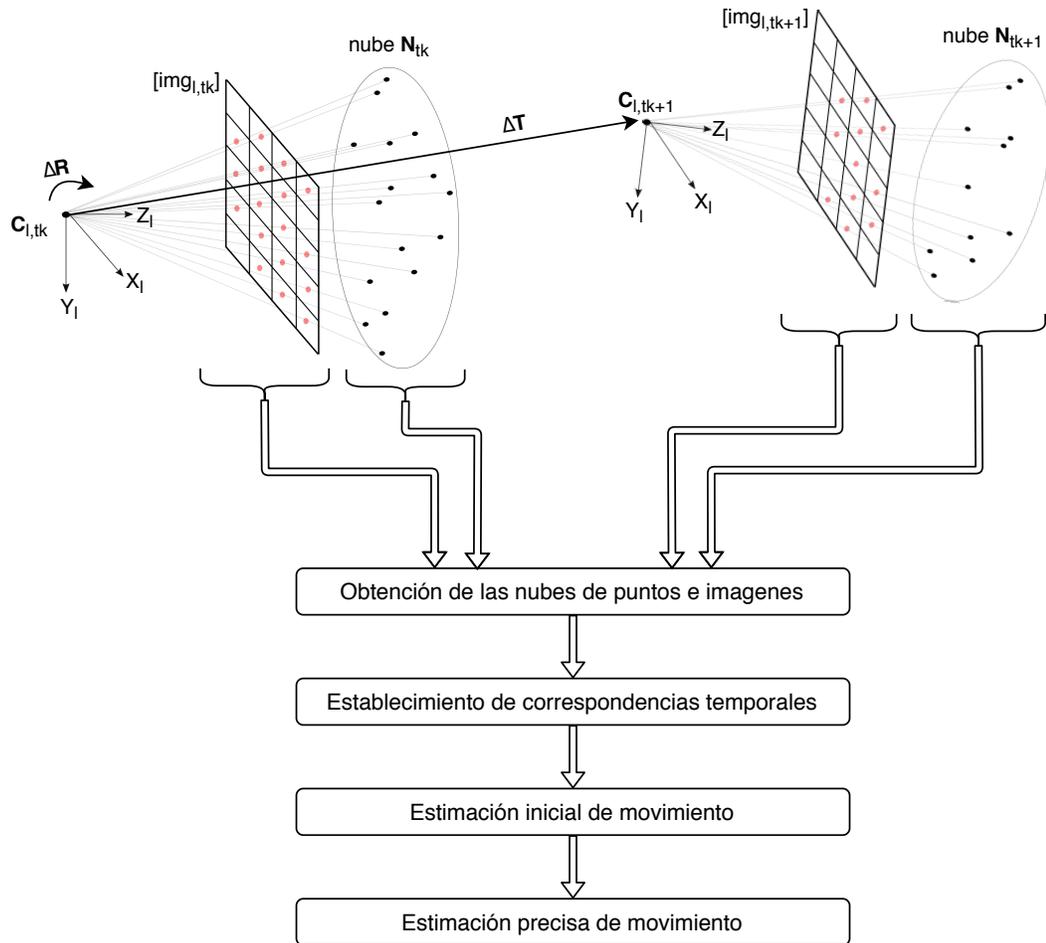


Figura 3.2: Esquema general de un algoritmo de odometría visual estereoscópico, donde las nubes de puntos son proporcionadas por un proceso externo.

En las siguientes secciones se describe y se explica de forma detallada la primera parte del algoritmo de la figura 3.2, los pasos del 1 al 3, utilizando los métodos empleados por Larry Matthies en su algoritmo de odometría visual estereoscópico (Matthies, 1989), para establecer correspondencias temporales y realizar una estimación inicial del movimiento, a partir de la imagen izquierda y la nube de punto antes y después del movimiento del robot, proporcionadas a través de un algoritmo externo basado en ROS. Cabe destacar que no se considera el error de posición que afecta a los puntos característicos y por consiguiente, a los respectivos puntos tridimensionales, el cual si es modelado

y contemplado por Larry Matthies en su algoritmo.

3.2. Obtención de las nubes de puntos e imágenes

Se quiere obtener la nube de puntos y la imagen izquierda antes y después del movimiento de un astromóvil, mediante algún proceso externo basado en ROS, de tal forma que estas sean las entradas, datos iniciales o el punto de partida del algoritmo de odometría visual estereoscópico y que este pueda acceder a dichas entradas de una forma fácil y sencilla, es decir, que la comunicación entre el proceso externo y el algoritmo de odometría sea simple, que este pueda acceder sin ninguna complicación a los mensajes publicados por el proceso externo, que contienen las nubes de puntos e imágenes. Para ello, se emplea el proceso basado en ROS que fue descrito en la sección 2.2.5 del capítulo anterior, el cual utiliza tres nodos de código abierto para obtener las imágenes capturadas por una cámara estereoscópica y calcular la nube de puntos correspondiente.

El protocolo para obtener las nubes de puntos e imágenes proporcionadas por el proceso anterior, es bastante sencillo. Antes de que el rover se mueva, el algoritmo de odometría visual estereoscópico se suscribe a los tópicos¹ de ROS llamados */points2* y */image_rect*, para recibir, procesar y almacenar el mensaje que contiene la nube de puntos \mathbf{N}_{t_k} y el mensaje que contiene la imagen izquierda img_{l,t_k} antes del movimiento, que fueron publicados por los nodos de ROS en los respectivos tópicos. Análogamente, luego de que el astromóvil se mueva una pequeña distancia, el algoritmo de odometría se vuelve a suscribir a los tópicos mencionados anteriormente, para obtener la nube de puntos $\mathbf{N}_{t_{k+1}}$ y la imagen izquierda $img_{l,t_{k+1}}$ después del movimiento. La otra alternativa es que el algoritmo de odometría se suscriba a los tópicos solamente una vez, después del movimiento del rover, ya que los tópicos pueden almacenar varios mensajes y entonces, de esta forma se obtiene la imagen izquierda y la nube de puntos antes y después del movimiento en un mismo instante. El protocolo anterior se resume en la figura 3.3.

Tal y como aprecia en la figura 3.3, cada uno de los puntos tridimensionales $\mathbf{H}_{i,t_k}(X_{li}, Y_{li}, Z_{li})$ que conforman la nube de puntos \mathbf{N}_{t_k} antes del movimiento, representan la posición tridimensional de un punto característico único $\mathbf{h}'_{i,t_k}(c_{li}, r_{li})$ en la imagen izquierda img_{l,t_k} antes del movimiento. Además, la cantidad de puntos tridimensionales que tiene la nube \mathbf{N}_{t_k} es conocida y se denota como $C_{N_{t_k}}$, la cual es exactamente igual a la cantidad de puntos característicos que el proceso externo basado en ROS seleccionó sobre la imagen img_{l,t_k} . Entonces, el valor de la variable i , que representa el i -ésimo punto

¹En el apéndice A se explica detalladamente como es que los nodos del proceso descrito en la sección 2.2.5 del capítulo 2, envían información a través de la publicación de mensajes en distintos tópicos y como se puede recibir dicha información.

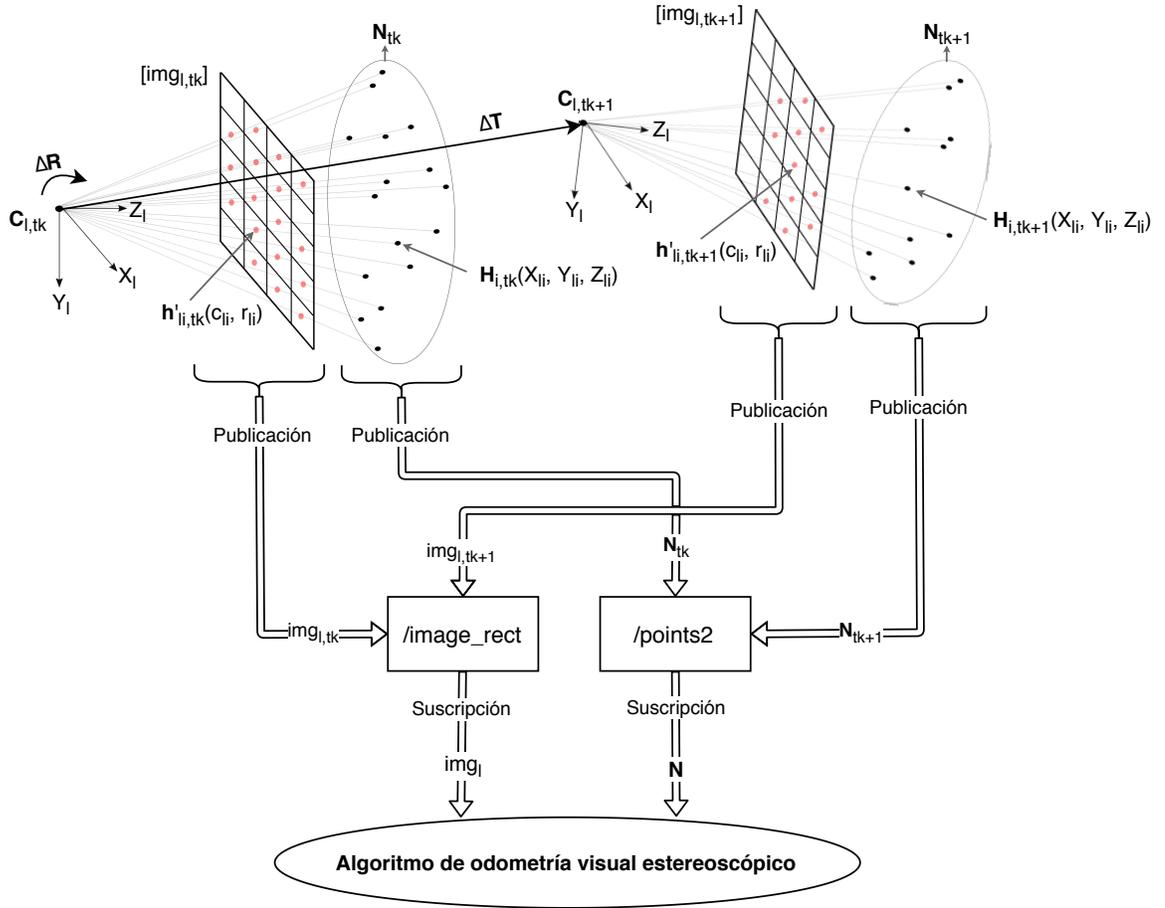


Figura 3.3: Protocolo para obtener las nubes de puntos e imágenes antes y después del movimiento, proporcionadas a través de tópicos de ROS mediante un proceso externo.

tridimensional y el i -ésimo punto característico en el instante t_k , se encuentra dentro del siguiente intervalo:

$$1 \leq i \leq C_{N_{t_k}} \quad (3.1)$$

Por otro lado, cada uno de los píxeles $\mathbf{p}_{l_j,t_k}(c_{l_j}, r_{l_j})$ de la imagen img_{l,t_k} , tienen un valor de intensidad asociado, el cual es conocido y está dado por I_{l_j,t_k} , como se aprecia en la figura 3.5. Además, la cantidad de píxeles que tiene una imagen es igual al largo de esta por su ancho ($w \cdot h$), por lo que el valor de la variable j , que representa el j -ésimo píxel y el j -ésimo valor de intensidad en el instante t_k , puede variar dentro de este intervalo:

$$1 \leq j \leq w \cdot h \tag{3.2}$$

Como se observa en la figura 3.4, cada uno de los píxeles \mathbf{p}_{l_j,t_k} y sus valores de intensidad I_{l_j,t_k} se numeran, es decir, se les asigna un valor de j , de acuerdo al orden de aparición en la imagen img_{l,t_k} , de izquierda a derecha y de abajo a arriba. Como los píxeles se numeran de esa forma, se puede relacionar el valor de la variable j con las respectivas componentes (c_{l_j}, r_{l_j}) del píxel mediante la expresión 3.3, tal y como se aprecia en la figura 3.5.

$$j = c_{l_j} + (r_{l_j} - 1)w \tag{3.3}$$

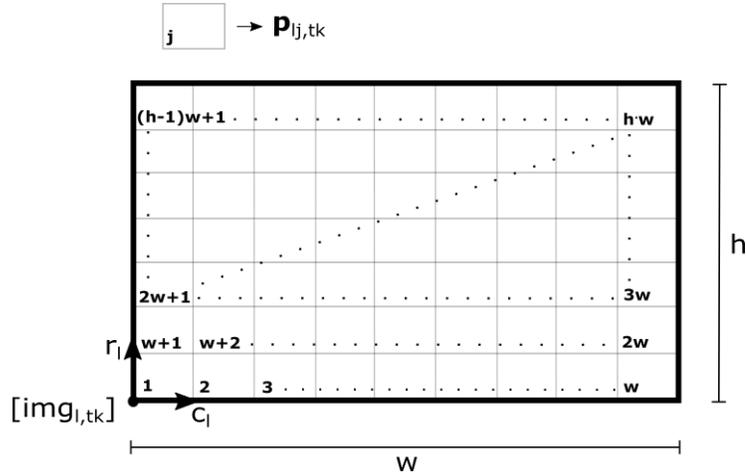


Figura 3.4: Numeración de cada uno de los píxeles de la imagen izquierda antes del movimiento.

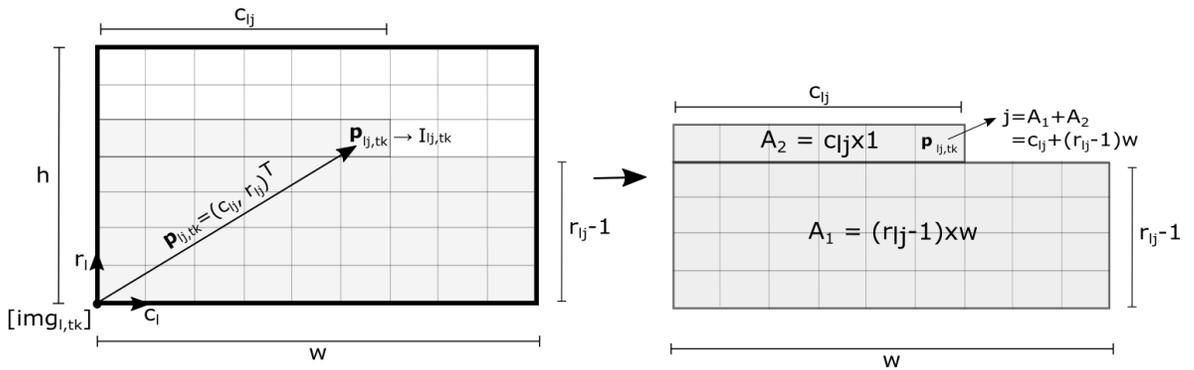


Figura 3.5: J-ésimo píxel de la imagen de intensidad izquierda antes del movimiento.

Los puntos característicos $\mathbf{h}'_{li,tk}$ se numeran de forma análoga a los píxeles, ver la figura 3.6. Además, dichos puntos representan un subconjunto de tamaño $C_{N_{tk}}$, del conjunto formado por los píxeles $\mathbf{p}_{lj,tk}$ de la imagen $img_{l,tk}$. Esto quiere decir que un punto característico dado por $\mathbf{h}'_{li,tk}$ es exactamente igual al píxel $\mathbf{p}_{lj=p_i,tk}$ y por lo tanto, tiene un valor de intensidad que es igual a $I_{lj=p_i,tk}$, como se puede ver en la figura 3.7, donde p_i se obtiene al evaluar la expresión 3.3 en las componentes (c_{li}, r_{li}) del punto característico:

$$p_i = c_{li} + (r_{li} - 1)w \quad (3.4)$$

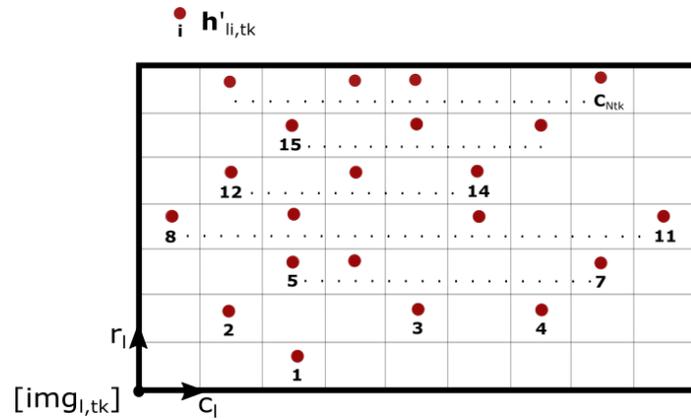


Figura 3.6: Numeración de cada uno de los puntos característicos en la imagen de izquierda antes del movimiento.

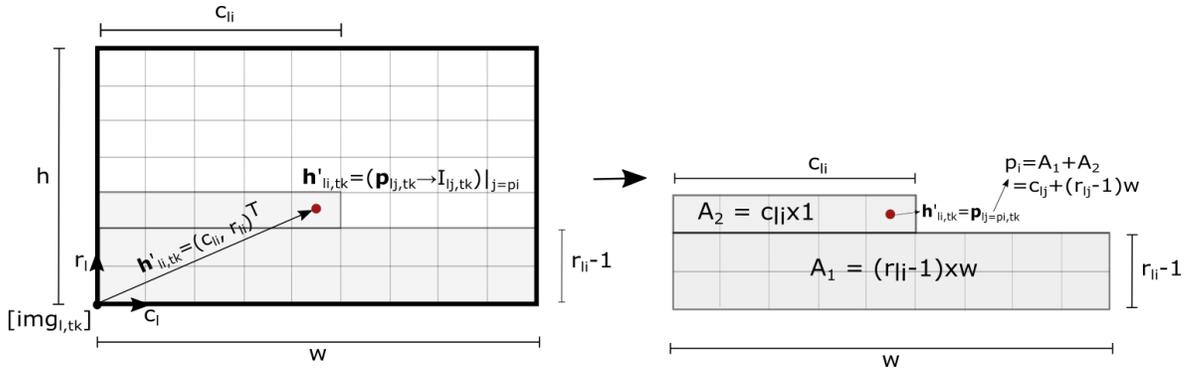


Figura 3.7: I-ésimo punto característico en la imagen de intensidad izquierda antes del movimiento.

Además, a cada píxel $\mathbf{p}_{lj,tk}$ se le asocia una variable denotada como pc_j , a la cual se le asigna un valor igual a cero cuando el píxel no es un punto

característico y un valor igual a i cuando el píxel es un punto característico dado por \mathbf{h}'_{li,t_k} . Entonces, dado un píxel \mathbf{p}_{lj,t_k} , a partir del valor asociado de pc_j se puede determinar si este no es punto característico, cuando $pc_j = 0$ o si es un punto característico dado por $\mathbf{h}'_{li=pc_j,t_k}$, cuando $pc_j > 0$.

Análogamente, los puntos tridimensionales $\mathbf{H}_{m,t_{k+1}}(X_{lm}, Y_{lm}, Z_{lm})$ que conforman la nube de puntos $\mathbf{N}_{t_{k+1}}$ después del movimiento, representan la posición tridimensional de un punto característico único $\mathbf{h}'_{lm,t_{k+1}}(c_{lm}, r_{lm})$ en la imagen izquierda $img_{l,t_{k+1}}$ después del movimiento, tal y como se aprecia en la figura 3.3. El tamaño de la nube de puntos $\mathbf{N}_{t_{k+1}}$ es conocido y se denota como $C_{N_{t_{k+1}}}$, por lo que, el intervalo que limita el valor de la variable m que representa que el m -ésimo punto tridimensional y el m -ésimo punto característico en el instante t_{k+1} es el siguiente:

$$1 \leq m \leq C_{N_{t_{k+1}}} \quad (3.5)$$

Cada uno de los píxeles $\mathbf{p}_{ln,t_{k+1}}(c_{ln}, r_{ln})$ de la imagen img_{l,t_k} , tienen un valor de intensidad asociado, dado por $I_{ln,t_{k+1}}$. Además, el intervalo de la variable n , que representa el n -ésimo píxel y el n -ésimo valor de intensidad en el instante t_{k+1} , es exactamente igual al intervalo de j :

$$1 \leq n \leq w \cdot h \quad (3.6)$$

Además, los píxeles $\mathbf{p}_{ln,t_{k+1}}$ de la imagen img_{l,t_k} , se enumeran exactamente igual que los píxeles \mathbf{p}_{lj,t_k} de la imagen $img_{l,t_{k+1}}$ y por lo tanto, el valor de la variable n se puede relacionar con las respectivas componentes (c_{ln}, r_{ln}) del píxel, de la misma forma que en la expresión 3.3:

$$n = c_{ln} + (r_{ln} - 1)w \quad (3.7)$$

Los puntos característicos $\mathbf{h}'_{lm,t_{k+1}}$ se enumeran de la misma manera y son un subconjunto de tamaño $C_{N_{t_{k+1}}}$ del conjunto formado por los píxeles $\mathbf{p}_{ln,t_{k+1}}$ de la imagen $img_{l,t_{k+1}}$, de tal forma que un punto característico $\mathbf{h}'_{lm,t_{k+1}}$ es exactamente igual al píxel $\mathbf{p}_{ln=p_m,t_{k+1}}$ y tiene un valor de intensidad dado por $I_{ln=p_m,t_{k+1}}$, donde p_m se obtiene de forma análoga a p_i :

$$p_m = c_{lm} + (r_{lm} - 1)w \quad (3.8)$$

Así mismo, a cada píxel $\mathbf{p}_{ln,t_{k+1}}$ se le asocia una variable denotada como pc_n , la cual toma un valor de cero cuando el píxel no es un punto característico y un valor igual a m cuando el píxel es un punto característico dado por $\mathbf{h}'_{lm,t_{k+1}}$. Entonces, dado un píxel $\mathbf{p}_{ln,t_{k+1}}$, mediante el valor asociado de pc_n se puede determinar si este no es punto característicos, cuando $pc_n = 0$, o si es un punto característico que esta dado por $\mathbf{p}_{lm=pc_n,t_{k+1}}$, cuando $pc_n > 0$.

Por último, cabe destacar que al inicio del algoritmo de odometría visual estereoscópico, solamente se tiene conocimiento de las nubes, imágenes, variables, subconjuntos y conjuntos de puntos y datos descritos y mencionados anteriormente. Eso quiere decir que se desconocen las correspondencias estereoscópicas entre la imagen izquierda y la imagen derecha en el instante t_k y en el instante t_{k+1} , así como las correspondencias temporales 2D entre la imagen img_{l,t_k} y la imagen $img_{l,t_{k+1}}$ y por lo tanto, las correspondencias temporales 3D entre la nube \mathbf{N}_{t_k} y la nube $\mathbf{N}_{t_{k+1}}$. Es por ello que el siguiente paso es establecer las correspondencias temporales, las cuales son imprescindibles para estimar el movimiento del astromóvil. Además, como las nubes de puntos \mathbf{N}_{t_k} y $\mathbf{N}_{t_{k+1}}$ ya son conocidas, no es necesario establecer las correspondencias estereoscópicas, las cuales son útiles solamente en el proceso de triangulación que se emplea para encontrar las nubes de puntos.

3.3. Establecimiento de correspondencias temporales

En este segundo paso se desea establecer las correspondencias temporales 2D entre los puntos característicos \mathbf{h}'_{li,t_k} de la imagen img_{l,t_k} antes del movimiento y los píxeles en la imagen $img_{l,t_{k+1}}$ después del movimiento. En este caso particular, como ya se conocen los puntos característicos $\mathbf{h}'_{lm,t_{k+1}}$ en la imagen $img_{l,t_{k+1}}$, todo se reduce a establecer las correspondencias temporales 2D entre los puntos característicos \mathbf{h}'_{li,t_k} y los puntos característicos $\mathbf{h}'_{lm,t_{k+1}}$. Dichas correspondencias se representan mediante pares de puntos dados por $(\mathbf{h}'_{li,t_k}, \mathbf{h}'_{lm,t_{k+1}})$, de tal forma que los puntos bidimensionales \mathbf{h}'_{li,t_k} y $\mathbf{h}'_{lm,t_{k+1}}$ representan la proyección en la imagen izquierda antes y después del movimiento, respectivamente, de un mismo punto en el espacio. Cabe destacar que si se establecen las correspondencias temporales 2D, también se establecen las correspondencias temporales 3D entre los puntos \mathbf{H}_{i,t_k} de la nube \mathbf{N}_{t_k} y los puntos $\mathbf{H}_{m,t_{k+1}}$ de la nube $\mathbf{N}_{t_{k+1}}$, debido a que como se ha mencionado anteriormente, cada punto de una nube está ligado a un único punto característico y ese enlace es conocido. Entonces, las posiciones tridimensionales de los puntos característicos que conforman las correspondencias temporales 2D $(\mathbf{h}'_{li,t_k}, \mathbf{h}'_{lm,t_{k+1}})$, son las correspondencias temporales 3D que se representan como $(\mathbf{H}_{i,t_k}, \mathbf{H}_{m,t_{k+1}})$, de tal forma que \mathbf{H}_{i,t_k} y $\mathbf{H}_{m,t_{k+1}}$ representan la posición tridimensional antes y después del movimiento, respectivamente, de un mismo punto en el espacio.

Pero resulta que como la nube de puntos $\mathbf{N}_{t_{k+1}}$ es conocida, no es necesario establecer cada uno de las correspondencias temporales 2D que existen entre los puntos característicos \mathbf{h}'_{li,t_k} y los puntos característicos $\mathbf{h}'_{lm,t_{k+1}}$, debido a que eso es útil únicamente cuando se quiere establecer las correspondencias estereoscópicas en el par de imágenes en el instante t_{k+1} , para así poder

determinar la nube $\mathbf{N}_{t_{k+1}}$ mediante triangulación. Además, la estimación de movimiento se puede llevar a cabo sin ningún problema si solamente se tiene una pequeña cantidad de correspondencias temporales 2D y 3D.

Entonces, lo que en realidad se quiere, es establecer las correspondencias temporales 2D entre un pequeño subconjunto formado por los puntos denotados como $\mathbf{q}_{l_s, t_k}(c_{l_s}, r_{l_s})$, los cuales son puntos característicos \mathbf{h}'_{li, t_k} en la imagen img_{l, t_k} , elegidos de forma aleatoria, y un pequeño subconjunto formado por los puntos denotados como $\mathbf{q}_{l_s, t_{k+1}}(c_{l_s}, r_{l_s})$, los cuales son puntos característicos $\mathbf{h}'_{lm, t_{k+1}}$ en la imagen $img_{l, t_{k+1}}$, elegidos cuidadosamente, tal que \mathbf{q}_{l_s, t_k} y $\mathbf{q}_{l_s, t_{k+1}}$ representan la proyección en la imagen img_{l, t_k} y en la imagen $img_{l, t_{k+1}}$, respectivamente, del mismo punto tridimensional en el espacio. Dichas correspondencias temporales 2D se representan mediante los pares de puntos dados por $(\mathbf{q}_{l_s, t_k}, \mathbf{q}_{l_s, t_{k+1}})$. Además, el tamaño de los subconjuntos está dado por C_q , de tal forma que el intervalo de la variable s que define al s -ésimo punto o la s -ésima correspondencia, está dado por:

$$1 \leq s \leq C_q \quad (3.9)$$

Lo anterior implica que también existe un subconjunto formado por los puntos denotados como $\mathbf{Q}_{s, t_k}(X_{l_s}, Y_{l_s}, Z_{l_s})$, los cuales son los puntos tridimensionales \mathbf{H}_{i, t_k} que representan las posiciones tridimensionales de los respectivos puntos característicos \mathbf{h}'_{li, t_k} elegidos aleatoriamente, \mathbf{q}_{l_s, t_k} . Análogamente, existe un subconjunto formado por los puntos denotados como $\mathbf{Q}_{s, t_{k+1}}(X_{l_s}, Y_{l_s}, Z_{l_s})$, los cuales son los puntos tridimensionales $\mathbf{H}_{m, t_{k+1}}$ que representan las posiciones tridimensionales de los respectivos puntos característicos $\mathbf{h}'_{lm, t_{k+1}}$ elegidos cuidadosamente, $\mathbf{q}_{l_s, t_{k+1}}$. Dichos subconjuntos forman las correspondencias temporales 3D que se representan como $(\mathbf{Q}_{s, t_k}, \mathbf{Q}_{s, t_{k+1}})$, tal que \mathbf{Q}_{s, t_k} y $\mathbf{Q}_{s, t_{k+1}}$ representan la posición tridimensional en el instante t_k y en el instante t_{k+1} de un mismo punto en el espacio. En la figura 3.8, se muestran todos los subconjuntos que se definieron anteriormente y las respectivas correspondencias temporales entre estos.

Para encontrar los subconjuntos formados por los puntos \mathbf{q}_{l_s, t_k} y $\mathbf{q}_{l_s, t_{k+1}}$, que forman las correspondencias temporales 2D dadas por $(\mathbf{q}_{l_s, t_k}, \mathbf{q}_{l_s, t_{k+1}})$, como se observa en la figura 3.8, se emplea una adaptación del algoritmo de correlación pseudo-normalizada que fue propuesto e implementado por Moravec (1980) y que posteriormente fue utilizado por Matthies (1989) para encontrar la mejor coincidencia de un punto característico de una imagen, en otra imagen diferente, pero similar.

Primero, se elige de forma totalmente aleatoria y con una probabilidad constante, un punto característico \mathbf{h}'_{li, t_k} en la imagen izquierda antes del movimiento, lo que equivale a elegir de forma aleatoria un número dentro del intervalo de la variable i , dado por la expresión 3.1. Si dicho número aleato-

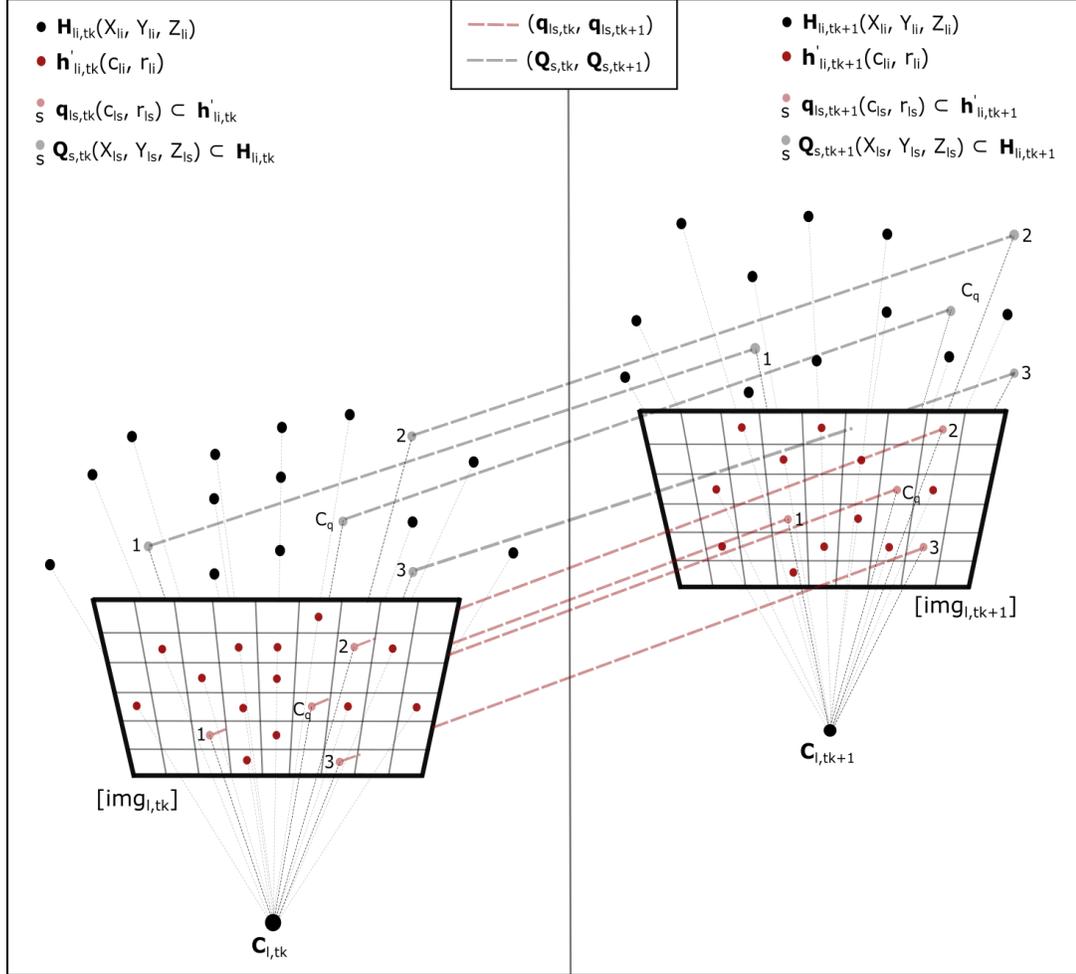


Figura 3.8: Subconjuntos de los puntos característicos y las nubes de puntos antes y después del movimiento, así como las correspondencias temporales entre estos.

rio se denota como a_q , el punto característico aleatorio asociado se obtiene al evaluar $h'_{li,tk}$ en $i = a_q$ (ver la figura 3.9):

$$h'_{li,tk}(c_{li}, r_{li})|_{i=a_q} = h'_{la_q,tk}(c_{la_q}, r_{la_q}) \quad (3.10)$$

donde:

$$1 \leq a_q \leq C_{N_{tk}} \quad (3.11)$$

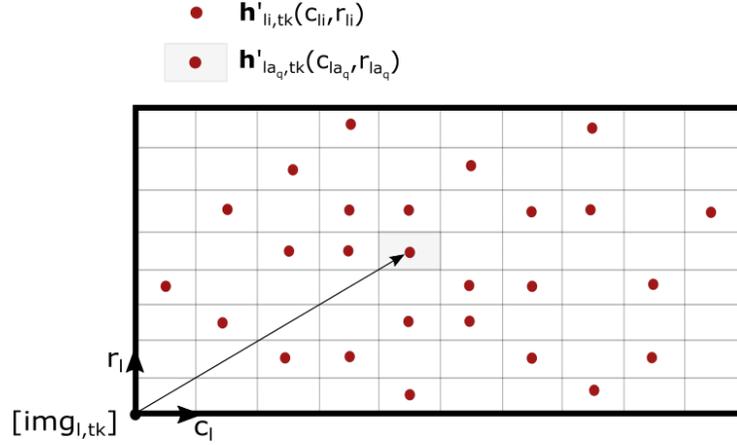


Figura 3.9: Punto característico elegido de forma aleatoria en la imagen izquierda antes del movimiento.

Luego de que se elige un punto característico de forma aleatoria, dado por la expresión 3.10, se establece una pequeña ventana rectangular en la imagen $img_{l, tk}$, con centro en el punto característico $h'_{la_q, tk}$, es decir, con centro en (c_{la_q}, r_{la_q}) . A dicha ventana se le conoce como la ventana de correlación, la cual se denota como V_c y tiene un ancho dado por w_c y una altura dada por h_c . Cabe destacar, que para asegurarse de que el centro geométrico de la ventana de correlación siempre sea el píxel ubicado en (c_{la_q}, r_{la_q}) , en lugar de un subpíxel, el valor de w_c y de h_c tiene que ser un entero impar. Además, dentro de la ventana de correlación hay un total de $w_c \cdot h_c$ píxeles que van desde el píxel dado por:

$$p_{lj, tk}(c_{lj}, r_{lj})|_{j=p_{vc1}} = p_{lp_{vc1}, tk}(c_{lp_{vc1}}, r_{lp_{vc1}}) = (c_{la_q} - \frac{w_c - 1}{2}, r_{la_q} - \frac{h_c - 1}{2}) \quad (3.12)$$

en la esquina inferior izquierda de V_c , hasta el píxel dado por:

$$p_{lj, tk}(c_{lj}, r_{lj})|_{j=p_{vcf}} = p_{lp_{vcf}, tk}(c_{lp_{vcf}}, r_{lp_{vcf}}) = (c_{la_q} + \frac{w_c - 1}{2}, r_{la_q} + \frac{h_c - 1}{2}) \quad (3.13)$$

en la esquina superior derecha de V_c , donde p_{vc1} y p_{vcf} se obtienen al evaluar la expresión 3.3 en las componentes de 3.12 y 3.13, respectivamente:

$$p_{vc1} = c_{lp_{vc1}} + (r_{lp_{vc1}} - 1) \cdot w = (c_{la_q} - \frac{w_c - 1}{2}) + (r_{la_q} - \frac{h_c - 1}{2} - 1) \cdot w \quad (3.14)$$

$$p_{v_{cf}} = c_{lp_{v_{cf}}} + (r_{lp_{v_{cf}}} - 1) \cdot w = \left(c_{la_q} + \frac{w_c - 1}{2}\right) + \left(r_{la_q} + \frac{h_c - 1}{2} - 1\right) \cdot w \quad (3.15)$$

Cada uno de los píxeles que se encuentran dentro de la ventana V_c tienen un valor de intensidad que va desde $I_{lj=p_{v_{c1}},t_k}$ hasta $I_{lj=p_{v_{cf}},t_k}$. En la figura 3.10 se muestra de forma detallada la ventana de correlación V_c .

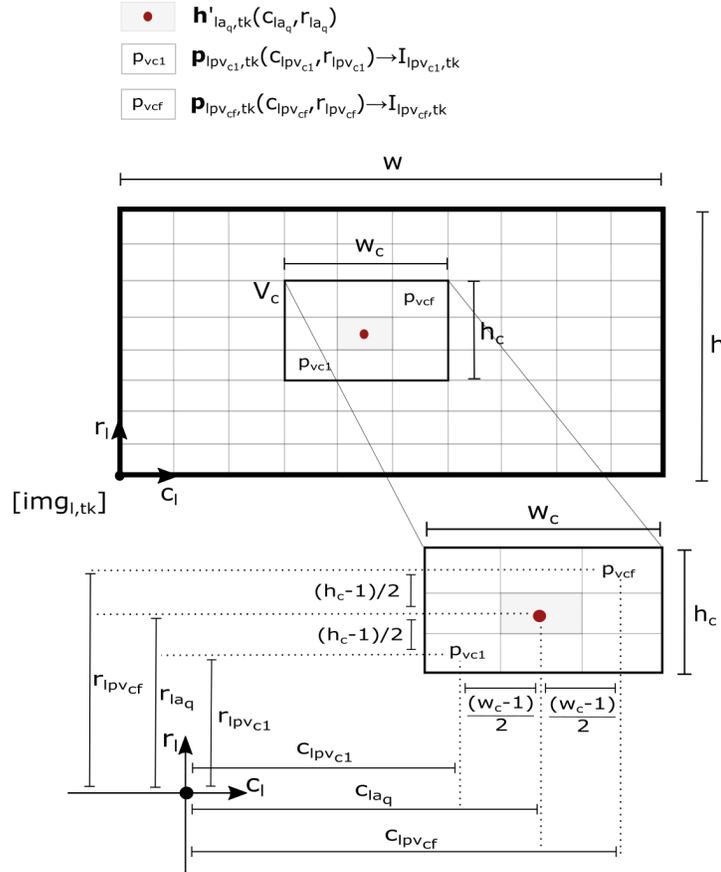


Figura 3.10: Ventana de correlación en la imagen izquierda antes del movimiento, con centro en el punto característico elegido aleatoriamente.

Luego de que se establece la ventana de correlación V_c con centro en el punto característico $\mathbf{h}'_{la_q, tk}$ en la imagen izquierda img_l, tk antes del movimiento, se procede a calcular la media y la varianza del conjunto de valores de intensidad de los píxeles dentro de dicha ventana, las cuales se necesitan más adelante para establecer la correspondencia. La media de la ventana V_c se denota como μ_{v_c} y la varianza como $\sigma_{v_c}^2$, las cuales se calculan de la siguiente forma:

$$\mu_{v_c} = \frac{1}{w_c \cdot h_c} \sum_{r_l=r_{lp_{vc1}}}^{r_{lp_{vcf}}} \sum_{c_l=c_{lp_{vc1}}}^{c_{lp_{vcf}}} I_{l_j,t_k} |_{j=[c_l+(r_l-1) \cdot w]} \quad (3.16)$$

$$\sigma_{v_c}^2 = \frac{1}{w_c \cdot h_c} \left(\sum_{r_l=r_{lp_{vc1}}}^{r_{lp_{vcf}}} \sum_{c_l=c_{lp_{vc1}}}^{c_{lp_{vcf}}} I_{l_j,t_k}^2 |_{j=[c_l+(r_l-1) \cdot w]} \right) - \mu_{v_c}^2 \quad (3.17)$$

Una vez que se calcula la media y la varianza de la ventana V_c con centro en el punto característico \mathbf{h}'_{la_q,t_k} , se procede a establecer la correspondencia temporal 2D entre dicho punto y un punto característico $\mathbf{h}'_{lm,t_{k+1}}$ elegido cuidadosamente. Un punto característicos $\mathbf{h}'_{lm,t_{k+1}}$ en la imagen $img_{l,t_{k+1}}$, responderá al punto \mathbf{h}'_{la_q,t_k} en la imagen img_{l,t_k} , si la ventana que se establece con centro en el punto $\mathbf{h}'_{lm,t_{k+1}}$ es la que más se asemeja a la ventana de correlación V_c , de entre todas la ventanas que se pueden establecer en la imagen $img_{l,t_{k+1}}$. Entonces, lo que en realidad se hace es rastrear la ventana de correlación V_c en la imagen $img_{l,t_{k+1}}$ después del movimiento, es decir, se encuentra la ventana en la imagen $img_{l,t_{k+1}}$ que más se asemeja a la ventana de correlación V_c , de tal forma que se establece una correspondencia temporal 2D entre los centros de las ventanas. La razón principal por la cual se rastrea la ventana V_c , en vez de rastrear solamente el píxel \mathbf{h}'_{la_q,t_k} , es para evitar correspondencias erróneas, debido a que la única característica de un píxel \mathbf{h}'_{la_q,t_k} es su valor de intensidad y existen muchos píxeles que tienen valores de intensidad iguales y similares. En cambio, una ventana es un pedazo de la imagen formado por un conjunto de píxeles alrededor \mathbf{h}'_{la_q,t_k} , con valores de intensidad distintos, que la diferencian de otras ventanas con el mismo ancho y largo.

El rastreo de la ventana V_c no se lleva a cabo a lo largo de toda la imagen $img_{l,t_{k+1}}$, debido a que eso sería un proceso muy largo e innecesario, que consumiría mucho tiempo de procesamiento. Es por ello que se establece una ventana de búsqueda en la imagen $img_{l,t_{k+1}}$, con centro en (c_{la_q}, r_{la_q}) , que es un píxel ubicado en la misma posición que el punto característico \mathbf{h}'_{la_q,t_k} , pero en la imagen después del movimiento. Dicha ventana se denota como V_b y tiene un ancho dado por w_b y una altura dada por h_b , donde:

$$w_b \gg w_c \quad (3.18)$$

$$h_b \gg h_c \quad (3.19)$$

Al igual que con la ventana de correlación V_c , el valor de w_b y h_b tiene que ser un entero impar. Además, de forma análoga a V_c , la ventana de búsqueda V_b tiene un total de $w_b \cdot h_b$ píxeles que van desde el píxel dado por:

$$\mathbf{p}_{ln,t_{k+1}}(c_{ln}, r_{ln})|_{n=p_{vb1}} = \mathbf{p}_{lp_{vb1},tk}(c_{lp_{vb1}}, r_{lp_{vb1}}) = (c_{la_q} - \frac{w_b - 1}{2}, r_{la_q} - \frac{h_b - 1}{2}) \quad (3.20)$$

en la esquina inferior izquierda de V_b , hasta el píxel dado por:

$$\mathbf{p}_{ln,t_{k+1}}(c_{ln}, r_{ln})|_{n=p_{vbf}} = \mathbf{p}_{lp_{vbf},tk}(c_{lp_{vbf}}, r_{lp_{vbf}}) = (c_{la_q} + \frac{w_b - 1}{2}, r_{la_q} + \frac{h_b - 1}{2}) \quad (3.21)$$

en la esquina superior derecha de V_b , donde:

$$p_{vb1} = c_{lp_{vb1}} + (r_{lp_{vb1}} - 1) \cdot w = (c_{la_q} - \frac{w_b - 1}{2}) + (r_{la_q} - \frac{h_b - 1}{2} - 1) \cdot w \quad (3.22)$$

$$p_{vbf} = c_{lp_{vbf}} + (r_{lp_{vbf}} - 1) \cdot w = (c_{la_q} + \frac{w_b - 1}{2}) + (r_{la_q} + \frac{h_b - 1}{2} - 1) \cdot w \quad (3.23)$$

En la figura 3.11 se muestra la ventana de búsqueda V_b junto con todos sus detalles particulares.

Una vez que se establece la ventana de búsqueda V_b en la imagen $img_{l,t_{k+1}}$, con centro en el píxel ubicado en (c_{la_q}, r_{la_q}) , se puede llevar a cabo el rastreo de la ventana de correlación V_c dentro de dicha ventana de búsqueda. Para ello, se recorre cada uno de los píxeles $\mathbf{p}_{ln,t_{k+1}}(c_{ln}, r_{ln})$ que se encuentran dentro la ventana de búsqueda V_b , desde el píxel $\mathbf{p}_{lp_{vb1},t_{k+1}}$ hasta el píxel $\mathbf{p}_{lp_{vbf},t_{k+1}}$, de la siguiente forma:

$$\begin{aligned} & \text{para } r_l = r_{lp_{vb1}}, \text{ hasta } r_{lp_{vbf}} : \\ & \text{para } c_l = c_{lp_{vb1}}, \text{ hasta } c_{lp_{vbf}} : \end{aligned} \quad (3.24)$$

$$\mathbf{p}_{ln,t_{k+1}}(c_{ln}, r_{ln})|_{n=[c_l+(r_l-1) \cdot w]}$$

Como se mencionó al inicio de esta sección, las correspondencias temporales 2D se deben de establecer entre los puntos característicos \mathbf{h}'_{li,t_k} en la imagen izquierda antes del movimiento y los puntos característicos $\mathbf{h}'_{lm,t_{k+1}}$ en la imagen izquierda después del movimiento. Es por ello, que por cada uno de los píxeles dentro de la ventana V_b , se verifica si cumple con la condición dada por la expresión 3.25, la cual sirve para determinar si el píxel es un punto característico o no, a partir del valor de pc_n asociado, tal y como se ejemplifica en la figura 3.12.

$$\begin{aligned} & \text{si } pc_n|_{n=[c_l+(r_l-1) \cdot w]} > 0 \longrightarrow \\ & \mathbf{p}_{ln,t_{k+1}}(c_{ln}, r_{ln})|_{n=[c_l+(r_l-1) \cdot w]} = \mathbf{h}'_{lm,t_{k+1}}(c_{lm}, r_{lm})|_{m=pc_n} \end{aligned} \quad (3.25)$$

$$n_{pc} = n_{pc} + 1$$

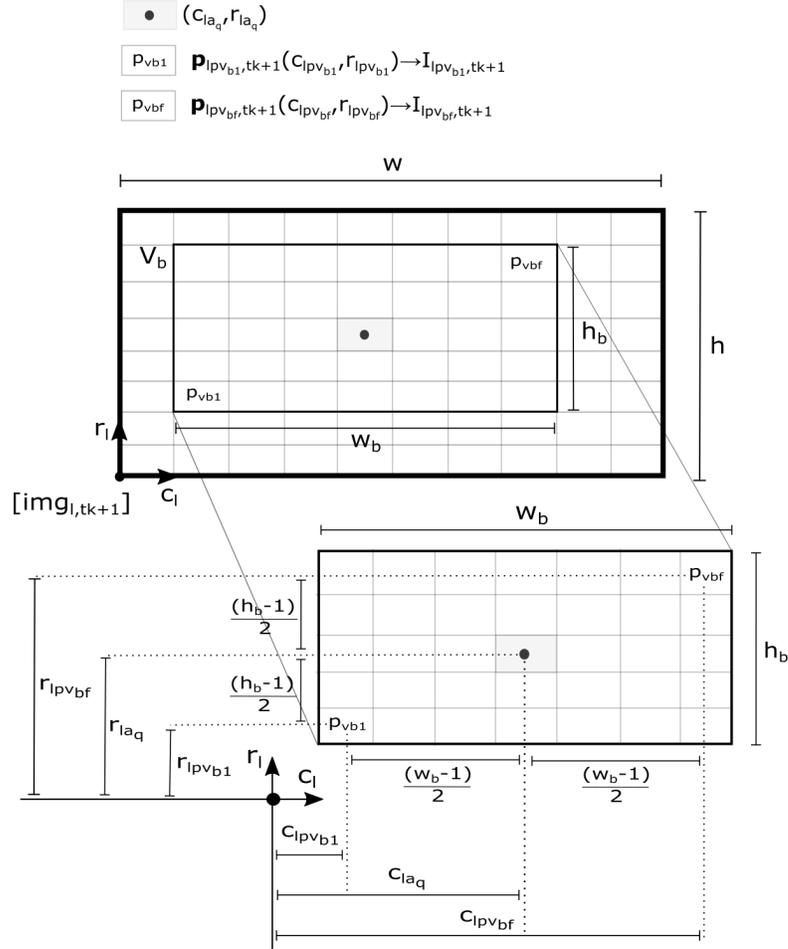


Figura 3.11: Ventana de búsqueda en la imagen izquierda después del movimiento, con centro en las mismas coordenadas del punto característico elegido aleatoriamente en la imagen izquierda antes del movimiento.

Además, se define una variable dada por n_{pc} que se utilizara más adelante y que sirve para determinar el número de aparición de cada punto característico dentro de la ventana de búsqueda, es decir, cuando se encuentra el primer punto característico entonces $n_{pc} = 1$, cuando se encuentre el segundo $n_{pc} = 2$ y así sucesivamente. El valor de n_{pc} empieza en 0 y se aumenta en 1 cada vez que se encuentra un punto característico dentro de V_b . En la figura 3.12 también se ejemplifica esto.

Entonces, se va recorriendo cada píxel dentro de la ventana de búsqueda V_b , mediante la expresión 3.24 y cada vez que se encuentra un píxel que cumple con la condición 3.25, que es un punto característico, se establece una ventana

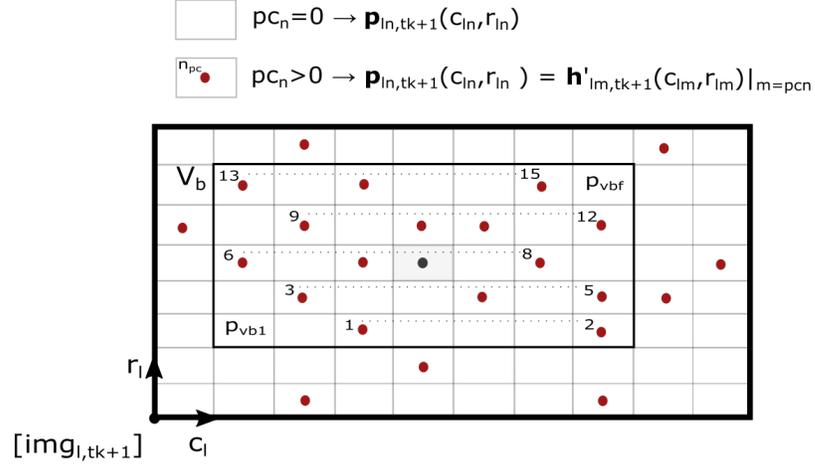


Figura 3.12: Puntos característicos dentro de la ventana de búsqueda y su número de aparición.

de correlación en la imagen $img_{l,tk+1}$, con centro en (c_{lpc_n}, r_{lpc_n}) , tal y como se muestra en la figura 3.13. Dicha ventana se denota como V_{c,pc_n} y tiene el mismo ancho y largo que V_c , dados por w_c y h_c , respectivamente. Los píxeles dentro de la ventana V_{c,pc_n} van desde el píxel dado por la expresión 3.26, hasta el píxel dado por la expresión 3.27.

$$\begin{aligned}
 \mathbf{p}_{ln,tk+1}(c_{ln}, r_{ln})|_{n=p_{vc1,pcn}} &= \mathbf{p}_{lp_{vc1,pcn},tk}(c_{lp_{vc1,pcn}}, r_{lp_{vc1,pcn}}) \\
 &= (c_{lpc_n} - \frac{w_b - 1}{2}, r_{lpc_n} - \frac{h_b - 1}{2})
 \end{aligned} \tag{3.26}$$

$$\begin{aligned}
 \mathbf{p}_{ln,tk+1}(c_{ln}, r_{ln})|_{n=p_{vcf,pcn}} &= \mathbf{p}_{lp_{vcf,pcn},tk}(c_{lp_{vcf,pcn}}, r_{lp_{vcf,pcn}}) \\
 &= (c_{lpc_n} + \frac{w_b - 1}{2}, r_{lpc_n} + \frac{h_b - 1}{2})
 \end{aligned} \tag{3.27}$$

donde:

$$\begin{aligned}
 p_{vc1,pcn} &= c_{lp_{vc1,pcn}} + (r_{lp_{vc1,pcn}} - 1) \cdot w \\
 &= (c_{lpc_n} - \frac{w_b - 1}{2}) + (r_{lpc_n} - \frac{h_b - 1}{2} - 1) \cdot w
 \end{aligned} \tag{3.28}$$

$$\begin{aligned}
 p_{vcf,pcn} &= c_{lp_{vcf,pcn}} + (r_{lp_{vcf,pcn}} - 1) \cdot w \\
 &= (c_{lpcn} + \frac{w_b - 1}{2}) + (r_{lpcn} + \frac{h_b - 1}{2} - 1) \cdot w
 \end{aligned}
 \tag{3.29}$$

Cada píxel dentro de $V_{c,pcn}$ tiene un valor de intensidad que va desde $I_{ln=p_{vc1,pcn},tk+1}$ hasta $I_{ln=p_{vc1,pcn},tk+1}$.

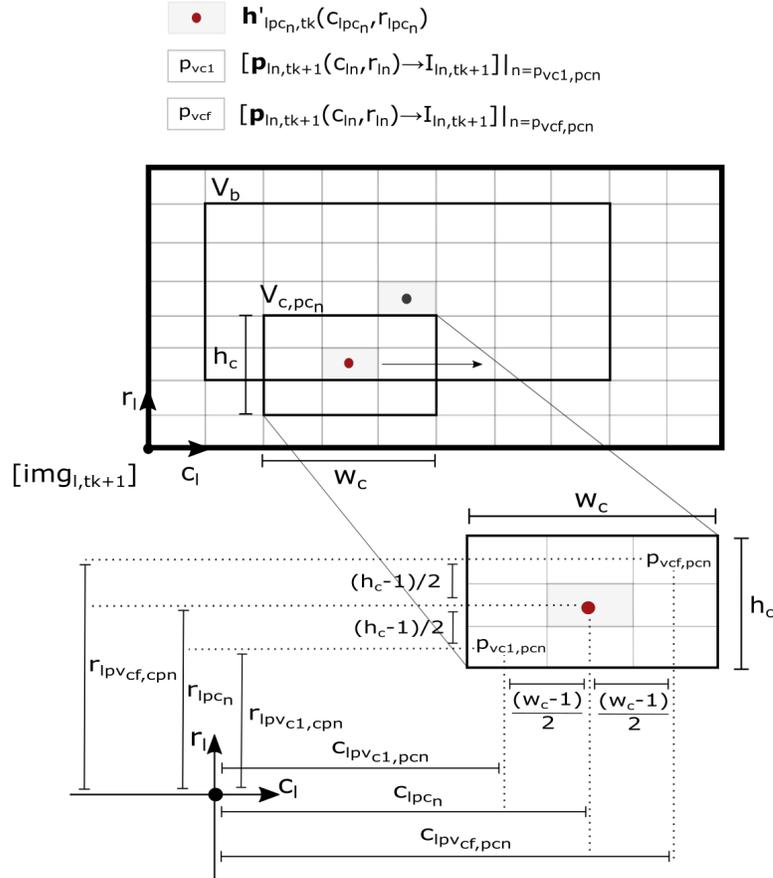


Figura 3.13: Ventana de correlación en la imagen izquierda después del movimiento, con centro en un punto característico que se encuentra dentro de la ventana de búsqueda.

Como se observa en la figura 3.14, cada vez que se establece una ventana de correlación $V_{c,pcn}$, se compara con la ventana V_c mediante un coeficiente de similitud definido por Moravec (1980), que sirve para determinar qué tan similar es la ventana $V_{c,pcn}$ con respecto a la ventana V_c , partir de los valores de intensidad de cada uno de los píxeles que las conforman.

Además, como se puede ver en la figura 3.14, a cada ventana V_{c,pc_n} se le asocia un coeficiente de similitud, que se denota como cpn_{pc_n} y que puede variar entre 0 y 1:

$$0 \leq cpn_{pc_n} \leq 1 \quad (3.30)$$

Un valor de 1 quiere decir que los valores de intensidad de los píxeles dentro de la ventana V_{c,pc_n} son totalmente idénticos a los valores de intensidad de los respectivos píxeles dentro de la ventana V_c y un valor de 0 que no se parecen absolutamente en nada. Dicho coeficiente se calcula a partir de la correlación pseudo-normalizada propuesta por Moravec, entre la ventana V_{c,pc_n} y la ventana V_c , el cual depende sus medias y las varianzas:

$$cpn_{pc_n} = \frac{2(S_I - \mu_{v_c} \cdot \mu_{v_c,pc_n})}{w_c \cdot h_c \cdot (\sigma_{v_c}^2 + \sigma_{v_c,pc_n}^2)} \quad (3.31)$$

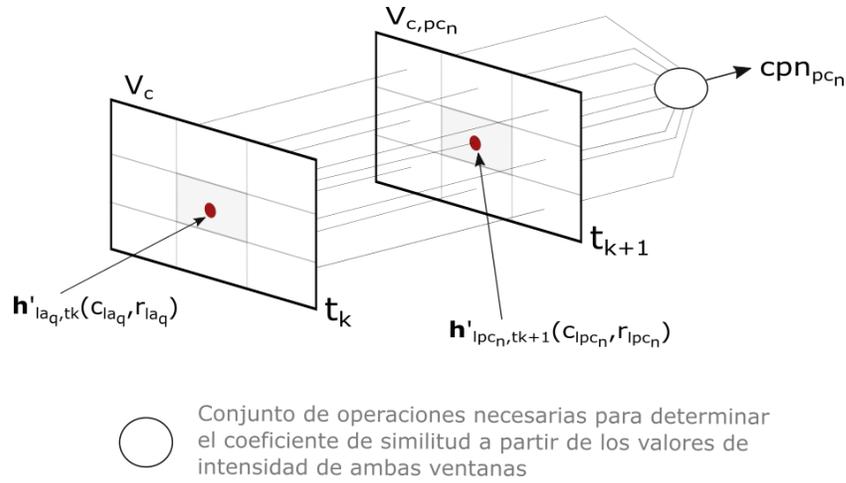


Figura 3.14: Comparación entre la ventana de correlación en la imagen izquierda antes del movimiento y las ventanas de correlación en la imagen izquierda después del movimiento.

donde S_I es una sumatoria dada por la expresión 3.32, que consiste en la suma de la multiplicación de cada uno de los valores de intensidad de V_c con los respectivos valores de intensidad de V_{c,pc_n} , es decir, se multiplica el valor de intensidad del primer píxel de V_c con el valor de intensidad del primer píxel de V_{c,pc_n} , luego se suma a la multiplicación de los valores de intensidad del segundo píxel de cada ventana y así sucesivamente hasta el último píxel. Las variables μ_{v_c} y $\sigma_{v_c}^2$ son la media y la varianza de los valores de intensidad de la ventana V_c , respectivamente y están dados por las expresiones 3.16 y 3.17.

Análogamente, las variables μ_{v_c,pc_n} y σ_{v_c,pc_n}^2 son la media y la varianza de los valores de intensidad de la ventana V_{c,pc_n} , respectivamente y se calculan con las expresiones 3.33 y 3.34.

$$S_I = \sum_{r_l=r_{lp_{vc1}}}^{r_{lp_{vcf}}} \sum_{c_l=c_{lp_{vc1}}}^{c_{lp_{vcf}}} (I_{lj,t_k} | j=[c_l+(r_l-1) \cdot w]) \cdot (I_{ln,t_{k+1}} | n=[c_l-c'_l+(r_l-r'_l-1) \cdot w]) \quad (3.32)$$

$$\mu_{v_c,pc_n} = \frac{1}{w_c \cdot h_c} \sum_{r_l=r_{lp_{vc1,pc_n}}}^{r_{lp_{vcf,pc_n}}} \sum_{c_l=c_{lp_{vc1,pc_n}}}^{c_{lp_{vcf,pc_n}}} I_{ln,t_{k+1}} | n=[c_l+(r_l-1) \cdot w] \quad (3.33)$$

$$\sigma_{v_c,pc_n}^2 = \frac{1}{w_c \cdot h_c} \left(\sum_{r_l=r_{lp_{vc1,pc_n}}}^{r_{lp_{vcf,pc_n}}} \sum_{c_l=c_{lp_{vc1,pc_n}}}^{c_{lp_{vcf,pc_n}}} I_{ln,t_{k+1}}^2 | n=[c_l+(r_l-1) \cdot w] \right) - \mu_{v_c,pc_n}^2 \quad (3.34)$$

Además, en la expresión 3.32, las variables c'_l y r'_l se calculan con las ecuaciones 3.35 y 3.36, que se obtienen a partir del vector que se forma entre el centro de la ventana V_c , dado por (c_{la_q}, r_{la_q}) y el centro de la ventana V_{c,pc_n} , dado por (c_{lpc_n}, r_{lpc_n}) , suponiendo que ambas ventanas están en una misma imagen ficticia, tal y como se observa en la figura 3.15. Dichas variables se utilizan para que dado un píxel en alguna posición (cl, rl) dentro de la ventana V_c en el instante t_k , se pueda obtener el píxel equivalente, pero en la ventana V_{c,pc_n} en el instante t_{k+1} . Por ejemplo, si en (cl, rl) se tiene el primer píxel de la ventana V_c , se puede obtener el primer píxel de la ventana V_{c,pc_n} a partir de c'_l y r'_l , como se aprecia en la figura 3.16.

$$c'_l = c_{la_q} - c_{lpc_n} \quad (3.35)$$

$$r'_l = r_{la_q} - r_{lpc_n} \quad (3.36)$$

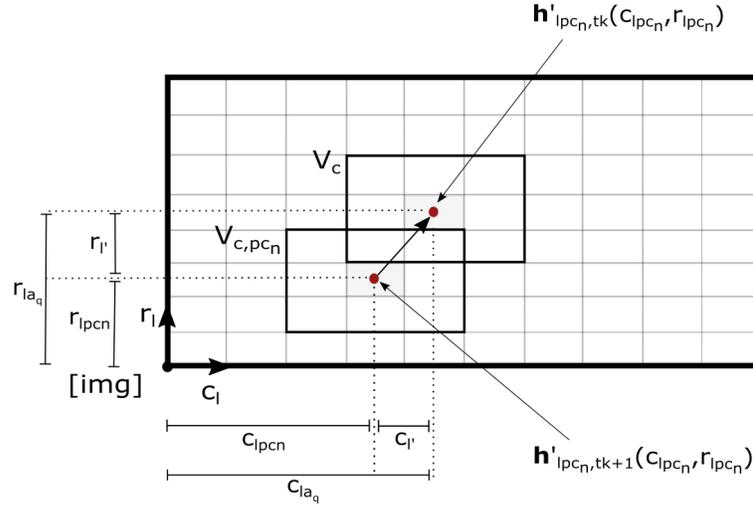


Figura 3.15: Vector que se forma entre el centro de las ventanas de correlación, si estas se encuentran en una misma imagen ficticia.

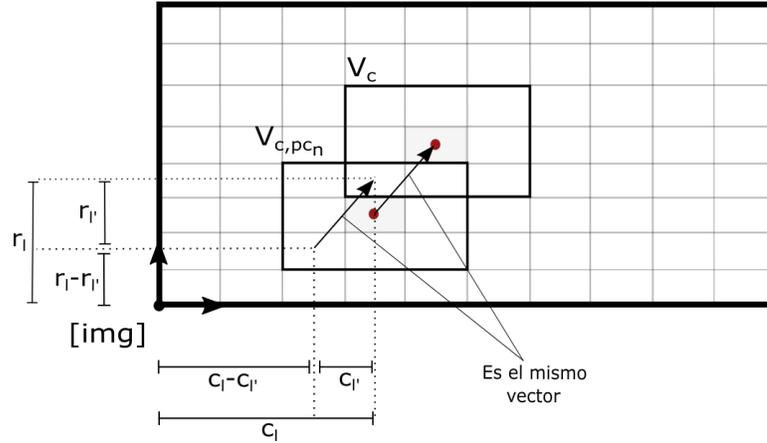


Figura 3.16: Relación que existe entre un píxel de la ventana V_c y el píxel equivalente en la ventana $V_{c,pcn}$.

Luego de que se calcula el coeficiente de similitud, a partir de la expresión 3.31, que permite determinar que tanto se asemeja la ventana $V_{c,pcn}$ a la ventana V_c , se guarda su valor en la posición n_{pc} de un vector dado por \mathbf{V}_{cpn} :

$$\mathbf{V}_{cpn}[n_{pc}] = cpn_{pcn} \quad (3.37)$$

Una vez que se ha recorrido toda la ventana de búsqueda V_b y se ha calculado y guardado el coeficiente de similitud cpn_{pcn} para cada una de las

ventanas V_{c,pc_n} , que se establecen para cada uno de los puntos característicos $\mathbf{h}'_{lm,t_{k+1}}(c_{lm}, r_{lm})|_{m=pc_n}$ que se encontraron dentro de la ventana V_b , se procede a encontrar la ventana V_{c,pc_n} con el coeficiente de similitud con el valor más cercano a 1, que se denota como $cpn_{pc_{n_{max}}}$ y que equivale a encontrar el máximo del vector \mathbf{V}_{pnc} :

$$cpn_{pc_{n_{max}}} = \max(\mathbf{V}_{cpn}) \quad (3.38)$$

Como se observa en la figura 3.17, dicho coeficiente pertenece a la ventana de correlación dada por $V_{c,pc_{n_{max}}}$, cuyo centro es el píxel dado por:

$$\mathbf{p}_{ln,t_{k+1}}(c_{ln}, r_{ln})|_{n=n_{max}} = \mathbf{p}_{ln_{max},t_{k+1}}(c_{ln_{max}}, r_{ln_{max}}) \quad (3.39)$$

el cual es un punto característico que se encuentra dentro del subconjunto formado por los puntos $\mathbf{h}'_{lm,t_{k+1}}$ y que se puede obtener a partir del valor de $pc_{n_{max}}$ asociado a él:

$$\mathbf{h}_{lm,t_{k+1}}(c_{lm}, r_{lm})|_{m=pc_{n_{max}}} = \mathbf{h}_{lpc_{n_{max}},t_{k+1}}(c_{lpc_{n_{max}}}, r_{lpc_{n_{max}}}) \quad (3.40)$$

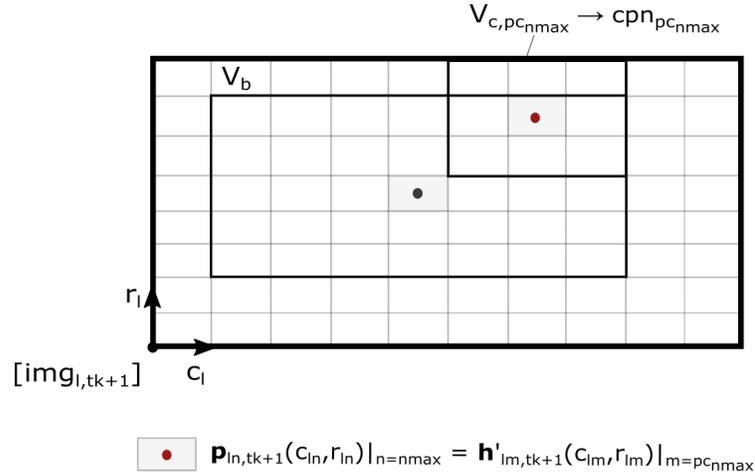


Figura 3.17: Ventana de correlación en la imagen después del movimiento con el coeficiente de similitud máximo.

El hecho de que la ventana de correlación $V_{c,pc_{n_{max}}}$ tenga el coeficiente de similitud máximo, quiere decir que de entre todas las ventanas \mathbf{V}_{cpn} que se establecieron, la ventana $V_{c,pc_{n_{max}}}$ es la más similar a la ventana de correlación V_c . Esto a su vez implica que existen altas posibilidades de que se establezca una correspondencia temporal 2D entre el punto característico \mathbf{h}_{la_q,t_k} que es el centro de la ventana V_c en la imagen izquierda antes del movimiento y el

punto característicos $\mathbf{h}_{lpc_{n_{max}},t_{k+1}}$ que es el centro de la ventana $V_{c,pc_{n_{max}}}$ en la imagen izquierda después del movimiento.

Entonces, luego de que se calcula el coeficiente de similitud máximo, mediante la expresión 3.38, se determina si se puede establecer una correspondencia temporal 2D entre \mathbf{h}_{la_q,t_k} y $\mathbf{h}_{lpc_{n_{max}},t_{k+1}}$ o no. Como ya se ha mencionado anteriormente, la correspondencia temporal 2D dada por $(\mathbf{h}_{la_q,t_k}, \mathbf{h}_{lpc_{n_{max}},t_{k+1}})$ quiere decir que dichos puntos característicos representan la proyección en la imagen izquierda antes y después del movimiento, respectivamente, de un mismo punto en el espacio. Resulta que existe la posibilidad de que la proyección de dicho punto en la imagen izquierda después del movimiento, haya quedado fuera del campo visión de la cámara izquierda después del movimiento y por lo tanto, el punto característico $\mathbf{h}_{lpc_{n_{max}},t_{k+1}}$ con el coeficiente de similitud más alto, es la proyección de otro punto en el espacio y no forma una correspondencia con \mathbf{h}_{la_q,t_k} . Algunos de esos casos se pueden identificar cuando el valor de $cpn_{pc_{n_{max}}}$ es igual o menor 0,7. Es por ello, que se verifica que el valor del coeficiente de similitud máximo $cpn_{pc_{n_{max}}}$ sea mayor a un cpn_{min} , como un primer paso para evitar correspondencias erróneas:

$$\begin{aligned} & \text{si } cpn_{pc_{n_{max}}} > cpn_{min} \longrightarrow \\ \mathbf{h}_{la_q,t_k}(cl_{a_q}, r_{la_q}) & \iff \mathbf{h}_{lpc_{n_{max}},t_{k+1}}(cl_{pc_{n_{max}}}, r_{lpc_{n_{max}}}) \\ & n_c = n_c + 1 \end{aligned} \quad (3.41)$$

donde cpn_{min} es una constante cuyo valor se puede elegir entre 0.8 y 0.95, dependiendo de qué tan alta se quiere que sea la probabilidad de que el punto $\mathbf{h}_{lpc_{n_{max}},t_{k+1}}$ forme una correspondencia temporal correcta con \mathbf{h}_{la_q,t_k} . También se define una variable dada por n_c que sirve para determinar el número de la correspondencia, es decir, cuando se encuentra la primer correspondencia valida entonces $n_c = 1$, cuando se encuentre la segunda $n_c = 2$ y así sucesivamente. El valor de n_c empieza en 0 y se aumenta en 1 cada vez que se encuentra una correspondencia valida.

Entonces, si la condición dada por la expresión 3.41 se cumple, quiere decir que se puede establecer una correspondencia temporal 2D entre \mathbf{h}_{la_q,t_k} y $\mathbf{h}_{lpc_{n_{max}},t_{k+1}}$, tal y como se observa en la figura 3.18 y que se representa como $(\mathbf{h}_{la_q,t_k}, \mathbf{h}_{lpc_{n_{max}},t_{k+1}})$. Pero, si la condición anterior no se cumple entonces no se pudo establecer una correspondencia temporal valida y, por lo tanto, se debe de seleccionar otro punto característico en la imagen img_{l,t_k} de forma aleatoria, es decir, un nuevo valor para a_q y repetir el proceso hasta que la condición se cumpla, hasta que se pueda establecer una correspondencia temporal.

Además, las posiciones tridimensionales de la correspondencia temporal 2D dada por $(\mathbf{h}_{la_q,t_k}, \mathbf{h}_{lpc_{n_{max}},t_{k+1}})$, forman una correspondencia temporal 3D dada por $(\mathbf{H}_{a_q,t_k}, \mathbf{H}_{pc_{n_{max}},t_{k+1}})$, que también se pueden representar de la siguiente forma:

$$\begin{aligned} \mathbf{h}_{la_q, t_k}(c_{la_q}, r_{la_q}) &\iff \mathbf{h}_{lpc_{n_{max}}, t_{k+1}}(c_{lpc_{n_{max}}}, r_{lpc_{n_{max}}}) \longrightarrow \\ \mathbf{H}_{a_q, t_k}(X_{la_q}, Y_{la_q}, Z_{la_q}) &\iff \mathbf{H}_{pc_{n_{max}}, t_{k+1}}(X_{lpc_{n_{max}}}, Y_{lpc_{n_{max}}}, Z_{lpc_{n_{max}}}) \end{aligned} \quad (3.42)$$

Por último, se guardan las correspondencias temporales 2D y 3D, dadas por la expresión anterior, en los respectivos subconjuntos que se definieron al principio y de acuerdo al valor de n_c :

$$\mathbf{q}_{ls, t_k}(c_{ls}, r_{ls})|_{s=n_c} = \mathbf{h}_{la_q, t_k}(c_{la_q}, r_{la_q}) \quad (3.43)$$

$$\mathbf{q}_{ls, t_{k+1}}(c_{ls}, r_{ls})|_{s=n_c} = \mathbf{h}_{lpc_{n_{max}}, t_{k+1}}(c_{lpc_{n_{max}}}, r_{lpc_{n_{max}}}) \quad (3.44)$$

$$\mathbf{Q}_{s, t_k}(X_{ls}, Y_{ls}, Z_{ls})|_{s=n_c} = \mathbf{H}_{a_q, t_k}(X_{la_q}, Y_{la_q}, Z_{la_q}) \quad (3.45)$$

$$\mathbf{Q}_{s, t_{k+1}}(X_{ls}, Y_{ls}, Z_{ls})|_{s=n_c} = \mathbf{H}_{pc_{n_{max}}, t_{k+1}}(X_{lpc_{n_{max}}}, Y_{lpc_{n_{max}}}, Z_{lpc_{n_{max}}}) \quad (3.46)$$

En la figura 3.18, se muestra de forma resumida todo el proceso que se debe de seguir para establecer una correspondencia temporal entre el punto característico $\mathbf{h}_{la_q, t_k}(c_{la_q}, r_{la_q})$ ubicado en la imagen img_{l, t_k} izquierda antes del movimiento y un punto característico $\mathbf{h}'_{lm, t_{k+1}}$ ubicado en la imagen izquierda $img_{l, t_{k+1}}$ después del movimiento

Todo el proceso descrito anteriormente se debe de repetir una y otra vez hasta que se $n_c = C_q$, es decir, hasta que se hayan establecido un total de C_q correspondencias temporales 2D y, por lo tanto, un total de C_q correspondencias temporales 3D. De tal forma que al final, los subconjuntos de tamaño C_q formados por los puntos característicos \mathbf{q}_{ls, t_k} y $\mathbf{q}_{ls, t_{k+1}}$ que forman las correspondencias temporales 2D dadas por $(\mathbf{q}_{ls, t_k}, \mathbf{q}_{ls, t_{k+1}})$, sean conocidos, así como los respectivos subconjuntos de tamaño C_q formados por las posiciones tridimensionales \mathbf{Q}_{s, t_k} y $\mathbf{Q}_{s, t_{k+1}}$, que forman las correspondencias temporales 3D dadas por $(\mathbf{Q}_{s, t_k}, \mathbf{Q}_{s, t_{k+1}})$, tal y como se muestra en la figura 3.8 al principio de esta sección.

3.4. Estimación inicial de movimiento

Es el tercer y último paso del algoritmo de odometría visual estereoscópico implementado, que consiste en estimar el movimiento del astromóvil entre en el instante de tiempo t_k y el instante de tiempo t_{k+1} , a partir de las correspondencias temporales 3D $(\mathbf{Q}_{s, t_k}, \mathbf{Q}_{s, t_{k+1}})$, lo que equivale a estimar el movimiento

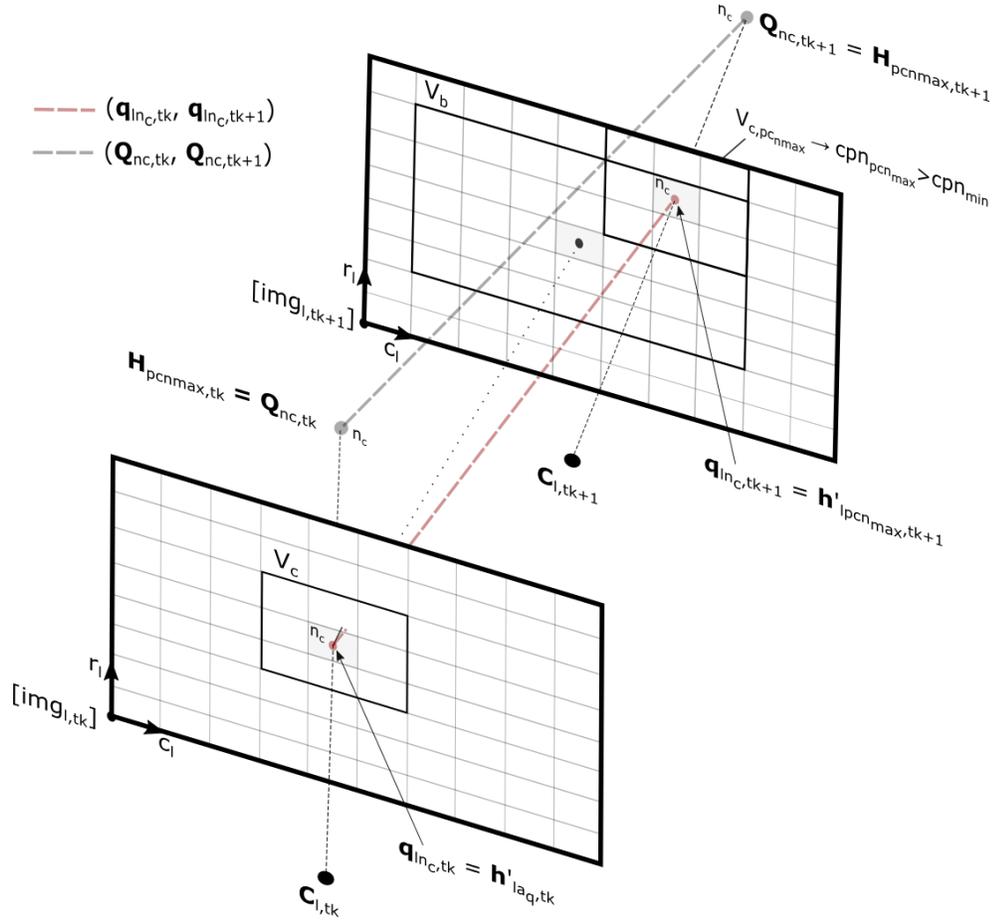


Figura 3.18: Establecimiento de una correspondencia temporal entre un punto característico elegido aleatoriamente en la imagen izquierda antes del movimiento y un punto característico elegido cuidadosamente en la imagen izquierda después del movimiento.

que experimenta la cámara izquierda del arreglo estereoscópico entre esos instantes, ya que el arreglo está atado rígidamente a un costado del rover. Para ello, se debe de encontrar la traslación $\Delta \mathbf{T}$ y rotación $\Delta \mathbf{R}$, con la que se debe de mover cada uno de los puntos tridimensionales \mathbf{Q}_{s,t_k} , que conforman un subconjunto de la nube de puntos \mathbf{N}_{t_k} antes del movimiento, para que estos coincidan con los respectivos puntos tridimensionales $\mathbf{Q}_{s,t_{k+1}}$, que conforman un subconjunto de la nube de puntos $\mathbf{N}_{t_{k+1}}$ después del movimiento.

La coincidencia entre los puntos tridimensionales $\mathbf{Q}_{s,t_{k+1}}$ y los puntos tridimensionales \mathbf{Q}_{s,t_k} transformados, que se obtienen al rotar y trasladar los puntos \mathbf{Q}_{s,t_k} con la matriz $\Delta \mathbf{R}$ y el vector $\Delta \mathbf{T}$, se puede representar a partir

de la ecuación de pose, dada por la expresión 2.22 de la sección 2.1.3 del capítulo 2. Simplemente se sustituye \mathbf{H} , \mathbf{H}^* , \mathbf{R} y \mathbf{T} , por $\mathbf{Q}_{s,t_{k+1}}$, \mathbf{Q}_{s,t_k} , $\Delta\mathbf{R}$ y $\Delta\mathbf{T}$, respectivamente, para así obtener la siguiente expresión:

$$\mathbf{Q}_{s,t_{k+1}} = \Delta\mathbf{R} \cdot \mathbf{Q}_{s,t_k} + \Delta\mathbf{T} \quad \forall s \quad (3.47)$$

Si el punto tridimensional \mathbf{Q}_{s,t_k} transformado, se pasa a restar al lado derecho de la ecuación anterior, se obtiene una expresión equivalente pero que tiene mucho más sentido:

$$\mathbf{Q}_{s,t_{k+1}} - \Delta\mathbf{R} \cdot \mathbf{Q}_{s,t_k} - \Delta\mathbf{T} = \mathbf{0} \quad \forall s \quad (3.48)$$

donde $\mathbf{0}$ es el vector tridimensional nulo cuyas componentes son $(0, 0, 0)^T$.

Además, de ahora en adelante los puntos tridimensionales \mathbf{Q}_{s,t_k} transformados, se van a representar como $\mathbf{Q}'_{s,t_{k+1}}$ y van a estar dados por la siguiente expresión:

$$\mathbf{Q}'_{s,t_{k+1}} = \Delta\mathbf{R} \cdot \mathbf{Q}_{s,t_k} + \Delta\mathbf{T} \quad \forall s \quad (3.49)$$

Cabe destacar, que como los puntos tridimensionales $\mathbf{Q}'_{s,t_{k+1}}$ se obtienen a partir de la rotación y traslación de \mathbf{Q}_{s,t_k} , se pueden establecer correspondencias 3D entre los puntos $\mathbf{Q}'_{s,t_{k+1}}$ y $\mathbf{Q}_{s,t_{k+1}}$, que se representa como $(\mathbf{Q}'_{s,t_{k+1}}, \mathbf{Q}_{s,t_{k+1}})$ y quiere decir que idealmente dichos puntos representan la misma posición tridimensional en el instante t_{k+1} del mismo punto en el espacio.

Entonces, de acuerdo a la expresión 3.48, la coincidencia se da cuando se encuentra un $\Delta\mathbf{R}$ y un $\Delta\mathbf{T}$, tal que la resta de los puntos tridimensionales $\mathbf{Q}_{s,t_{k+1}}$ menos los puntos tridimensionales $\mathbf{Q}'_{s,t_{k+1}}$, es el vector nulo, es decir, que cada uno de los puntos $\mathbf{Q}'_{s,t_{k+1}}$ sean idénticos a los respectivos puntos $\mathbf{Q}_{s,t_{k+1}}$, como se aprecia en la figura 3.19.

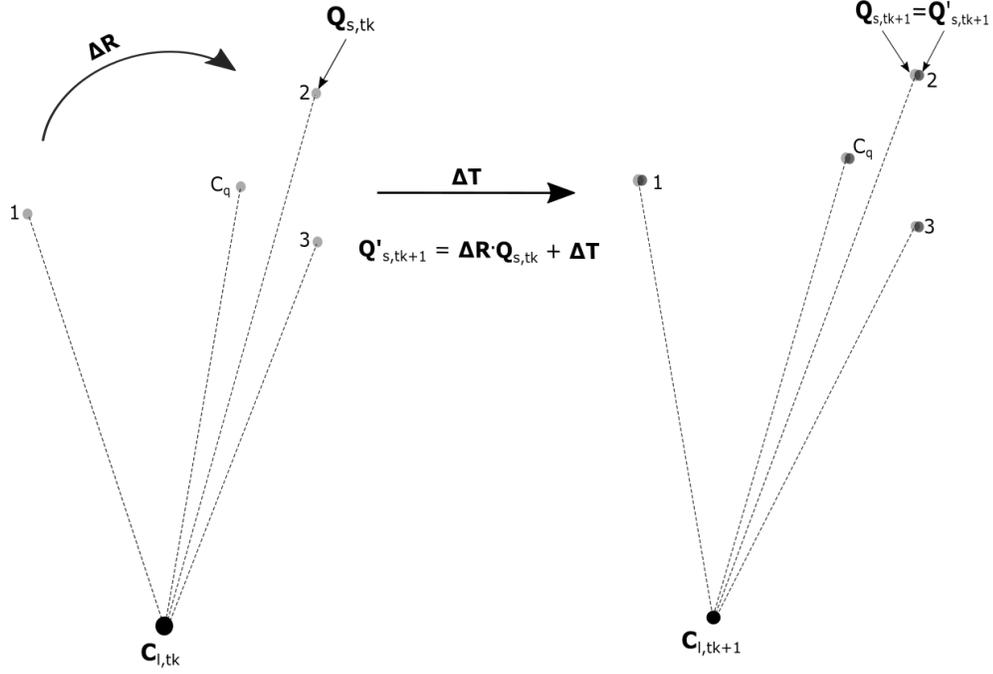


Figura 3.19: Coincidencia ideal entre los puntos tridimensionales \mathbf{Q}_{s,t_k} transformados y los puntos tridimensionales $\mathbf{Q}_{s,t_{k+1}}$, en el instante de tiempo t_{k+1} después del movimiento.

Resulta que solamente en el caso ideal, existe una solución única para los parámetros de movimiento $\Delta\mathbf{R}$ y $\Delta\mathbf{T}$, que satisfacen la expresión 3.48. Esto se debe a que en realidad, las posiciones tridimensionales de los puntos \mathbf{Q}_{s,t_k} y $\mathbf{Q}_{s,t_{k+1}}$ se ven perturbadas por una cierta cantidad de ruido (Matthies, 1989; Arun et al., 2007; Maimone et al., 2007) y por lo tanto, la ecuación 3.48 es igual al vector de error residual dado por \mathbf{e}_s , en lugar del vector nulo $\mathbf{0}$ (ver la figura 3.20):

$$\mathbf{e}_s = \mathbf{Q}_{s,t_{k+1}} - \Delta\mathbf{R} \cdot \mathbf{Q}_{s,t_k} - \Delta\mathbf{T} \quad \forall s \quad (3.50)$$

Si se suma la magnitud al cuadrado de cada uno de los vectores de error residual, definidos por la expresión 3.50 para todo valor de s y se divide entre el número total de dichos vectores, que es igual C_q , el cual es el límite superior de la variable s , se obtiene entonces el error cuadrático medio, que se denota como e^2 :

$$e^2 = \frac{1}{C_q} \sum_{s=1}^{C_q} \mathbf{e}_s^T \mathbf{e}_s \quad (3.51)$$

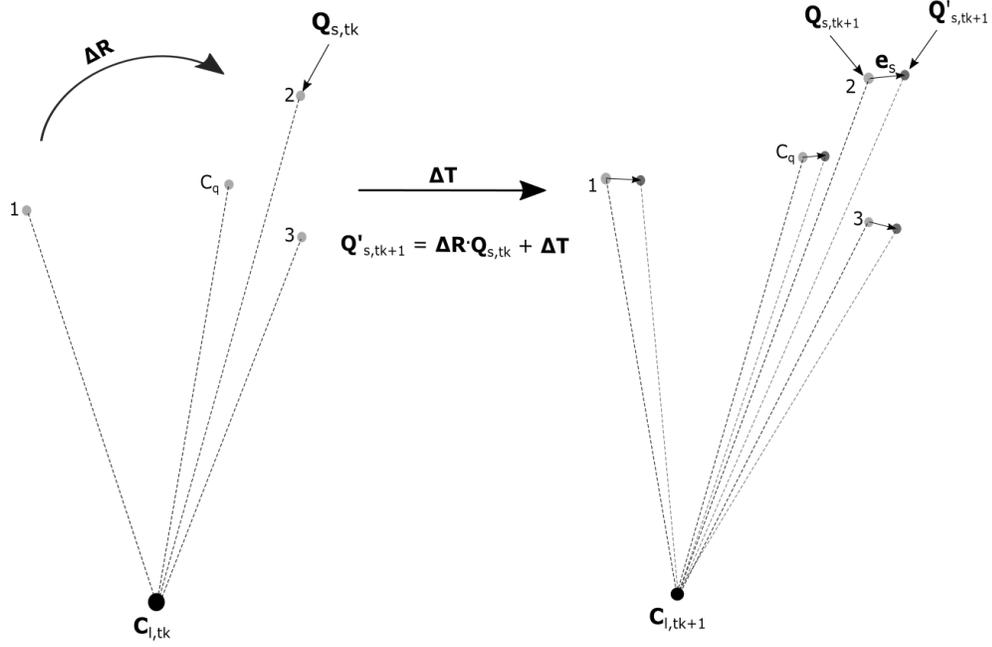


Figura 3.20: Coincidencia con ruido entre los puntos tridimensionales \mathbf{Q}_{s,t_k} transformados y los puntos tridimensionales $\mathbf{Q}_{s,t_{k+1}}$, en el instante de tiempo t_{k+1} después del movimiento.

Al sustituir la expresión 3.50 de los vectores de error residual, en la expresión anterior, se obtiene una expresión equivalente, pero más fácil de entender:

$$e^2 = \frac{1}{C_q} \sum_{s=1}^{C_q} |\mathbf{Q}_{s,t_{k+1}} - (\Delta\mathbf{R} \cdot \mathbf{Q}_{s,t_k} + \Delta\mathbf{T})|^2 \quad (3.52)$$

Como se aprecia en la expresión 3.52, el error cuadrático medio es igual a la suma del cuadrado de las distancias de las correspondencias 3D dadas por $(\mathbf{Q}'_{s,t_{k+1}}, \mathbf{Q}_{s,t_{k+1}})$, es decir, de las distancias entre los puntos tridimensionales $\mathbf{Q}'_{s,t_{k+1}}$ y los respectivos puntos $\mathbf{Q}_{s,t_{k+1}}$, dividida entre el valor de C_q . Entonces, la coincidencia en realidad se da cuando se encuentra un $\Delta\mathbf{R}$ y un $\Delta\mathbf{T}$ que minimicen el error cuadrático medio dado por la expresión 3.52, que equivale a minimizar la suma del cuadrado de las distancias descritas anteriormente, entre el valor de C_q , tal y como se ejemplifica en la figura 3.21.

Cabe destacar, que en la mayoría de algoritmos de odometría visual estereoscópica, como en el algoritmo de Matthies (1989), la coincidencia se logra al encontrar los parámetros de movimiento $\Delta\mathbf{R}$ y $\Delta\mathbf{T}$ que minimicen el error cuadrático sopesado, que se denota como e_w^2 :

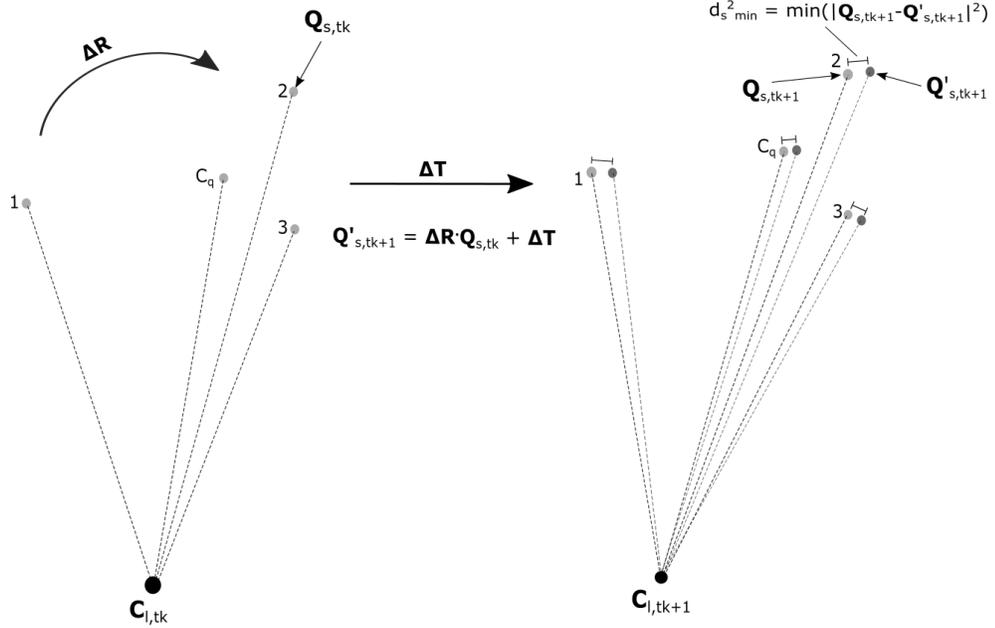


Figura 3.21: Coincidencia con ruido entre los puntos tridimensionales \mathbf{Q}_{s,t_k} transformados y los puntos tridimensionales $\mathbf{Q}_{s,t_{k+1}}$, en el instante de tiempo t_{k+1} después del movimiento.

$$e_w^2 = \sum_{s=1}^{C_q} w_s \mathbf{e}_s^T \mathbf{e}_s = \sum_{s=1}^{C_q} w_s |\mathbf{Q}_{s,t_{k+1}} - (\Delta \mathbf{R} \cdot \mathbf{Q}_{s,t_k} + \Delta \mathbf{T})|^2 \quad (3.53)$$

donde w_s es un factor de peso escalar que refleja la calidad de los puntos tridimensionales \mathbf{Q}_{s,t_k} y $\mathbf{Q}_{s,t_{k+1}}$ y que usualmente se obtiene a partir del modelo del error de posición de dichos puntos tridimensionales, de tal forma que si el valor del factor de peso es igual 1, quiere decir que el error de posición es insignificante y por lo tanto, la calidad de los puntos es excelente y tienen un mayor peso en el error cuadrático sopesado, mientras que si el valor del factor de peso es igual a 0, quiere decir que el error de posición es bastante significativo y por lo tanto, la calidad de los puntos es pésima y tienen un peso nulo en el error cuadrático sopesado. En el algoritmo de Matthies, los factores de peso asociados a cada par de puntos \mathbf{Q}_{s,t_k} y $\mathbf{Q}_{s,t_{k+1}}$, se obtienen de la siguiente forma:

$$w_s = (\det \Sigma_{\mathbf{Q}_{s,t_k}} + \det \Sigma_{\mathbf{Q}_{s,t_{k+1}}})^{-1} \quad (3.54)$$

donde $\Sigma_{\mathbf{Q}_{s,t_k}}$ y $\Sigma_{\mathbf{Q}_{s,t_{k+1}}}$ son las matrices de covarianza de los vectores de error que perturban las posiciones tridimensionales de los puntos \mathbf{Q}_{s,t_k} y $\mathbf{Q}_{s,t_{k+1}}$, respectivamente y que se modelan como vectores aleatorios gaussianos con medias conocidas. Además, dichas matrices de covarianza se obtienen a partir de las matrices de covarianza de los vectores de error que perturban las posiciones bidimensionales de los puntos característicos en la imagen izquierda y en la imagen derecha de cada instante de tiempo, y que también se modelan como vectores aleatorios gaussianos con medias conocidas (Matthies, 1989; Arun et al., 2007). Lo que sucede es que los puntos característicos y sus posiciones tridimensionales están relacionados con la ecuación 2.60 del proceso de triangulación o estimación de profundidad y, por lo tanto, si los puntos característicos tienen un error, sus posiciones tridimensionales también.

En este caso, los parámetros de movimiento $\Delta \mathbf{R}$ y $\Delta \mathbf{T}$ no se estiman minimizando el error cuadrático sopesado, dado por la expresión 3.53, debido a que no se está modelando el ruido que perturba a los puntos característicos \mathbf{q}_{l_s,t_k} y $\mathbf{q}_{l_s,t_{k+1}}$ y por lo tanto, a sus posiciones tridimensionales \mathbf{Q}_{s,t_k} y $\mathbf{Q}_{s,t_{k+1}}$. Es por ello que el movimiento se estima minimizando el error cuadrático medio, dado por la expresión 3.52, que en realidad es exactamente igual a minimizar el error cuadrático sopesado si cada uno de los factores de peso w_s se consideran como una constante igual C_q^{-1} , que se puede sacar fuera de la suma como factor común:

$$e_w^2 = \sum_{s=1}^{C_q} w_s \mathbf{e}_s^T \mathbf{e}_s |_{w_s = \frac{1}{C_q}} \longleftrightarrow e^2 = \frac{1}{C_q} \sum_{s=1}^{C_q} \mathbf{e}_s^T \mathbf{e}_s \quad (3.55)$$

3.4.1 Solución basada en la descomposición en valores singulares

Para encontrar los parámetros de movimiento $\Delta \mathbf{R}$ y $\Delta \mathbf{T}$ que minimizan la expresión 3.55, es decir, que minimizan el error cuadrático medio e^2 , que es la suma entre C_q de las distancias de las correspondencias 3D dadas por $(\mathbf{Q}'_{s,t_{k+1}}, \mathbf{Q}_{s,t_{k+1}})$ o que minimizan el error cuadrático sopesado e_w^2 de dichas distancias cuando cada uno de los factores de peso w_s es igual a C_q^{-1} , se puede optar por el procedimiento estándar que consiste en encontrar la solución diferenciando con respecto a $\Delta \mathbf{R}$ y $\Delta \mathbf{T}$, lo cual conduce a un problema de optimización no lineal. Sin embargo, existe un procedimiento basado en la descomposición de valores singulares de una matriz 3x3, el cual proporciona una solución cerrada de la minimización del error cuadrático medio e^2 y se describe detalladamente en (Arun et al., 1987) y (Horn et al., 1987) ,y, además, fue implementado en el algoritmo Matthies (1989) para minimizar el error cuadrático sopesado e_w^2 . En este caso, se va a emplear el procedimiento descrito en (Umeyama, 1991), que en realidad es prácticamente igual al pro-

cedimiento basado en la descomposición de valores singulares, pero se corrige un error que conduce a resultados erróneos en algunos casos, cuando los datos están gravemente dañados.

La idea detrás del método es buscar una forma de desacoplar la matriz de rotación $\Delta\mathbf{R}$ y el vector de traslación $\Delta\mathbf{T}$, es decir, que cada uno se pueda obtener por separado, ya que la minimización del error cuadrático medio, dado por la expresión 3.52, implica encontrar ambos a la vez mediante un mismo proceso.

Tal y como se demuestra en (Arun et al., 1987), si se supone que $\Delta\mathbf{R}'$ y $\Delta\mathbf{T}'$ son la solución de la minimización del error cuadrático medio e^2 , ecuación 3.52, entonces el centroide del conjunto formado por cada uno de los puntos \mathbf{Q}'_{s,t_k} (\mathbf{Q}_{s,t_k} rotado con $\Delta\mathbf{R}'$ y trasladado con $\Delta\mathbf{T}'$):

$$\mathbf{Q}'_{s,t_k} = \Delta\mathbf{R}' \cdot \mathbf{Q}_{s,t_k} + \Delta\mathbf{T}' \quad \forall s \quad (3.56)$$

que es un vector y se denota como $\mu_{Q'_{t_k}}$, es exactamente igual al centroide del conjunto formado por cada uno de los puntos $\mathbf{Q}_{s,t_{k+1}}$, que también es un vector y se denota como $\mu_{Q_{t_{k+1}}}$:

$$\mu_{Q'_{t_k}} = \mu_{Q_{t_{k+1}}} \quad (3.57)$$

donde:

$$\mu_{Q'_{t_k}} = \frac{1}{C_q} \sum_{s=1}^{C_q} \mathbf{Q}'_{s,t_k} \quad (3.58)$$

$$\mu_{Q_{t_{k+1}}} = \frac{1}{C_q} \sum_{s=1}^{C_q} \mathbf{Q}_{s,t_{k+1}} \quad (3.59)$$

Además, el vector que representa el centroide del conjunto formado por cada uno de los puntos \mathbf{Q}_{s,t_k} es el siguiente:

$$\mu_{Q_{t_k}} = \frac{1}{C_q} \sum_{s=1}^{C_q} \mathbf{Q}_{s,t_k} \quad (3.60)$$

Si se sustituye 3.50 en 3.58 y si se toma en cuenta las expresiones 3.57 y 3.60, se obtiene la siguiente ecuación:

$$\mu_{Q'_{t_k}} = \mu_{Q_{t_{k+1}}} = \Delta\mathbf{R}' \cdot \mu_{Q_{t_k}} + \Delta\mathbf{T}' \quad (3.61)$$

De la cual se puede obtener una expresión para $\Delta\mathbf{T}'$ en términos de los centroides $\mu_{Q_{t_k}}$ y $\mu_{Q_{t_{k+1}}}$ y de la matriz de rotación $\Delta\mathbf{R}'$:

$$\Delta \mathbf{T}' = \boldsymbol{\mu}_{Q_{t_{k+1}}} - \Delta \mathbf{R}' \cdot \boldsymbol{\mu}_{Q_{t_k}} \quad (3.62)$$

En (Arun et al., 1987) y (Horn et al., 1987) se demuestra que si se sustituye la ecuación anterior en la ecuación 3.52, se obtiene una nueva expresión para el error cuadrático medio e^2 , que depende solamente de la rotación $\Delta \mathbf{R}'$ y no de la traslación $\Delta \mathbf{T}'$:

$$e^2 = \frac{1}{C_q} \sum_{s=1}^{C_q} |\mathbf{P}_{s,t_{k+1}} - \Delta \mathbf{R}' \cdot \mathbf{P}_{s,t_k}|^2 \quad (3.63)$$

donde:

$$\mathbf{P}_{s,t_k} = \mathbf{Q}_{s,t_k} - \boldsymbol{\mu}_{Q_{t_k}} \quad \forall s \quad (3.64)$$

$$\mathbf{P}_{s,t_{k+1}} = \mathbf{Q}_{s,t_{k+1}} - \boldsymbol{\mu}_{Q_{t_{k+1}}} \quad \forall s \quad (3.65)$$

Por lo tanto, el problema original de minimizar el error cuadrático medio e^2 , dado por la ecuación 3.52, se divide en dos partes:

- (i) Se encuentra el $\Delta \mathbf{R}'$ que minimiza e^2 en la ecuación 3.63.
- (ii) Luego, se encuentra $\Delta \mathbf{T}'$ a partir de $\Delta \mathbf{R}'$, mediante la ecuación 3.62.

El problema de minimización en (i) se resuelve siguiendo los pasos generales y resumidos del procedimiento basado en la descomposición en valores singulares de una matriz 3x3, los cuales se numeran en (Arun et al., 2007) y se demuestran detalladamente en (Umeyama, 1991; Arun et al., 1987; Horn et al., 1987).

Primero se calculan los centroides $\boldsymbol{\mu}_{Q_{t_k}}$ y $\boldsymbol{\mu}_{Q_{t_{k+1}}}$ de los conjuntos formados por los respectivos puntos tridimensionales \mathbf{Q}_{s,t_k} y $\mathbf{Q}_{s,t_{k+1}}$. Para ello, se utilizan las ecuaciones 3.60 y 3.59, respectivamente. Cabe destacar que, en este caso, los centroides también representan el vector promedio de los respectivos conjuntos.

Una vez que se calculan los centroides $\boldsymbol{\mu}_{Q_{t_k}}$ y $\boldsymbol{\mu}_{Q_{t_{k+1}}}$, se calcula la siguiente matriz 3x3:

$$\Sigma_{Q_{t_k-t_{k+1}}} = \frac{1}{C_q} \sum_{s=1}^{C_q} \mathbf{P}_{s,t_{k+1}} \cdot \mathbf{P}_{s,t_k}^T \quad (3.66)$$

Que en realidad es la matriz de covarianza de los conjuntos formados por los puntos \mathbf{Q}_{s,t_k} y $\mathbf{Q}_{s,t_{k+1}}$, lo cual queda más claro al sustituir 3.64 y 3.65 en la expresión anterior:

$$\Sigma_{Q_{t_k-t_{k+1}}} = \frac{1}{C_q} \sum_{s=1}^{C_q} (\mathbf{Q}_{s,t_{k+1}} - \boldsymbol{\mu}_{Q_{t_{k+1}}}) \cdot (\mathbf{Q}_{s,t_k} - \boldsymbol{\mu}_{Q_{t_k}})^T \quad (3.67)$$

Luego de que se calcula la matriz de covarianza $\Sigma_{Q_{t_k-t_{k+1}}}$, se procede a obtener la descomposición en valores singulares de esta, que consiste en realizar una factorización de dicha matriz en tres matrices diferentes:

$$\Sigma_{Q_{t_k-t_{k+1}}} = \mathbf{U} \cdot \mathbf{D} \cdot \mathbf{V}^T \quad (3.68)$$

donde \mathbf{U} y \mathbf{V} son matrices reales unitarias de tamaño 3×3 y \mathbf{D} es una matriz diagonal con valores reales no negativos de tamaño 3×3 . Las entradas de la diagonal de la matriz \mathbf{D} son los valores singulares de la matriz de covarianza $\Sigma_{Q_{t_k-t_{k+1}}}$, ordenados de mayor a menor. Las columnas de la matriz \mathbf{U} y las columnas de la matriz \mathbf{V} son los vectores singulares izquierdos y los vectores singulares derechos de la matriz de covarianza $\Sigma_{Q_{t_k-t_{k+1}}}$, respectivamente. En (Baker, 2005) y (Press et al., 2007) se explica detalladamente como obtener la descomposición en valores singulares de una matriz cuadrada, así como su significado geométrico y matemático.

Finalmente, tal y como se demuestra en (Arun et al., 1987) y (Horn et al., 1987), la matriz de rotación $\Delta \mathbf{R}'$ que minimiza el error cuadrático medio e^2 de la ecuación 3.63, se puede obtener a partir de las matrices \mathbf{U} y \mathbf{V} :

$$\Delta \mathbf{R}' = \mathbf{U} \cdot \mathbf{V}^T \quad (3.69)$$

Pero resulta que, en algunos casos, la expresión anterior no da una matriz de rotación correcta y en su lugar, da una reflexión, que es cuando el determinante de $\Delta \mathbf{R}'$ es exactamente igual a -1. En (Umeyama, 1991) se soluciona dicho problema al introducir una matriz denotada como \mathbf{S} en el cálculo de $\Delta \mathbf{R}'$, de la siguiente forma:

$$\Delta \mathbf{R}' = \mathbf{U} \cdot \mathbf{S} \cdot \mathbf{V}^T \quad (3.70)$$

donde \mathbf{S} es una matriz diagonal que depende del determinante de la matriz \mathbf{U} y el determinante de la matriz \mathbf{V} :

$$\mathbf{S} = \begin{cases} \mathbf{I}, & \text{si } \det(\mathbf{U}) \cdot \det(\mathbf{V}) = 1 \\ \text{diag}(1, 1, -1), & \text{si } \det(\mathbf{U}) \cdot \det(\mathbf{V}) = -1 \end{cases} \quad (3.71)$$

Para el paso (ii), una vez que se haya calculado la matriz $\Delta \mathbf{R}'$ mediante la ecuación 3.70, se utiliza la ecuación 3.62 para obtener el respectivo vector de traslación $\Delta \mathbf{T}'$, tal y como se muestra en la siguiente expresión:

$$\Delta \mathbf{T}' = \boldsymbol{\mu}_{Q_{t_{k+1}}} - \Delta \mathbf{R}' \cdot \boldsymbol{\mu}_{Q_{t_k}} = \boldsymbol{\mu}_{Q_{t_{k+1}}} - \mathbf{U} \cdot \mathbf{S} \cdot \mathbf{V}^T \cdot \boldsymbol{\mu}_{Q_{t_k}} \quad (3.72)$$

Es de esta forma como finalmente se estima el movimiento del rover, la rotación $\Delta\mathbf{R}'$ y la traslación $\Delta\mathbf{T}'$ que este experimenta entre los instantes de tiempo t_k y t_{k+1} , mediante las ecuaciones dadas por 3.72 y 3.70, respectivamente, que son la solución cerrada del problema de minimización de la suma dividida entre C_q del cuadrado de las distancias de las correspondencias 3D ($\mathbf{Q}'_{s,t_k}, \mathbf{Q}_{s,t_{k+1}}$), las distancias entre los puntos tridimensionales \mathbf{Q}'_{s,t_k} y $\mathbf{Q}_{s,t_{k+1}}$, es decir, son la solución cerrada del problema de minimización del error cuadrático medio e^2 , dado por la ecuación 3.52, la cual se obtuvo mediante un procedimiento basado en el desacople de la traslación y la rotación a partir de centroides, así como la descomposición en valores singulares de una matriz.

3.4.2 Búsqueda de correspondencias típicas mediante RANSAC

En la sección anterior describió un procedimiento o algoritmo para estimar el movimiento del rover o de la cámara estereoscópica atada rígidamente a este, a partir de las correspondencias temporales 3D dadas por ($\mathbf{Q}_{s,t_k}, \mathbf{Q}_{s,t_{k+1}}$). Resulta que dicho algoritmo funciona adecuadamente si todas las correspondencias temporales 2D y 3D que se establecieron son correctas, las cuales se conocen comúnmente como correspondencias típicas.

Hay que recordar que las correspondencias temporales 2D son los pares de puntos ($\mathbf{q}_{l_s,t_k}, \mathbf{q}_{l_s,t_{k+1}}$) y las correspondencias temporales 3D son sus posiciones tridimensionales que forman los pares de puntos ($\mathbf{Q}_{s,t_k}, \mathbf{Q}_{s,t_{k+1}}$). Lo que sucede es que la condición dada por la expresión 3.41, de que el coeficiente de similitud tiene que ser mayor a un valor establecido no es suficiente para garantizar que las correspondencias sean típicas y es por ello que pueden haber correspondencias atípicas entre \mathbf{q}_{l_s,t_k} y $\mathbf{q}_{l_s,t_{k+1}}$ y por lo tanto, entre \mathbf{Q}_{s,t_k} y $\mathbf{Q}_{s,t_{k+1}}$, que pueden ocasionar que algoritmo no funcione bien. Una correspondencia atípica es una correspondencia incorrecta o falsa, de modo que los puntos bidimensionales del par ($\mathbf{q}_{l_s,t_k}, \mathbf{q}_{l_s,t_{k+1}}$) no son las proyecciones en el instante t_k y t_{k+1} del mismo punto en el espacio y por lo tanto, los puntos tridimensionales del par ($\mathbf{Q}_{s,t_k}, \mathbf{Q}_{s,t_{k+1}}$) no son las posiciones en el instante t_k y t_{k+1} del mismo punto en el espacio, tal y como se ejemplifica en la figura 3.22. Sin embargo, existe un método iterativo conocido como RANSAC² que se puede utilizar para encontrar correspondencias típicas y estimar el movimiento a partir de estas. Los pasos generales y resumidos de dicho método se pueden encontrar en (Maimone et al., 2007; Arun et al., 2007).

En el primer paso, se sigue el procedimiento descrito en la sección 3.3, se establecen las correspondencias temporales 2D entre un pequeño subconjunto formado por los puntos \mathbf{q}_{l_s,t_k} que son puntos característicos \mathbf{h}'_{l_i,t_k} elegidos aleatoriamente y un pequeño subconjunto formado por los puntos $\mathbf{q}_{l_s,t_{k+1}}$ que

²El fundamento teórico del algoritmo de RANSAC se puede encontrar en el apéndice B.

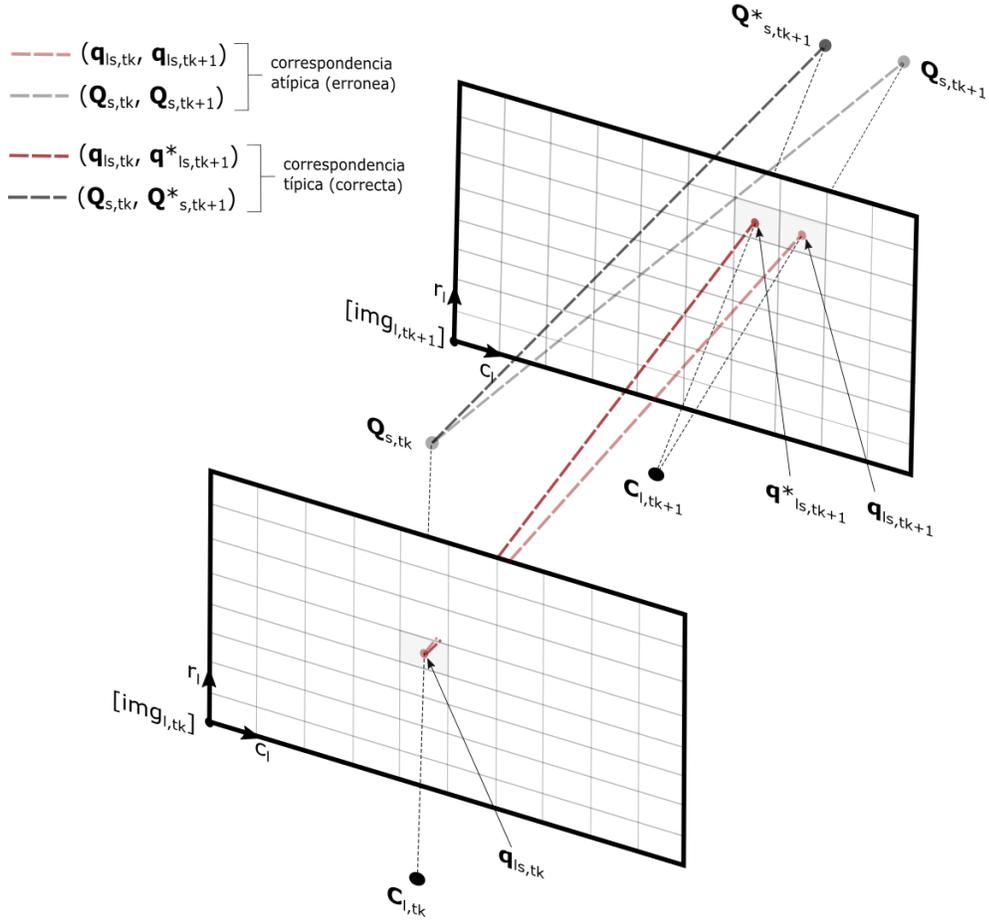


Figura 3.22: Correspondencias temporales 2D típicas y atípicas entre un punto bidimensional en la imagen izquierda antes del movimiento y un punto bidimensional en la imagen izquierda después del movimiento, así como las respectivas correspondencias temporales 3D típicas y atípicas entre sus posiciones tridimensionales.

son puntos característicos $\mathbf{h}'_{lm,t_{k+1}}$ elegidos cuidadosamente, que se representan como $(\mathbf{q}_{ls,t_k}, \mathbf{q}_{ls,t_{k+1}})$. Además, las correspondencias temporales 3D que se representan como $(\mathbf{Q}_{s,t_k}, \mathbf{Q}_{s,t_{k+1}})$ son las posiciones tridimensionales de las correspondencias temporales 2D, que forman subconjuntos de \mathbf{H}_{i,t_k} y $\mathbf{H}_{m,t_{k+1}}$, respectivamente. El tamaño de los subconjuntos es igual a C_q . Seguidamente, se sigue el procedimiento descrito en la sección 3.4.1 para estimar el movimiento del rover entre los instantes t_k y t_{k+1} , es decir, para encontrar la traslación $\Delta\mathbf{T}'$ y la rotación $\Delta\mathbf{R}'$ que minimiza la media de los cuadrados de las distancias de las correspondencias temporales 3D entre $\mathbf{Q}_{s,t_{k+1}}$ y $\mathbf{Q}'_{s,t_{k+1}}$ (\mathbf{Q}_{s,t_k}

rotado y trasladado).

Para el segundo paso, cada uno de los puntos tridimensionales \mathbf{Q}_{s,t_k} se rotan y se trasladan utilizando el movimiento estimado en el paso anterior, $\Delta\mathbf{T}'$ y $\Delta\mathbf{R}'$, tal y como se muestra en la ecuación 3.56 de la sección 3.4.1, que es igual a la siguiente expresión:

$$\mathbf{Q}'_{s,t_{k+1}} = \Delta\mathbf{R}' \cdot \mathbf{Q}_{s,t_k} + \Delta\mathbf{T}' \quad \forall s$$

Seguidamente, cada uno de los puntos tridimensionales $\mathbf{Q}'_{s,t_{k+1}}$ que se obtienen mediante la expresión anterior, se proyectan sobre la imagen izquierda después del movimiento, a partir de sus componentes $(X'_{ls}, Y'_{ls}, Z'_{ls})$ y de las ecuaciones de proyección 2.45 y 2.49 de la sección 2.1.9 del capítulo anterior, de la siguiente manera:

$$\mathbf{q}'_{ls,t_{k+1}} = \begin{pmatrix} c'_{ls} \\ r'_{ls} \end{pmatrix} = f \left(\begin{array}{c} \frac{N_u}{d_u} \frac{X'_{ls}}{Z'_{ls}} + \frac{w}{2} \\ -\frac{1}{d_v} \frac{Y'_{ls}}{Z'_{ls}} + \frac{h}{2} \end{array} \right) \quad (3.73)$$

Cabe destacar, que como los puntos $\mathbf{q}'_{ls,t_{k+1}}$ y $\mathbf{q}_{ls,t_{k+1}}$ son las proyecciones en la imagen después del movimiento de los puntos tridimensionales dados por $\mathbf{Q}'_{s,t_{k+1}}$ y $\mathbf{Q}_{s,t_{k+1}}$, respectivamente, se puede establecer una correspondencia 2D que se representa como $(\mathbf{q}'_{ls,t_{k+1}}, \mathbf{q}_{ls,t_{k+1}})$ y quiere decir que idealmente dichos puntos representan la misma proyección sobre la imagen después del movimiento del mismo punto en el espacio.

Luego, se calcula la distancia en píxeles de cada una de las correspondencias temporales 2D dadas por $(\mathbf{q}'_{ls,t_{k+1}}, \mathbf{q}_{ls,t_{k+1}})$, es decir, la distancia entre los puntos característicos \mathbf{q}'_{ls,t_k} y $\mathbf{q}_{ls,t_{k+1}}$, mediante la siguiente expresión:

$$d_{qs} = |\mathbf{q}'_{ls,t_{k+1}} - \mathbf{q}_{ls,t_{k+1}}| \quad \forall s \quad (3.74)$$

Si el valor de dicha distancia es menor a un umbral dado por $d_{q,max}$, que usualmente es un valor cercano a 0, quiere decir que \mathbf{q}'_{ls,t_k} es aproximadamente igual a $\mathbf{q}_{ls,t_{k+1}}$ y por lo tanto, el punto característico \mathbf{q}_{ls,t_k} (la transformación de este punto es \mathbf{q}'_{ls,t_k}) y el punto característico $\mathbf{q}_{ls,t_{k+1}}$, representan la proyección en la imagen izquierda antes y después del movimiento, respectivamente, del mismo punto en el espacio, de tal forma que la correspondencia temporal 2D dada por $(\mathbf{q}_{ls,t_k}, \mathbf{q}_{ls,t_{k+1}})$ es típica y por lo tanto, la correspondencia temporal 3D $(\mathbf{Q}_{s,t_k}, \mathbf{Q}_{s,t_{k+1}})$ también lo es. Esto se ejemplifica en la figura 3.23, en la cual se muestra el proceso general para determinar si una correspondencia es típica o no.

Entonces, si esta es la iteración número n_q , se le asocia un puntaje que se denota como p_{n_q} , el cual empieza en 0 y se va a aumentar en 1 por cada distancia d_{qs} que sea menor al umbral $d_{q,max}$, tal y como se muestra en la siguiente expresión:

$$\text{si } d_{qs} < d_{q,max} \longrightarrow p_{n_q} = p_{n_q} + 1 \quad (3.75)$$

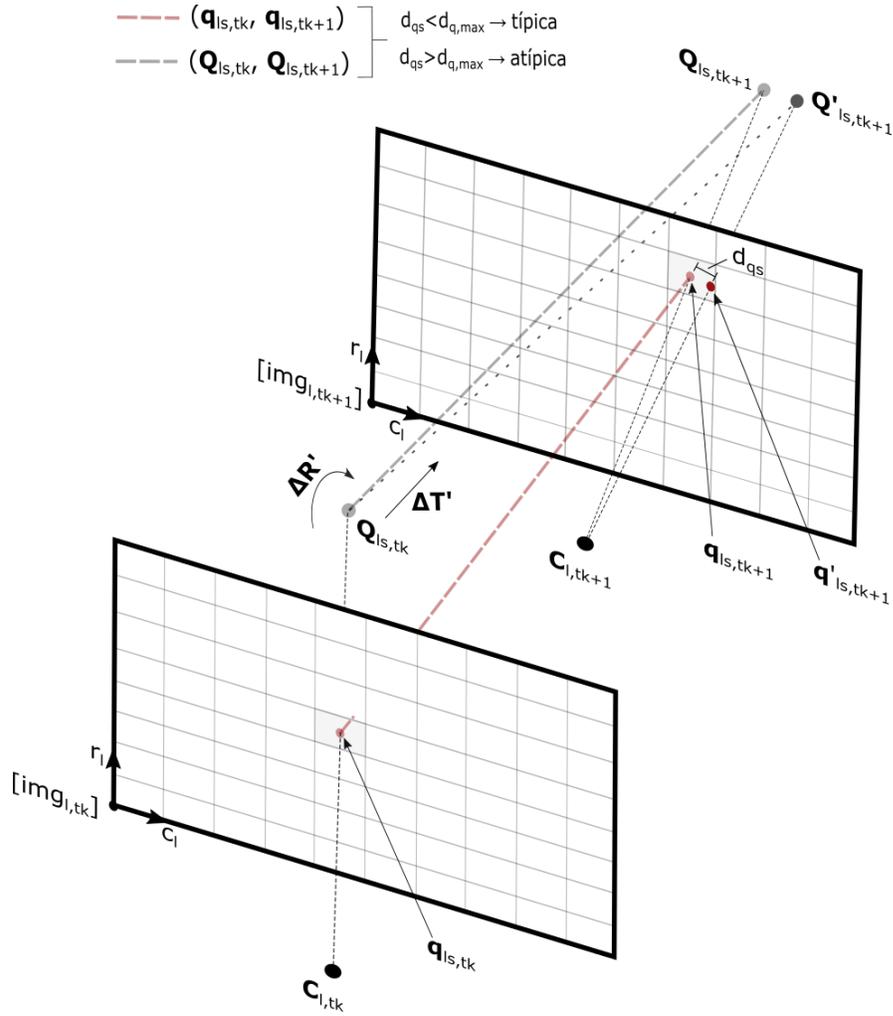


Figura 3.23: Clasificación de correspondencias temporales en típicas o atípicas de acuerdo a la distancia entre los puntos característicos $\mathbf{q}'_{l_s, tk}$ y $\mathbf{q}_{l_s, tk+1}$.

Además, el puntaje p_{n_q} representa la cantidad de correspondencias típicas que se encontraron en la iteración número n_q , donde el máximo número de correspondencias típicas y por lo tanto el máximo valor de p_{n_q} es igual a C_q , que se da cuando cada una de las correspondencias dadas por $(\mathbf{q}_{l_s, tk}, \mathbf{q}_{l_s, tk+1})$ y $(\mathbf{Q}_{l_s, tk}, \mathbf{Q}_{l_s, tk+1})$ es típica, es decir, todas las distancias d_{qs} cumplieron con la condición anterior.

En el tercer paso, se repite el primer y segundo paso un número fijo de iteraciones dado por N_q , es decir, hasta que se llegue a la iteración número $n_q = N_q$. Se eligen los parámetros de movimiento que se obtuvieron en la iteración con la puntuación más alta que se denota como $p_{n_q, max}$.

Además, puede que se dé el caso en el que haya una iteración cuyo puntaje es igual al máximo posible, $p_{n_q} = C_q$. Si se da un caso como ese, simplemente se finaliza el bucle sin importar el número de la iteración n_q :

$$si \ p_{n_q} = C_q \longrightarrow \textit{fin} \quad (3.76)$$

Y entonces, se eligen los parámetros de movimiento que se obtuvieron en esa iteración.

De una u otro forma, los parámetros de movimiento elegidos se denotan como $\Delta \mathbf{R}'_{p_{n_q, max}}$ y $\Delta \mathbf{T}'_{p_{n_q, max}}$, y describen el movimiento del astromóvil entre el instante de tiempo t_k y t_{k+1} con una mayor exactitud que los parámetros $\Delta \mathbf{R}'$ y $\Delta \mathbf{T}'$ que se obtienen sin el método de RANSAC.

Es de esta forma como finalmente se estima el movimiento del astromóvil entre el instante de tiempo t_k y el instante de tiempo t_{k+1} , a partir de correspondencias temporales clasificadas como típicas entre el subconjunto formado por los puntos tridimensionales dados por \mathbf{Q}_{s, t_k} y el subconjunto formado por los puntos tridimensionales $\mathbf{Q}_{s, t_{k+1}}$, minimizando la suma dividida entre C_q del cuadrado de las distancias entre los puntos tridimensionales $\mathbf{Q}_{s, t_{k+1}}$ y los puntos tridimensionales \mathbf{Q}_{s, t_k} rotados y trasladados, es decir, los puntos $\mathbf{Q}'_{s, t_{k+1}}$.

3.5. Implementación del algoritmo

Todo el procedimiento matemático descrito anteriormente, desde la sección 3.2 hasta la sección 3.4, es un algoritmo de odometría visual estereoscópico que cuyo objetivo es estimar el movimiento que experimenta un rover entre un instante de tiempo t_k y un instante de tiempo t_{k+1} , cuando la imagen izquierda de los pares de imágenes capturados en ambos instantes por una cámara estereoscópica que se encuentra atada rígidamente a este, es conocida, así como las nubes de puntos tridimensionales que se obtienen a partir de la triangulación de correspondencias estereoscópicas entre los pares de imágenes para cada instante. Específicamente, el algoritmo encuentra aquella traslación y rotación con la que se tiene que mover un subconjunto de la nube de puntos antes del movimiento (en el instante t_k), elegido aleatoriamente, para que este coincida con un subconjunto de la nube de puntos después del movimiento (en el instante t_{k+1}), elegido cuidadosamente, mediante la minimización de la media de los cuadrados de las distancias de correspondencias temporales clasificadas como típicas entre el primer subconjunto transformado (rotado y trasladado) y el segundo subconjunto, a partir de un método basado en

la descomposición en valores singulares de una matriz que fue utilizado por Matthies (1989). Además, las correspondencias temporales se establecen de acuerdo a un método basado en la correlación pseudo normalizada, que fue propuesto por Moravec (1980) y la clasificación de estas en típicas y atípicas se lleva a cabo mediante el proceso iterativo conocido como RANSAC, que fue propuesto por primera vez por Fischler y Bolles (1981).

Dicho algoritmo se implementó en el lenguaje de programación C, haciendo uso de la plataforma de programación conocida como ROS³. Cabe destacar que todo el algoritmo fue programado desde cero, excepto por el procedimiento para calcular la descomposición en valores singulares de una matriz 3x3, el cual se sabe que es muy complejo. Es por ello que se utilizó una rutina conocida como *svdcmp*, que se extrajo del libro llamado *Numerical Recipes* (Press et al., 2007) y que fue programada en C. Dicha rutina calcula la descomposición en valores singulares de cualquier matriz cuadrada NxN.

Resulta que la nube de puntos antes y después del movimiento, así como la imagen izquierda antes y después del movimiento no son las únicas entradas del algoritmo, es decir, no son los únicos datos de los cuales este depende. Este también depende de los valores de los parámetros de control, que son variables que se fueron definiendo a lo largo de las secciones anteriores y que son vitales para la afinación y funcionamiento del algoritmo, es decir, que repercuten directamente en la salida de este. Las variables que son consideradas como parámetros de control dentro del algoritmo, son las siguientes:

- w y h : ancho y alto de las imágenes capturadas por la cámara estereoscópica.
- Cq : cantidad de correspondencias temporales 2D y 3D que se quieren establecer entre los puntos característicos y sus posiciones tridimensionales antes y después del movimiento. También es la cantidad mínima de correspondencias típicas dentro del método de RANSAC, que se conoce como el consenso aceptable. Su valor es un número entero puede ser mayor o igual a 6.
- w_c y h_c : ancho y alto de las ventanas de correlación que se establecen en la imagen izquierda antes y después del movimiento. Sus valores tienen que ser enteros impares y menor a los valores de w y h .
- w_b y h_b : ancho y alto de la ventana de búsqueda que se establece en la imagen izquierda después del movimiento. Sus valores tienen que ser enteros impares y mucho mayores a los valores de w_c y h_c .

³Un mayor desarrollo sobre el funcionamiento de ROS se puede encontrar en el apéndice A.

- cpn_{min} : coeficiente de similitud mínimo entre las ventanas de correlación antes y después del movimiento, para poder establecer una correspondencia temporal y que sirve como primer paso para rechazar correspondencias atípicas. Su valor es un número decimal que se escoge usualmente entre 0.8 y 0.95.
- dq_{max} : distancia máxima de una correspondencia temporal 3D proyectada en la imagen después del movimiento. Se le conoce como el umbral de distancia dentro del método de RANSAC y sirve para clasificar las correspondencias temporales como típicas o atípicas. Su valor es un número decimal que se escoge usualmente entre 0.2 y 0.7 píxeles.
- N_q : número máximo de iteraciones dentro del método de RANSAC. Su valor es un número entero que depende mucho de qué tan corrompidos se encuentren los datos (nubes de puntos e imágenes).

Además, los valores de cada uno de los parámetros de control que se mencionaron anteriormente, se establecen de forma totalmente externa al algoritmo, mediante un archivo de texto que se denota como *parametros_de_control.txt* y es considerado como una entrada del algoritmo, a parte de las nubes de puntos y las imágenes de la cámara izquierda. Dicho archivo de texto tiene el siguiente formato general:

```

Imagen y ventanas:
Ancho_de_la_imagen(w): valor1
Largo_de_la_imagen(h): valor2
Ancho_de_la_ventana_de_correlacion(wc): valor3
Largo_de_la_ventana_de_correlacion(hc): valor4
Ancho_de_la_ventana_de_busqueda(wb): valor5
Largo_de_la_ventana_de_busqueda(hb): valor6

Coeficiente de similitud y algoritmo de RANSAC:
Coeficiente_de_similitud_minimo(cpn_min): valor7
Consenso aceptable_o_cantidad_de_correspondencias(Cq): valor8
Umbral_de_distancia(dq_max): valor9
Numero_de_iteraciones(Nq): valor10

```

Cabe destacar que cada uno de los valores de los parámetros de control se establecen y se justifican detalladamente en el siguiente capítulo.

Por otro lado, la salida o el resultado final del algoritmo implementado, es la traslación y rotación que experimento el astromóvil entre dos instantes

de tiempos consecutivos. Específicamente, las componentes del vector de traslación y las entradas de la matriz de rotación, las cuales se guardan en un archivo de texto que se denota como *parametros_de_movimiento_estimados.txt* y que tiene el siguiente formato general:

```

Parámetros de movimiento estimados:
Vector de traslación (T):
  TX TY TZ
Matriz de rotación (R):
  r1 r2 r3
  r4 r5 r6
  r7 r8 r9

```

El algoritmo implementado visto desde ROS es un nodo llamado *motion_estimation_node* y su funcionamiento depende totalmente de los tres nodos de ROS de código abierto descritos en la sección 2.2.5 y explicados de forma detallada en el apéndice A. También depende de los valores de los parámetros de control que se encuentran en el archivo de texto llamado *parametros_de_control.txt*, tal y como se mencionó anteriormente.

En la figura 3.24 se muestra un diagrama general del algoritmo completo, el cual está conformado por cuatro nodos: los tres nodos de ROS que se utilizan para obtener las nubes de puntos y las imágenes antes y después del movimiento y el nodo del algoritmo de odometría visual estereoscópico implementado, el cual estima el movimiento del rover a partir de las nubes e imágenes y de los valores de los parámetros de control. En dicho diagrama también se muestra la comunicación entre los cuatro nodos, así como las salidas y entradas del algoritmo. A continuación, se explica de formada detallada el diagrama del algoritmo completo mediante los siguientes pasos:

1. El nodo llamado *camera1394stereo* toma las imágenes crudas de intensidad capturadas por la cámara estereoscópica antes del movimiento y las publica como un mensaje a través del tópico */image_raw*.
2. Se ejecuta el nodo *camera_calibration* que se suscribe al tópico */image_raw* para obtener las imágenes crudas de intensidad antes del movimiento, a partir de las cuales calcula los parámetros intrínsecos y extrínsecos de la cámara estereoscópica y los guarda en un archivo *.yaml* dentro de una carpeta interna de ROS llamada *camera_info*.
3. Se ejecuta el nodo *stereo_image_proc* el cual también se suscribe al tópico */image_raw* para obtener las imágenes crudas de intensidad antes del movimiento y seguidamente. Luego, accede a los parámetros que fueron creados por el nodo anterior y guardados en la carpeta interna llamada

camera_info, para a partir de estos calibrar y rectificar las imágenes, las cuales las publica a través del tópico */image_rect*. Además, calcula la nube de puntos antes del movimiento, a partir de las imágenes rectificadas y calibradas, mediante el método de triangulación de correspondencias estereoscópicas, y la publica en el tópico llamado */points2*.

4. Todo el proceso anterior se repite para el instante después del movimiento del rover, de tal forma que en el tópico */image_rect* se publiquen las imágenes rectificadas y calibradas después del movimiento y en el tópico */points2* la nube de puntos después del movimiento.
5. Se ejecuta el nodo *motion_estimation_node* que se suscribe a los tópicos */image_rect* y */points2* para obtener y guardar la imagen izquierda y la nube de puntos antes y después del movimiento del astromóvil y además, lee el archivo de texto *parametros_de_control.txt* para guardar los valores previamente establecidos de los parámetros de control. Seguidamente, a partir de los datos de entrada anteriores, realiza todos los cálculos y procedimientos necesarios para estimar el movimiento del rover. Por último, crea un archivo de texto llamado *parametros_de_movimiento_estimados.txt*, el cual guarda los parámetros de movimiento estimados (vector de traslación y matriz de rotación).

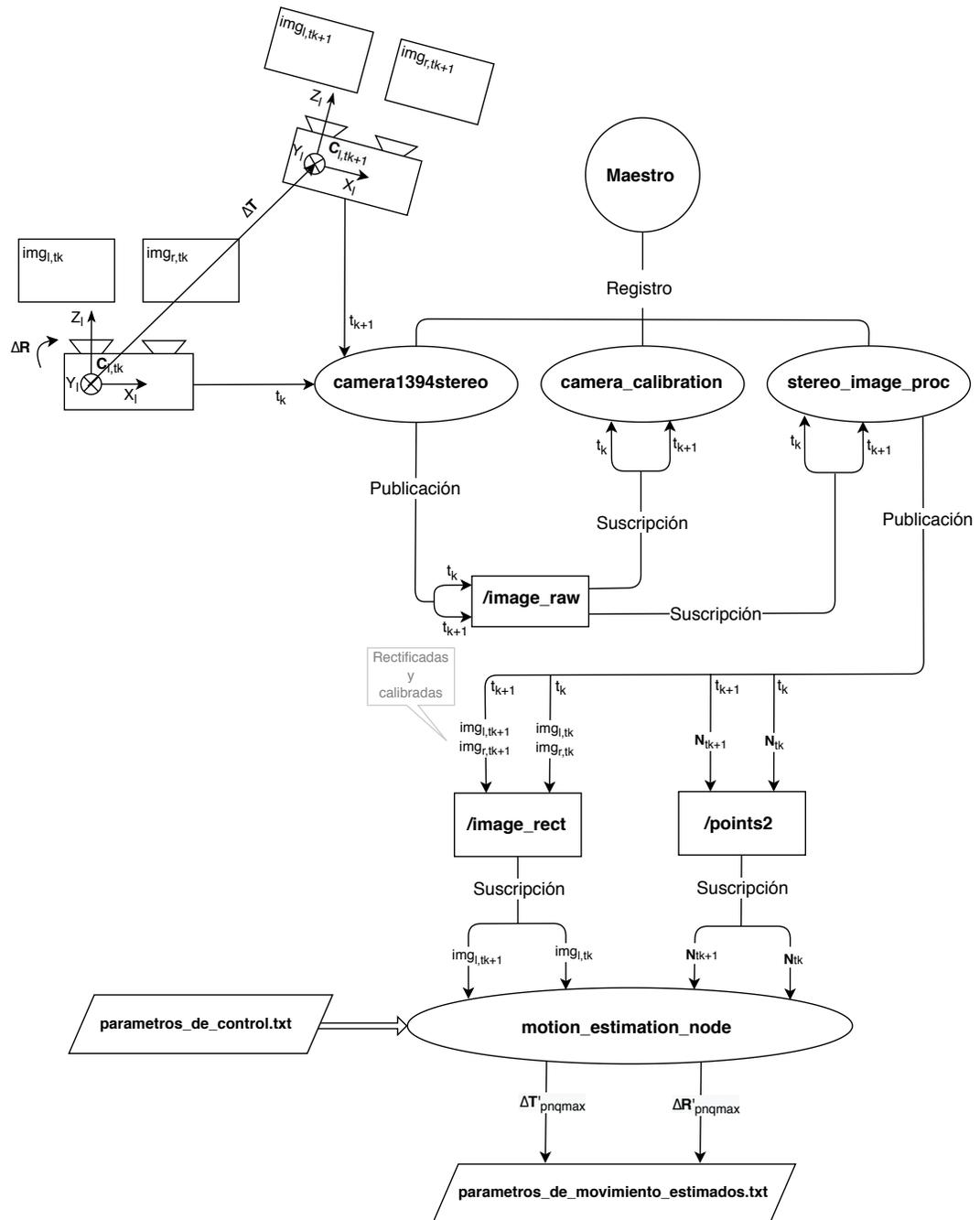


Figura 3.24: Diagrama general del algoritmo completo donde se muestra la comunicación entre todos los nodos de ROS, así como las salidas y entradas del algoritmo o nodo principal.

4 Pruebas experimentales

4.1. Obtención de los datos de entrada de prueba

Para probar el correcto funcionamiento del algoritmo de odometría visual estereoscópico implementado, así como su precisión y exactitud bajo ciertas circunstancias, se optó por hacerlo de forma totalmente controlada, en lugar de hacerlo en tiempo real, es decir, se utilizan datos de entrada de prueba, los cuales se deben de obtener con anterioridad a la ejecución de las pruebas, en caso de que se adquieren a partir del mundo real o se deben de obtener antes o durante la ejecución de cada prueba, en caso de que se generan de forma sintética.

En este caso, los datos de entrada de prueba se pueden dividir en dos grupos. Al primer grupo se le conoce como los datos de entrada de prueba antes del movimiento y está conformado por la imagen de intensidad izquierda y la respectiva nube de puntos antes del movimiento del rover. Análogamente, al segundo grupo se le conoce como los datos de entrada de prueba después del movimiento y está conformado por la imagen de intensidad izquierda y la respectiva nube de puntos después del movimiento del rover.

Los datos de entrada de prueba antes del movimiento se adquieren a partir del mundo real, específicamente, mediante nodos de ROS, los cuales se han explicado y mencionado en capítulos anteriores, que proporcionan la imagen de intensidad izquierda y la respectiva nube de puntos, que son una representación bidimensional y tridimensional, respectivamente, de la superficie regular o plana que está siendo capturada por una cámara estereoscópica, atada rígidamente al costado izquierdo de una plataforma de desarrollo robótico real e inclinada de tal forma que el plano de la cámara izquierda y derecha quede paralelo a la superficie por la cual se puede llegar a mover dicha plataforma.

Por otro lado, los datos de entrada después del movimiento, se generan de forma totalmente sintética durante la ejecución de cada prueba y a partir de los datos de entrada antes del movimiento, es decir, mediante fórmulas, ecuaciones y transformaciones matemáticas que se aplican a la imagen izquierda y a la nube de puntos antes del movimiento, que se obtuvieron a partir del mundo real y con anterioridad a la ejecución de las pruebas, tal y como se describió en el párrafo anterior. Resulta que como el proceso para obtener dichos datos forma parte de la ejecución de cada prueba, no se explicara en esta sección, en cambio, se explicara de forma detallada más adelante, cuando se hable sobre el planteamiento de cada una de las pruebas que se llevaron a cabo.

Entonces, esta sección se centra solamente en la obtención de los datos de entrada de prueba antes del movimiento y en los aspectos fundamentales que se deben de tomar en cuenta para ello, como la marca, modelo y características del equipo utilizado (cámara estereoscópica y plataforma robótica), así como la configuración y adaptación del entorno que rodea a dicho equipo y los factores extrínsecos que se pueden controlar y modificar.

4.1.1 Equipo utilizado

Para obtener los datos de entrada de prueba antes del movimiento, se utilizó la cámara estereoscópica que se observa en la fotografía de la figura 4.1, la cual es una cámara Bumblebee 2 de la marca Point Grey, específicamente, el modelo BB2-03S2M-60. Es una cámara rectificada que ha sido pre-calibrada contra la distorsión y la desalineación, lo cual implica que está compuesta por dos lentes idénticos, uno izquierdo y uno derecho, con distancias focales iguales y que se encuentran totalmente alineados, es decir, sus ejes ópticos son paralelos y la línea base que une sus centros ópticos es paralela a las filas de las imágenes o al eje horizontal.



Figura 4.1: Cámara estereoscópica Bumblebee 2 de la marca Point Grey.

La Bumblebee 2 proporciona un equilibrio entre la calidad de los datos, la velocidad de procesamiento, el tamaño y el precio. De acuerdo a su hoja técnica, que se puede encontrar en (PointGrey, 2009), sus características y especificaciones más sobresalientes son las siguientes:

- Tipo de sensor de imagen: Sony ICX424, CCD de barrido progresivo de $1/3''$.
- Distancia entre sensores a lo largo del eje horizontal y vertical: $7.4 \mu m$.
- Numero de píxeles por sensor a lo largo del eje horizontal: 0.97117.
- Numero de píxeles por sensor a lo largo del eje vertical: 1

- Máxima resolución: 648x488.
- Máxima cantidad de cuadros por segundo: 48.
- Distancia de la línea base entre ambos lentes: 12 *cm*.
- Distancia focal de ambos lentes: 6 *mm*.
- Interfaz: IEEE-1394a de 6 pines para control de cámara y transmisión de datos de vídeo.
- Temperatura de operación: de 0° a 45°C.
- Relación señal a ruido: 60dB.

Además, dicha cámara se sujetó rígidamente a un costado de la plataforma de desarrollo robótico real, tal y como se observa en la fotografía de la figura 4.2, la cual es una plataforma Husky A200, de la marca Clearpath Robotics. Su gran capacidad de carga útil y sus sistemas de energía se adaptan a una vasta cantidad de aplicaciones y funciones personalizadas para satisfacer las necesidades de investigación. Se pueden integrar múltiples dispositivos y accesorios como cámaras estereoscópicas, LIDARs, GPSs, IMUs y muchos más. Además, es totalmente compatible con ROS.

De acuerdo a la página oficial de Clearpath Robotics, que se puede encontrar en (Clearpath, 2020), las características más sobresalientes del Husky A200 son las siguientes:

- Fácil de usar: fue la primera plataforma robótica que admitió ROS desde su configuración de fábrica y por lo tanto, existe una gran base de conocimientos en la prospera comunidad de ROS, que ayuda producir resultados de investigación de forma más rápida.
- Resistente y todo terreno: tiene un diseño simple, construido con materiales duraderos y con muy pocas partes móviles. Además, sus grandes neumáticos con tacos permiten su uso en múltiples terrenos.
- Un referente de confianza: se han publicado incontables artículos de investigación que han utilizado lo han utilizado como plataforma de prueba.
- Control de precisión: tiene un controlador que se puede ajustar para obtener perfiles de movimiento bastante suaves y con un excelente rechazo a perturbaciones.
- Personalizable: se puede integrar una gran variedad de cargas útiles, es decir, una combinación de múltiples accesorios y dispositivos.



Figura 4.2: Plataforma de desarrollo robótico Husky A200 de la marca Clear-path Robotics.

Además, sus especificaciones técnicas más notables son:

- Máxima carga útil: 75 Kg.
- Máxima velocidad: 1.0 m/s.
- Grado de ascenso máximo: 45° .
- Grado transversal máximo: 30° .
- Duración de la batería - uso nominal: 3 horas.
- Tiempo de carga: 4 horas.
- Grado de protección: IP44.

Cabe destacar que tanto la cámara estereoscópica Bumblebee 2, como la plataforma de desarrollo robótico Husky A200, son propiedad del IPCV-LAB de la Escuela de Ingeniería Eléctrica de la Universidad de Costa Rica.

4.1.2 Entorno controlado de pruebas

Los datos de entrada de prueba antes del movimiento se adquirieron en un entorno controlado, es decir, un espacio del mundo real definido para tal fin, en el cual se encuentre la cámara estereoscópica y la plataforma robótica y que además, se puedan controlar y modificar una serie de factores extrínsecos que influyen directamente en la calidad de los datos que se puedan llegar a obtener.

En la figura 4.3 se muestra el entorno controlado dentro del cual se obtuvieron los datos de entrada de prueba. Dicho entorno es un espacio dentro del Laboratorio de Investigación en Procesamiento de Imágenes y Visión por Computador (IPCV-LAB), el cual se configuro y se adaptó con el único fin de obtener una imagen izquierda y una nube de puntos de que describen de forma bidimensional y tridimensional, respectivamente, la superficie o suelo dentro del campo de visión de la cámara estereoscópica Bumblebee 2, que está sujeta rígidamente a un costado izquierdo de la plataforma robótica Husky A200, mediante un soporte de brazo que permite seis grados de libertad de movimiento.



Figura 4.3: Espacio controlado de pruebas, ubicado dentro del laboratorio IPCV-LAB de la Universidad de Costa Rica.

En ese entorno existen cinco factores que se pueden controlar y modificar:

el tipo de superficie, el tipo de iluminación, la inclinación horizontal de la cámara, la inclinación vertical de la cámara y la altura de la cámara respecto al suelo. Dichos factores se fijaron de tal forma que se obtuvieran datos correctos, es decir, que representaran de forma adecuada la superficie capturada por la cámara estereoscópica:

- Tipo de superficie: se utilizó una fachaleta cerámica con acabado de piedra, la cual se ubicó en el suelo de tal forma que abarcara todo el campo de visión de la cámara, como se aprecia en la figura 4.4, para así simular la superficie rocosa de un planeta de exploración como Marte. Además, este tipo de superficies tienen muchas esquinas que facilitan la búsqueda de puntos característicos y por lo tanto, aumenta la calidad de los datos.
- Tipo de iluminación: se utilizó la técnica de iluminación difusa, se colocaron varios focos proyectores LED alrededor de la plataforma robótica, como el que se observa en la figura 4.5, de tal forma que iluminaran la superficie de forma indirecta para conseguir una iluminación suave y difusa. De esta forma se redujeron las sombras y los brillos en la superficie que podían llegar a afectar la calidad de los datos.
- Inclinación horizontal: la inclinación de la cámara con respecto a la horizontal se fijó en aproximadamente 0° , como se observa en la figura 4.6. Para ello se utilizó un nivel de burbuja (instrumento utilizado para determinar la verticalidad y horizontalidad de un objeto).
- Inclinación vertical: la inclinación de la cámara con respecto a la vertical se fijó en aproximadamente 0° , como se observa en la figura 4.7. También se utilizó un nivel de burbuja. Cabe destacar que con una inclinación vertical y horizontal de 0° se garantiza que el plano de la cámara quede paralelo a la superficie. La principal razón de esto, es para facilitar la transformación de la imagen izquierda antes del movimiento, para obtener de forma sintética una imagen izquierda después del movimiento, proceso que se explicara luego.
- Altura: la altura de la cámara respecto al suelo se fijó en 80 centímetros, como se puede ver en la figura 4.8, donde se utilizó una cinta métrica para la medición. En este caso particular, la altura no influye en la obtención de los datos de entrada.



Figura 4.4: Superficie utilizada dentro del campo de visión de la cámara estereoscópica atada rígidamente a la plataforma robótica dentro del entorno controlado.



Figura 4.5: Foco proyectores LED usados para iluminar de forma difusa el entorno controlado.



Figura 4.6: Inclinación horizontal de la cámara estereoscópica atada rígidamente a la plataforma robótica dentro del entorno controlado.



Figura 4.7: Inclinación vertical de la cámara estereoscópica atada rígidamente a la plataforma robótica dentro del entorno controlado.



Figura 4.8: Altura con respecto al suelo de la cámara estereoscópica atada rígidamente a la plataforma robótica dentro del entorno controlado.

4.1.3 Imagen de intensidad izquierda y nube de puntos de prueba

Una vez que se configuro el entorno controlado de prueba y se fijaron los factores extrínsecos de este, se obtuvieron los datos de entrada de prueba antes del movimiento, es decir, la imagen de intensidad izquierda y la nube de puntos de prueba antes del movimiento. Para ello, se empleó el proceso que se basa en múltiples nodos de ROS, el cual se ha explicado y mencionado en capítulos anteriores. Además, se programó un pequeño algoritmo que se suscribiera a los respectivos tópicos, */image_rect* y */points2*, para poder obtener y guardar la imagen izquierda y la nube de puntos, respectivamente, que fueran publicadas por el proceso de ROS para un instante dado.

Entonces, la imagen izquierda de prueba que se obtuvo es la que se observa en la figura 4.9. Es una imagen de intensidad en formato BMP (imagen de mapa de bits) con una resolución de 640x480, que se guardó con el nombre de *intensity1.bmp*.

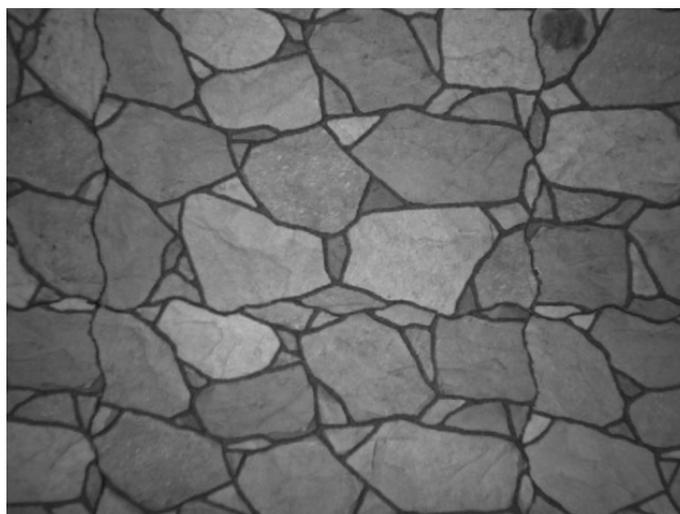


Figura 4.9: Imagen de intensidad izquierda de prueba.

Por otro lado, la nube de puntos que se obtuvo se guardó en un archivo de texto con el nombre *Nube1.txt*, el cual contiene lo siguiente:

- La cantidad total de píxeles que tiene la imagen de intensidad izquierda, es decir, el largo por el ancho de la imagen.
- Una lista con todas las posiciones tridimensionales respecto al sistema de coordenadas de la cámara izquierda, dadas en centímetros, de cada uno de los píxeles que conforman la imagen de intensidad izquierda de la figura . Dicha lista empieza con la posición tridimensional del píxel

ubicado en la esquina inferior izquierda y finaliza con la posición tridimensional del píxel ubicado en la esquina superior derecha. Pero, como se ha mencionado anteriormente, hay muchos píxeles que no son un punto característico y por lo tanto, no tienen una posición tridimensional asociada, en esos casos la posición se pone como NaN (del inglés *not a number*) y no forma parte de la nube de puntos como tal.

Entonces, el archivo *Nube1.txt* tiene el siguiente formato general:

wxh	(cantidad de píxeles)
X1 Y1 Z1	(posición tridimensional del píxel 1)
NaN NaN NaN	(el píxel 2 no es un punto característico)
X3 Y3 Z3	(posición tridimensional del píxel 3)
X4 Y4 Z4	(posición tridimensional del píxel 4)
NaN NaN NaN	(el píxel 5 no es un punto característico)
X6 Y6 Z6	(posición tridimensional del píxel 6)
.	.
.	.
.	.
Xwh Ywh Zwh	(posición tridimensional del píxel wxh)

4.2. Planteamiento de las pruebas

Para poder comprobar la funcionalidad del algoritmo de odometría visual estereoscópico implementado, así como su precisión y exactitud, se plantearon y se llevaron a cabo varias pruebas que utilizaron únicamente como entrada o punto de partida la imagen de intensidad izquierda y la nube de puntos que se obtuvieron en un entorno controlado del mundo real, tal y como se describió anteriormente y que se consideran como los datos de entrada de prueba antes del movimiento. Dichas pruebas someten al algoritmo a distintas circunstancias y situaciones generadas de forma controlada y sintética, gracias a que los datos de entrada de prueba después del movimiento, es decir, la imagen de intensidad izquierda y la nube de puntos después del movimiento, se obtienen manipulando los datos de entrada de prueba antes del movimiento, mediante transformaciones y ecuaciones matemáticas conocidas, cuyos parámetros se pueden manipular fácilmente y siempre serán conocidos.

Las pruebas se delimitaron teniendo en cuenta que el algoritmo está constituido por tres procesos primordiales, obviando el proceso externo basado en los nodos de ROS. Dichos procesos son los siguientes:

1. El establecimiento de correspondencias temporales 2D entre un subconjunto de los puntos característicos en la imagen de intensidad izquierda

antes del movimiento, elegido de forma aleatoria y un subconjunto de los puntos característicos en la imagen de intensidad izquierda después del movimiento, elegidos cuidadosamente, mediante el uso de la pseudo correlación normalizada como una medida de similitud. También se establecen las correspondencias temporales 3D, que son las posiciones tridimensionales de las correspondencias temporales 2D, que forman una correspondencia entre un subconjunto de las nubes de puntos antes del movimiento y un subconjunto de la nube de puntos después del movimiento.

2. La estimación de movimiento entre dos instantes de tiempo consecutivos, encontrando aquella traslación y rotación con la que se debe de mover la nube de puntos antes del movimiento, para que esta coincida con la nube de puntos después del movimiento, donde la coincidencia se logra minimizando un error cuadrático medio que depende de las correspondencias temporales 3D entre subconjuntos de las nubes de puntos, utilizando un procedimiento basado en la descomposición en valores singulares de un matriz. Específicamente, se minimiza la media de los cuadrados de las distancias de correspondencias temporales 3D entre el subconjunto de la primera nube de puntos transformado (rotado y trasladado) y el subconjunto de la segunda nube de puntos.
3. La iteración de los procesos anteriores mediante un procedimiento conocido como RANSAC, para estimar los parámetros de movimiento entre dos instantes de tiempo consecutivos, a partir de correspondencias temporales clasificadas como típicas, es decir, se excluye de la estimación las correspondencias temporales clasificadas como atípicas, que puede generar resultados erróneos o incongruencias. Cabe destacar que este proceso en realidad es el algoritmo completo, pero excluyendo la parte de ROS.

Entonces, para corroborar el correcto funcionamiento y la precisión de cada uno de los tres procesos primordiales que conforman el algoritmo implementado, los cuales se describieron anteriormente, se plantearon y se llevaron a cabo un total de tres pruebas, es decir, una para cada proceso. Entonces, el propósito de esta sección es explicar y describir de forma detallada el planteamiento y los pasos de cada una de esas pruebas.

4.2.1 Prueba I: establecimiento de correspondencias temporales

El objetivo principal de esta prueba es probar el correcto funcionamiento y determinar la precisión del proceso del algoritmo de odometría visual estereoscópico implementado, encargado de establecer correspondencias temporales

entre los puntos característicos de dos imágenes de intensidad que fueron capturadas en diferentes instantes de tiempo (antes y después del movimiento), mediante un procedimiento basado en la correlación pseudo normalizada. Esta prueba tiene un total de ocho pasos, los cuales se explican a continuación.

Primero, se lee la imagen de intensidad izquierda de prueba antes del movimiento llamada *intensity1.bmp*. Seguidamente, se guardan los valores de intensidad I_{l_j, t_k} de cada uno de los píxeles $\mathbf{p}_{l_j, t_k}(c_{l_j}, r_{l_j})$ de dicha imagen, desde el primer píxel en la esquina inferior izquierda, hasta el último píxel en la esquina superior derecha. Cabe destacar que al ancho w de dicha imagen es 640 y el largo h es 480.

Luego, en el segundo paso, se obtiene la imagen de intensidad izquierda después del movimiento, de manera totalmente sintética. Para ello, se rota y se traslada la imagen izquierda antes del movimiento, a partir de una transformación proyectiva plana, que solamente es posible cuando la imagen a la que se le aplica esta transformación se obtiene a partir de una cámara cuyo plano se encuentra paralelo a la superficie (Martínez, 2019b; Hartley y Zisserman, 2004). Es por esa razón que la cámara estereoscópica Bumblebee 2 se colocó de tal forma que su plano fuera paralelo al plano que describe el suelo del entorno controlado.

Existen varias clases de transformaciones proyectivas planas que dependen de la cantidad de grados de libertad. En este caso se utilizó la clase I que se conoce como isometría o transformación Euclidiana y que solamente tiene 3 grados de libertad: una rotación dada por un ángulo de θ grados alrededor de un eje perpendicular a la imagen y que pasa por su centro, seguida de una traslación $(t_c, t_r)^T$ dada en píxeles, a lo largo del eje horizontal c y del eje vertical r de la imagen, respectivamente (Martínez, 2019b; Hartley y Zisserman, 2004).

Entonces, para rotar y trasladar los píxeles \mathbf{p}_{l_j, t_k} de la imagen izquierda antes del movimiento, se emplea la transformación Euclidiana, mediante la siguiente ecuación vectorial (Martínez, 2019b; Hartley y Zisserman, 2004):

$$\mathbf{p}_{ln, t_{k+1}} = \mathbf{K} \mathbf{p}_{lj, tk} \quad (4.1)$$

donde $\mathbf{p}_{ln, t_{k+1}}(c_{ln}, r_{ln})$ son los píxeles de la imagen izquierda después del movimiento que tienen un valor de intensidad asociado, dado por $I_{ln, t_{k+1}}$ que es igual a I_{lj, t_k} y \mathbf{K} es la matriz de transformación, dada por la siguiente expresión (Martínez, 2019b; Hartley y Zisserman, 2004):

$$\mathbf{K} = \begin{pmatrix} \cos(\theta) & -\sin(\theta) & [1 - \cos(\theta)]\frac{w}{2} + \sin(\theta)\frac{h}{2} + t_c \\ \sin(\theta) & \cos(\theta) & [1 - \cos(\theta)]\frac{h}{2} - \sin(\theta)\frac{w}{2} + t_r \end{pmatrix} \quad (4.2)$$

Pero resulta que el proceso de rotar y trasladar una imagen o de transformarla, es decir, el proceso que consiste en obtener la imagen después del

movimiento a partir de una transformación euclidiana de la imagen antes del movimiento, no es tan simple como solo aplicar la ecuación 4.1 a cada uno de los píxeles \mathbf{p}_{l_j,t_k} de la imagen antes del movimiento, para así obtener los píxeles $\mathbf{p}_{l_n,t_{k+1}}$ de la imagen después del movimiento y copiar el valor de intensidad I_{l_j,t_k} de cada uno de los píxeles \mathbf{p}_{l_j,t_k} en el valor de intensidad $I_{l_n,t_{k+1}}$ de los respectivos píxeles $\mathbf{p}_{l_n,t_{k+1}}$. En realidad es un proceso complicado que implica el empleo de pasos adicionales y procedimientos complejos como los siguientes: el redondeo inteligente y dinámico de los píxeles transformados dados por $\mathbf{K}\mathbf{p}_{l_j,t_k}$, debido a que los píxeles, por su naturaleza, solo pueden tomar valores enteros positivos; la exclusión de aquellos píxeles transformados que quedan fuera de una imagen de tamaño $w \cdot h$, ya sea porque la componente horizontal, vertical o ambas quedan fuera; la extrapolación de valores de intensidad de los grupos de píxeles de la imagen después del movimiento que no se pudieron asociar con un píxel transformado; la interpolación de los valores de intensidad de píxeles solitarios de la imagen después del movimiento que no se pudieron asociar a un píxel transformado pero que están rodeados por píxeles que sí (Hartley y Zisserman, 2004; OpenCV, 2020b).

Es por ello que en este segundo paso, se utilizó una función de *opencv* conocida como *warpAffine* (OpenCV, 2020b,a), que calcula diversas transformaciones a imágenes 2D mediante un procedimiento bastante completo que emplea adecuadamente todos los pasos descritos en el párrafo anterior. En este caso, las entradas de dicha función son: la imagen de prueba antes del movimiento y la matriz de transformación euclidiana \mathbf{K} , dada por la expresión 4.2. Además, la salida de la función es la imagen después del movimiento, es decir, la imagen que corresponde a la transformación euclidiana de la imagen antes del movimiento. Específicamente, la salida son los valores de intensidad $I_{l_n,t_{k+1}}$ de cada uno de los píxeles $\mathbf{p}_{l_n,t_{k+1}}$ de la imagen después del movimiento. A manera de ejemplo, en la figura 4.10 se muestra una imagen después del movimiento que se obtuvo mediante la función *warpAffine*.

Cabe destacar que las variables de las cuales depende la matriz de transformación Euclidiana, dada por \mathbf{K} , que son el ángulo de rotación θ y la traslación $(t_c, t_r)^T$, se eligen de forma totalmente aleatoria dentro de los siguientes intervalos:

$$1 \leq \theta \leq l_\theta \quad (4.3)$$

$$1 \leq t_c \leq l_t \quad (4.4)$$

$$1 \leq t_r \leq l_t \quad (4.5)$$

donde l_{θ} es el límite del ángulo de rotación que puede tomar valores de 5, 10, 15 o 20 grados. Lo anterior se debe a que la prueba tiene un objetivo

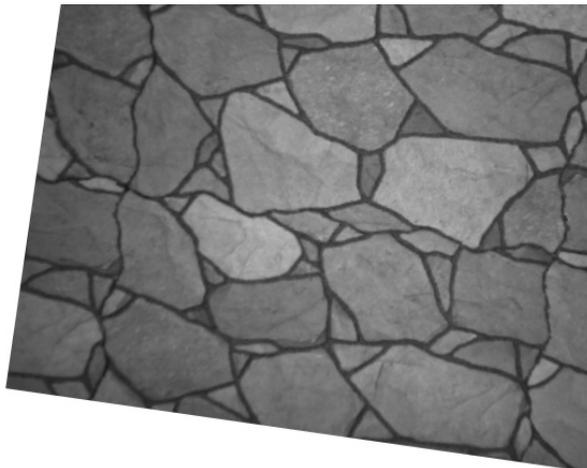


Figura 4.10: Imagen después del movimiento obtenida a partir de una transformación euclidiana de la imagen antes del movimiento, llevada a cabo por la función *warpaffine*.

secundario, el cual es determinar el valor máximo del límite de rotación con el cual el proceso de establecimiento de correspondencias funciona adecuadamente, es por ello que se pueden usar diferentes valores para dicho límite. Mientras que l_t es el límite de las componentes horizontales y verticales de la traslación, cuyo valor se eligió de tal forma que en el peor de los casos, cuando $(t_c, t_r)^T = (l_t, l_t)^T$, no se perdiera más de un 40% de la imagen antes del movimiento transformada, es decir, que el área de la imagen después del movimiento conformada por aquellos píxeles que no tienen un valor de intensidad asociado a algún píxel de la imagen original antes del movimiento no sea mayor a un 40%. Cabe destacar que la función *warpaffine* le asigna un valor de intensidad de 255 (color blanco) a los píxeles que conforman dicha área, como se aprecia en la figura 4.10. Entonces, lo anterior se cumple cuando l_t es igual a 120 píxeles. En la figura 4.11 se muestra todo el procedimiento para llegar a dicho valor.

En resumen, para este segundo paso, primero se elige de forma totalmente aleatoria el ángulo de rotación θ y las componentes de la traslación $(t_c, t_r)^T$, dentro del intervalo dado por las expresiones 4.3, 4.4 y 4.5. Luego, a partir de los valores elegidos aleatoriamente para dichos parámetros, se calcula la matriz de transformación euclidiana \mathbf{K} , mediante la expresión 4.2, Por último, se utiliza la función *warpaffine* para obtener la imagen después del movimiento.

En el tercer paso, se elige de forma totalmente aleatoria un píxel de la imagen de prueba antes del movimiento, lo que equivale a elegir aleatoriamente

un número a_p dentro del siguiente intervalo:

$$1 \leq a_p \leq w \cdot h \quad (4.6)$$

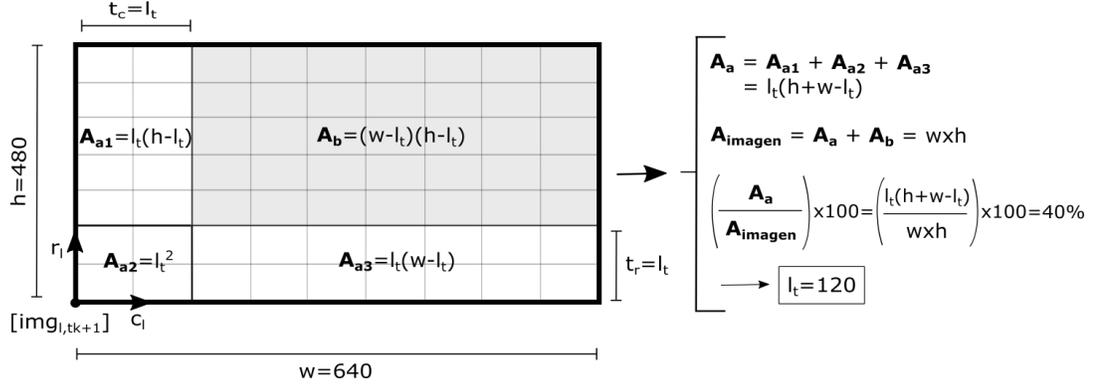


Figura 4.11: Obtención del límite de traslación para que en el peor caso posible no se pierda más de un 40 % de la imagen antes del movimiento transformada.

Entonces, el píxel aleatorio asociado se obtiene al evaluar $\mathbf{p}_{l_j,tk}$ en $j = a_p$:

$$\mathbf{p}_{l_j,tk}(c_{l_j}, r_{l_j}) = \mathbf{p}_{l_{a_p},tk}(c_{l_{a_p}}, r_{l_{a_p}}) \quad (4.7)$$

Seguidamente, se establece la correspondencia 2D entre el píxel $\mathbf{p}_{l_{a_p},tk}$ elegido aleatoriamente en la imagen de prueba antes del movimiento y un píxel $\mathbf{p}_{l_n,tk+1}$ de la imagen después del movimiento, elegido cuidadosamente, utilizando el procedimiento descrito en la sección 3.3 del capítulo 3, que fue propuesto e implementado por Moravec y posteriormente utilizado por Matthies, para encontrar la mejor coincidencia de un píxel de una imagen, en otra imagen diferente, pero similar. De esta forma se obtiene la correspondencia $(\mathbf{p}_{l_{a_p},tk}, \mathbf{p}_{l_{pc_{nmax}},tk+1})$, donde $\mathbf{p}_{l_{pc_{nmax}},tk+1}(c_{l_j}, r_{l_j})$ es la mejor coincidencia en la imagen después del movimiento del píxel $\mathbf{p}_{l_{a_p},tk}$ y que además, tiene un coeficiente de similitud máximo dado por $cpn_{pc_{nmax}}$.

Es importante destacar que para esta prueba, no se están considerando puntos característicos en la imagen antes del movimiento ni en la imagen después del movimiento y por lo tanto, para efectos del proceso de establecimiento de correspondencias, es como si todos los píxeles fueran puntos característicos, es decir, $\mathbf{p}_{l_j,tk} = \mathbf{h}_{i,tk}$ y $\mathbf{p}_{l_n,tk+1} = \mathbf{h}_{l_m,tk+1}$. Adicionalmente, se obviaron los subconjuntos de puntos característicos dados por $\mathbf{q}_{l_s,tk}$ y $\mathbf{q}_{l_s,tk+1}$. Asimismo, se obvió la condición de que el valor de $cpn_{pc_{nmax}}$ tiene que ser mayor a un valor dado por cpn_{min} , que puede elegirse entre 0.8 y 0.95. Aunque, dicha condición si se utilizara más adelante, pero solamente para clasificar la correspondencia. La razón de todo lo anterior, es que en este paso solamente se quiere establecer

una correspondencia entre dos píxeles, sin importar si es errónea o no o si son puntos característicos o no.

Por otro lado, se utilizó una ventana de correlación de tamaño 11×11 ($w_c \cdot h_c$), que se determinó por prueba y error y una ventana de búsqueda de tamaño 275×275 ($w_b \cdot h_b$), que se determinó a partir del valor del límite de traslación l_t , mediante la siguiente expresión:

$$w_b = h_b = (2 \cdot l_t) \cdot 1,15 = (2 \cdot 120) \cdot 1,15 \approx 275 \quad (4.8)$$

La razón por la que el tamaño de la ventana de búsqueda se calculó de la forma anterior, es para asegurarse de que en el peor de los casos, cuando $(t_c, t_r)^T = (l_t, l_t)^T$, la mejor coincidencia del píxel $\mathbf{p}_{la_p, tk}$ en la imagen después del movimiento, estuviera dentro de la ventana de búsqueda.

En el cuarto paso, se rota y se traslada el píxel $\mathbf{p}_{la_p, tk}$ de la imagen de prueba antes del movimiento, Para ello, se aplica una transformación euclidiana al píxel $\mathbf{p}_{la_p, tk}$, mediante el uso de la ecuación 4.1:

$$\mathbf{p}'_{la_p, tk+1} = \mathbf{K} \mathbf{p}_{la_p, tk} \quad (4.9)$$

donde $\mathbf{p}'_{la_p, tk+1}(c'_{la_p}, r'_{la_p})$ es el respectivo píxel transformado y \mathbf{K} es exactamente igual a la matriz de transformación euclidiana que se utilizó en la función *warpaffine*, para obtener la imagen después del movimiento, la cual depende de los parámetros de rotación y traslación elegidos aleatoriamente. Además, el píxel $\mathbf{p}'_{la_p, tk+1}$ es la verdadera coincidencia en la imagen después del movimiento del píxel $\mathbf{p}_{la_p, tk}$.

Seguidamente, se calcula la distancia, denotada como d_p , entre la verdadera coincidencia del píxel $\mathbf{p}_{la_p, tk}$ de la imagen antes del movimiento, que es el píxel $\mathbf{p}'_{la_p, tk+1}$ de la imagen después del movimiento y la coincidencia determinada en el tercer paso, que es el píxel $\mathbf{p}_{lpc_{nmax}, tk+1}$:

$$d_p = |\mathbf{p}_{lpc_{nmax}, tk+1} - \mathbf{p}'_{la_p, tk+1}|^2 \quad (4.10)$$

Idealmente, si la correspondencia fue establecida sin el más mínimo error, dicha distancia debería de ser exactamente igual a 0, es decir, que la coincidencia en la imagen después del movimiento del píxel $\mathbf{p}_{la_p, tk}$, determinada por el proceso de establecimiento de correspondencias empleado en el tercer paso, $\mathbf{p}_{lpc_{nmax}, tk+1}$, debería de tener las mismas coordenadas que la coincidencia real, $\mathbf{p}'_{la_p, tk}$.

En el quinto paso, se procede a clasificar la correspondencia dada por $(\mathbf{p}_{la_p, tk}, \mathbf{p}_{lpc_{nmax}, tk+1})$, de acuerdo al cumplimiento o incumplimiento de dos condiciones, que se denotan como c_1 y c_2 y que dependen del valor del coeficiente de similitud $cp_{lpc_{nmax}}$ y de la distancia d_p :

$$c_1 : cp_{lpc_{nmax}} \geq cp_{nmin} \quad (4.11)$$

$$c_2 : d_p \leq d_{p,max} \quad (4.12)$$

La primera condición, dada por la expresión 4.11 corresponde a la condición utilizada por el procedimiento de establecimiento de correspondencias temporales del algoritmo de odometría visual estereoscópico implementado, que es un primer paso para excluir correspondencias erróneas. Dicho de otra forma, es la condición que usa el proceso para determinar si una correspondencia es correcta o no, en una primera instancia. El valor de cpn_{min} se eligió empíricamente como 0.85.

Mientras que la segunda condición, dada por la expresión 4.12, es una condición para determinar si verdaderamente la correspondencia es correcta o no, es decir, proporciona la verdad absoluta. Dicho de otra forma, sirve para determinar si la clasificación de la correspondencia realizada por el algoritmo, mediante la condición c_1 (expresión 4.11), fue verdadera o falsa.

Además, el valor de $d_{p,max}$ se eligió de tal forma que fuera igual al valor del error máximo que se genera cuando la función *wapafine* redondea los píxeles transformados de la imagen antes del movimiento, dados por \mathbf{Kp}_{l_j,t_k} . Supóngase que el píxel transformado se representa como \mathbf{p}'_{l_j,t_k} y el píxel transformado redondeado como \mathbf{p}''_{l_j,t_k} . Entonces, en el peor de los casos, si las componentes del píxel transformado redondeado $\mathbf{p}''_{l_j,t_{k+1}}$ son (c''_{l_j}, r''_{l_j}) , las componentes del píxel transformado $\mathbf{p}'_{l_j,t_{k+1}}$ sin redondear eran $(c''_{l_j} \pm 0,5, r''_{l_j} \pm 0,5)$. Por lo tanto, el error máximo se puede obtener al calcular la distancia entre el píxel transformado sin redondear y el redondeado, en el peor de los casos descrito anteriormente, que da un valor de 0.7 aproximadamente. Por lo tanto, se usó un valor de 0.7 para $d_{p,max}$.

Entonces, cuando la condición c_1 y la condición c_2 se cumplen, la correspondencia es un verdadero positivo (VP), es decir, es una correspondencia que fue verdaderamente identificada por el proceso como correcta (Géron, 2017). Por cada correspondencia clasificada como VP se aumenta en 1 el valor de C_{VP} que es la cantidad de correspondencias que se han clasificado como VP:

$$\begin{aligned} & \text{si } c_1 \text{ y } c_2 \text{ se cumplen} \\ & \longrightarrow (\mathbf{p}_{l_p,t_k}, \mathbf{p}_{l_{pcn_{max}},t_{k+1}}) = VP \\ & \longrightarrow C_{VP} = C_{VP} + 1 \end{aligned} \quad (4.13)$$

Si sucede lo contrario a lo anterior, que la condición c_1 y la condición c_2 no se cumplen, entonces la correspondencia es un verdadero negativo (VN), es decir, es una correspondencia que fue verdaderamente identificada por el proceso como incorrecta (Géron, 2017). Por cada correspondencia clasificada como VN se aumenta en 1 el valor de C_{VN} que es la cantidad de correspondencias que se han clasificado como VN:

$$\begin{aligned}
& \text{si } c_1 \text{ y } c_2 \text{ no se cumplen} \\
& \longrightarrow (\mathbf{p}_{la_p,tk}, \mathbf{p}_{lpc_{n_{max}},t_{k+1}}) = VN \\
& \longrightarrow C_{VN} = C_{VN} + 1
\end{aligned} \tag{4.14}$$

Por otro lado, cuando la condición c_1 se cumple y la condición c_2 no se cumple, la correspondencia es un falso positivo (FP), es decir, es una correspondencia que fue falsamente identificada por el proceso como correcta (Géron, 2017). Por cada correspondencia clasificada como FP se aumenta en 1 el valor de C_{FP} que es la cantidad de correspondencias que se han clasificado como FP:

$$\begin{aligned}
& \text{si } c_1 \text{ se cumple y } c_2 \text{ no se cumple} \\
& \longrightarrow (\mathbf{p}_{la_p,tk}, \mathbf{p}_{lpc_{n_{max}},t_{k+1}}) = FP \\
& \longrightarrow C_{FP} = C_{FP} + 1
\end{aligned} \tag{4.15}$$

Si sucede lo contrario a lo anterior, que la condición c_1 no se cumple y la condición c_2 se cumple, entonces la correspondencia es un falso negativo (FN), es decir, es una correspondencia que fue falsamente identificada por el proceso como incorrecta (Géron, 2017). Por cada correspondencia clasificada como FN se aumenta en 1 el valor de C_{FN} que es la cantidad de correspondencias que se han clasificado como FN:

$$\begin{aligned}
& \text{si } c_1 \text{ no se cumple y } c_2 \text{ se cumple} \\
& \longrightarrow (\mathbf{p}_{la_p,tk}, \mathbf{p}_{lpc_{n_{max}},t_{k+1}}) = FN \\
& \longrightarrow C_{FN} = C_{FN} + 1
\end{aligned} \tag{4.16}$$

La clasificación de una correspondencia $(\mathbf{p}_{la_p,tk}, \mathbf{p}_{lpc_{n_{max}},t_{k+1}})$, de acuerdo al cumplimiento o incumplimiento de las condiciones c_1 (expresión 4.11) y c_2 (expresión 4.12), se puede resumir con una matriz, como se aprecia en la figura 4.12. Dicha matriz se conoce como la matriz de confusión (Géron, 2017).

En el sexto paso se repite el tercer, cuarto y quinto paso un total de N_c veces, es decir, se establecen y se clasifican un total de N_c correspondencias entre píxeles de la imagen de prueba antes del movimiento, elegidos aleatoriamente y píxeles de la imagen después del movimiento, elegidos cuidadosamente. Además, el valor de N_c se estableció como 200.

En el séptimo paso, se iteran todos los pasos anteriores una cantidad de N_a veces, a excepción del primer paso que solo se debe de ejecutar una vez, leer y guardar la imagen de prueba antes del movimiento. Entonces, de forma resumida, los pasos que se iteran N_a veces son los siguientes: (2) se elige de forma aleatoria nuevos valores para el ángulo de rotación θ y para las componentes de la traslación (t_c, t_r) dentro de los respectivos intervalos y a partir de estos se calcula una nueva matriz de transformación euclidiana \mathbf{K} , luego, se

		Verdad Absoluta (condición c_2)	
		Positivo	Negativo
		c_2 se cumple ($dp \leq dp_{max}$)	c_2 no se cumple ($dp \geq dp_{max}$)
Algoritmo (condición c_1)	Positivo c_1 se cumple ($cpn_{pcnmax} \geq cpn_{min}$)	VP	FP
	Negativo c_1 no se cumple ($cpn_{pcnmax} \leq cpn_{min}$)	FN	VN

Figura 4.12: Matriz de confusión para la clasificación de las correspondencias.

genera una nueva imagen después del movimiento mediante el uso de la función *warpaffine*, que es la transformación euclidiana de la imagen antes del movimiento; (3) se elige de forma aleatoria un píxel $\mathbf{p}_{la_p,tk}$ en la imagen antes del movimiento y se establece la correspondencia 2D entre dicho píxel y un píxel $\mathbf{p}_{lpcn_{max},tk+1}$ elegido cuidadosamente en la imagen después del movimiento, que es la mejor coincidencia; (4) se calcula la distancia entre la verdadera coincidencia $\mathbf{p}'_{la_p,tk}$ del píxel aleatorio y la coincidencia establecida $\mathbf{p}_{lpcn_{max},tk+1}$; (5) se clasifica la correspondencia establecida ($\mathbf{p}_{la_p,tk}, \mathbf{p}_{lpcn_{max},tk+1}$) como un verdadero o falso positivo o negativo y se actualiza la cantidad correspondiente; (6) se repite (3), (4) y (5) un total de N_c veces. Cabe aclarar que el valor de N_a se estableció como 1000.

En el octavo y último paso, justo después de que haya finalizado la última iteración, se calculan tres métricas que permiten cuantificar que tan preciso y efectivo es el proceso de establecimiento de correspondencias, en función de la cantidad total de correspondencias clasificadas como verdaderos positivos (C_{VP}), verdaderos negativos (C_{VN}), falsos positivos (C_{FP}) y falsos negativos (C_{FN}).

La primera métrica se conoce como precisión e intenta responder la siguiente pregunta: ¿Qué porcentaje de las correspondencias que fueron identificadas por el proceso como correctas ($C_{VP} + C_{FP}$) verdaderamente lo son? También se podría decir que mide que tan propenso es el proceso a cometer falsos positivos. Matemáticamente, la precisión se denota como P_m y se calcula de la siguiente forma (Géron, 2017):

$$P_m = \frac{C_{VP}}{C_{VP} + C_{FP}} \cdot 100\% \quad (4.17)$$

Entonces, cuando el proceso identifica una correspondencia como correcta,

está en lo cierto un P_m % del tiempo.

La segunda métrica se conoce como sensibilidad e intenta responder a la siguiente pregunta: ¿Qué porcentaje de las correspondencias verdaderamente correctas ($C_{VP} + C_{FN}$) el proceso logro identificar como tal? También se podría decir que mide que tan propenso es el proceso a cometer falsos negativos. Matemáticamente, la sensibilidad se denota como S_m y se calcula de la siguiente forma (Géron, 2017):

$$S_m = \frac{C_{VP}}{C_{VP} + C_{FN}} \cdot 100\% \quad (4.18)$$

Entonces, el proceso identifica como correctas un S_m % de todas las correspondencias verdaderamente lo son.

La tercer y ultima métrica se conoce como el valor F_1 que se utiliza para combinar las medidas de precisión y sensibilidad en un solo valor. Esto es práctico porque hace más fácil el poder comparar el rendimiento combinado de la precisión y la sensibilidad entre varios casos de una misma prueba. El valor F1 se calcula haciendo la media armónica entre P_m y S_m (Géron, 2017):

$$F_1 = \left[2 \cdot \frac{P_m \cdot S_m}{P_m + S_m} \right] \cdot 100\% \quad (4.19)$$

Finalmente, la salida o el resultado final de la prueba, son los valores de las tres métricas definidas anteriormente: la precisión P_m , la sensibilidad S_m y el valor F_1 .

Por otro lado, para la prueba descrita anteriormente, que está compuesta por ocho pasos, se plantearon un total de 19 casos de acuerdo a todas las posibles combinaciones entre los tres parámetros de movimiento 2D (θ , t_c y t_r) y los distintos valores que puede tomar el límite del ángulo de rotación ($l_\theta = 5,10,15,20$):

- | | |
|--|--|
| Caso 1: t_c . | Caso 9: t_c, θ ($l_\theta = 10^\circ$). |
| Caso 2: t_r . | Caso 10: t_r, θ ($l_\theta = 10^\circ$). |
| Caso 3: t_c, t_r . | Caso 11: t_c, t_r, θ ($l_\theta = 10^\circ$). |
| Caso 4: θ ($l_\theta = 5^\circ$). | Caso 12: θ ($l_\theta = 15^\circ$). |
| Caso 5: t_c, θ ($l_\theta = 5^\circ$). | Caso 13: t_c, θ ($l_\theta = 15^\circ$). |
| Caso 6: t_r, θ ($l_\theta = 5^\circ$). | Caso 14: t_r, θ ($l_\theta = 15^\circ$). |
| Caso 7: t_c, t_r, θ ($l_\theta = 5^\circ$). | Caso 15: t_c, t_r, θ ($l_\theta = 15^\circ$). |
| Caso 8: θ ($l_\theta = 10^\circ$). | Caso 16: θ ($l_\theta = 20^\circ$). |

Caso 17: t_c, θ ($l_\theta = 20^\circ$).

Caso 19: t_c, t_r, θ ($l_\theta = 20^\circ$).

Caso 18: t_r, θ ($l_\theta = 20^\circ$).

En la lista anterior, si para un caso dado un parámetro de movimiento 2D (θ, t_c y t_r) no aparece del todo, quiere decir que su límite se está tomando como 0, es decir, que el valor de dicho parámetro se mantiene como 0, no se toma en cuenta. Además, el límite para las componentes de traslación que si aparecen es 120. Por ejemplo, para el caso número 2, el límite de la componente vertical de la traslación (t_r) es 120, mientras que el límite de la componente vertical de la traslación y del ángulo de rotación es 0, debido a que no aparecen.

Entonces, la prueba se llevó a cabo una vez para cada caso de la lista, de tal forma que se obtuvieron los respectivos valores de la precisión, sensibilidad y el valor F1 para cada uno. Es importante destacar que como se mencionó anteriormente, la prueba tiene un objetivo adicional, que es determinar el valor máximo del límite de rotación con el cual el proceso de establecimiento de correspondencias funciona adecuadamente, es por ello que para el planteamiento de los casos se usaron diferentes valores de dicho límite.

Por último, los parámetros de control de la prueba, junto con sus respectivos valores, los cuales se mencionaron, se definieron y se eligieron o calcularon a lo largo de toda esta sección, se resumen a continuación:

Imagen y ventanas:

Ancho_de_la_imagen(w): 640
 Largo_de_la_imagen(h): 480
 Ancho_de_la_ventana_de_correlacion(wc): 11
 Largo_de_la_ventana_de_correlacion(hc): 11
 Ancho_de_la_ventana_de_busqueda(wb): 275
 Largo_de_la_ventana_de_busqueda(hb): 275

Límites:

Límite_de_la_traslación_horizontal(tc): 0 o 120
 Límite_de_la_traslación_vertical(tv): 0 o 120
 Límite_del_ángulo_de_rotación(theta): 0,5,10,15 o 20

Condiciones:

Coefficiente_de_similitud_minimo(cpn_min): 0.9
 Distancia_maxima(dp_max): 0.7

Iteraciones:

Cantidad_de_correspondencias_por_iteración(Nc): 200
 Numero_de_iteraciones(Na): 1000

Dichos parámetros y sus respectivos valores, se establecieron de forma externa a la prueba, mediante un archivo de texto que tiene exactamente el mismo formato de la lista anterior. Ese archivo se lee en el primer paso, es decir, al puro inicio de la ejecución de la prueba, justo antes de leer la imagen de entrada de prueba. Cada destacar que para cada caso, los únicos valores que se cambiaron fueron los valores de los límites de traslación y rotación, los valores de los demás parámetros se mantuvieron igual.

4.2.2 Prueba II: estimación de movimiento

La finalidad de esta segunda prueba es probar el correcto funcionamiento y determinar la precisión y exactitud del proceso del algoritmo de odometría visual estereoscópico implementado, encargado de encontrar los parámetros de movimiento tridimensionales, la traslación y rotación tridimensional, con los que se tiene que mover una nube de puntos obtenida en un instante antes del movimiento, para que esta coincida con la correspondiente nube de puntos obtenida en un instante después del movimiento, mediante un procedimiento basado en la descomposición en valores singulares de una matriz. Esta prueba consiste de ocho pasos, los cuales se explican a lo largo de toda esta sección.

Primero, se lee el archivo de texto llamado *Nube1.txt* que contiene la nube de puntos de prueba antes del movimiento. Seguidamente, se guardan las posiciones tridimensionales, dadas en centímetros, de los puntos que conforman la nube de puntos de prueba, dados por $\mathbf{H}_{i,t_k}(X_{li}, Y_{li}, Z_{li})$, excluyendo aquellos puntos del archivo de texto que tienen una posición con un valor tipo NaN, es decir, que no son parte de la nube. Además, el tamaño de la nube, dado por $C_{N_{t_k}}$, es conocido, ya que es igual a la cantidad de puntos cuyas posiciones se guardaron.

En el segundo paso, se genera la nube de puntos después del movimiento, de forma sintética. Para ello, cada uno de los puntos \mathbf{H}_{i,t_k} que conforman la nube de prueba antes del movimiento, se rotan y se trasladan mediante el uso de la ecuación de pose, dada por la expresión 2.22 de la sección 2.1.3 del capítulo 2, para así obtener la nube después del movimiento que estaría conformada por los puntos \mathbf{H}_{i,t_k} rotados y trasladados, que se denotan como $\mathbf{H}_{m,t_{k+1}}$.

Entonces, para obtener los puntos $\mathbf{H}_{m,t_{k+1}}$ que forman la nube de puntos antes del movimiento, simplemente se emplea la ecuación de pose, sustituyendo \mathbf{H} , \mathbf{H}^* , \mathbf{R} y \mathbf{T} , por $\mathbf{H}_{m,t_{k+1}}$, \mathbf{H}_{i,t_k} , $\Delta\mathbf{R}$ y $\Delta\mathbf{T}$, respectivamente:

$$\mathbf{H}_{m=i,t_{k+1}} = \Delta\mathbf{R} \cdot \mathbf{H}_{i,t_k} + \Delta\mathbf{T} \quad (4.20)$$

donde $\Delta\mathbf{T}$ es el vector de traslación dado por $(\Delta T_{X_l}, \Delta T_{Y_l}, \Delta T_{Z_l})^T$ y $\Delta\mathbf{R}$ es la matriz de rotación, cuyas entradas dependen de los ángulos de rotación dados por $(\Delta W_{X_l}, \Delta W_{Y_l}, \Delta W_{Z_l})^T$, que forman el vector de rotación $\Delta\mathbf{W}$.

Además, tanto las componentes del vector de traslación $\Delta\mathbf{T}$, como las componentes del vector de rotación $\Delta\mathbf{W}$ son conocidas y se eligen de forma totalmente aleatoria dentro de los siguientes intervalos:

$$1 \leq \Delta W_{X_l}, \Delta W_{Y_l}, \Delta W_{Z_l} \leq l_W \quad (4.21)$$

$$1 \leq \Delta T_{X_l}, \Delta T_{Y_l}, \Delta T_{Z_l} \leq l_T \quad (4.22)$$

donde l_W es el límite de las tres componentes del vector de rotación alrededor de los ejes X_l , Y_l y Z_l , cuyo valor es igual a 10 grados y se obtuvo a partir de los resultados de la prueba anterior, los cuales se muestran y se analizan detalladamente en el siguiente capítulo. Por ahora solo interesa saber que el valor del límite l_W es 10 grados. Mientras que l_T es el límite de las tres componentes del vector de traslación a lo largo de los ejes X_l , Y_l y Z_l , cuyo valor es igual a 18 centímetros y se obtuvo a partir de límite de traslación de 120 píxeles utilizado en la prueba anterior. Para ello, se programó un pequeño algoritmo que calculara el valor del límite de la traslación tridimensional (l_T), de tal forma que fuera equivalente al límite de la traslación bidimensional (l_t), usado en la prueba anterior y que es igual a 120 píxeles. Entonces, lo que se hizo fue buscar en la imagen de prueba antes del movimiento todos los posibles pares de píxeles que estuvieran a una distancia de 120 píxeles y que además, tuvieran asociado un punto tridimensional dentro de la respectiva nube de puntos de prueba. Luego, por cada par de píxeles se calculó la distancia entre los respectivos puntos tridimensionales. Por último, se calcula la media de dichas distancias y se obtuvo como resultado el valor de 18 centímetros. En la figura 4.13 se muestra el procedimiento resumido para llegar a dicho valor.

Entonces, las componentes del vector de rotación $\Delta\mathbf{W}$ y del vector de traslación $\Delta\mathbf{T}$ se calculan de forma aleatoria dentro de los intervalos dados por las expresiones 4.21 y 4.22, respectivamente. Mientras que la matriz de rotación $\Delta\mathbf{R}$ se obtiene a partir de las componentes de $\Delta\mathbf{W}$ (ángulos de rotación), usando las ecuaciones de la 2.8 a la 2.16, de la sección 2.1.4 del capítulo 2. Luego, se obtiene la nube de puntos después del movimiento al rotar y trasladar la nube de puntos de prueba antes del movimiento, con el vector $\Delta\mathbf{T}$ y la matriz $\Delta\mathbf{R}$, empleando la ecuación 4.20.

Cabe destacar que el tamaño de la nube de puntos después del movimiento, dado por $C_{N_{t_{k+1}}}$, es exactamente igual a $C_{N_{t_k}}$, que es el tamaño de la nube antes del movimiento. Así mismo, las correspondencias temporales 3D entre las dos nubes de puntos son conocidas y están dadas por $(\mathbf{H}_{i,t_k}, \mathbf{H}_{m=i,t_{k+1}})$ para todo valor de i que va desde 1 hasta $C_{N_{t_k}}$. Lo anterior se debe a que la nube de puntos después del movimiento se obtiene a partir de la nube de prueba antes del movimiento, con la expresión 4.20, lo que permite establecer las correspondencias de forma una forma demasiado sencilla, el primer punto

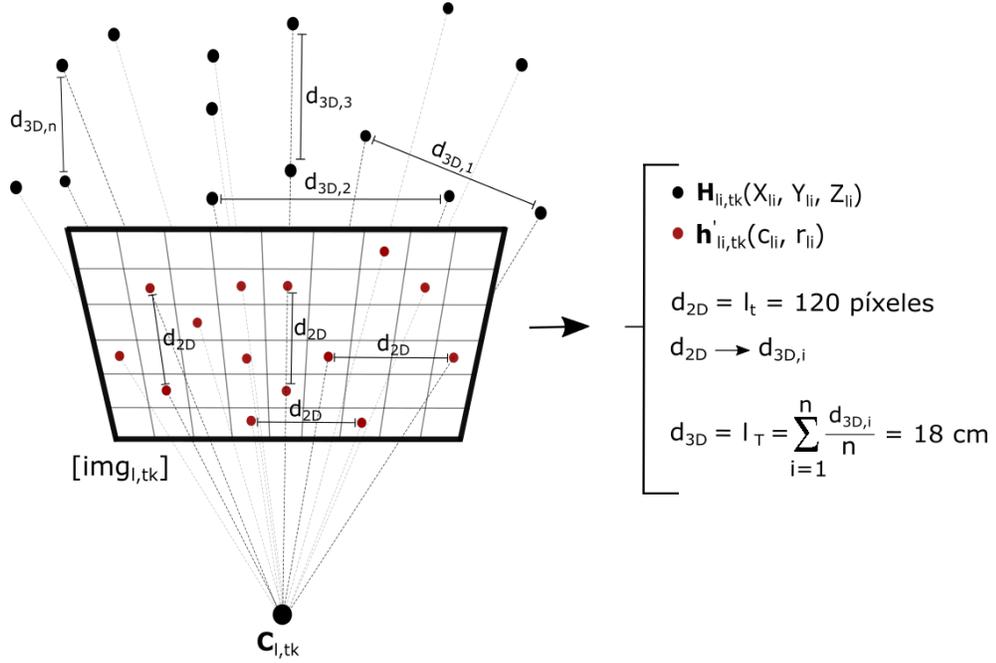


Figura 4.13: Obtención del límite de traslación tridimensional de la nube de puntos de prueba a partir del límite de traslación bidimensional de la imagen izquierda prueba.

de la primera nube corresponde al primer punto de la segunda nube y así sucesivamente.

Seguidamente, en el tercer paso, se introduce un error de posición aleatorio gaussiano, de media cero y varianza conocida, a la nube de puntos después del movimiento, es decir, a las posiciones tridimensionales de cada uno de los puntos $\mathbf{H}_{m,t_{k+1}}$ que la conforman. El fin de esto es simular de forma controlada el ruido que perturba a las posiciones de las nubes de puntos debido a una cierta cantidad de ruido en las respectivas imágenes y que se da cuando el algoritmo de odometría visual estereoscópico se corre o se prueba en tiempo real.

Matemáticamente, el error es introducido a la nube de puntos después del movimiento mediante una simple adición vectorial, tal y como se muestra en la siguiente ecuación:

$$\mathbf{H}_{m,t_{k+1}} = \mathbf{H}_{m,t_k} + \mathbf{e}_{pm} \quad (4.23)$$

donde \mathbf{e}_{pm} son los vectores del error posición tridimensional que se adicionan a cada uno de los respectivos puntos $\mathbf{H}_{m,t_{k+1}}$ que conforman la nube de puntos después del movimiento, cuyas componentes se representan como

$(e_{X_{lm}}, e_{Y_{lm}}, e_{Z_{lm}})^T$, que son el error de posición escalar en X_l , en Y_l y en Z_l , respectivamente.

Cada uno de los vectores \mathbf{e}_{pm} y sus componentes, se calculan mediante la siguiente ecuación:

$$\mathbf{e}_{pm} = \begin{pmatrix} e_{X_{lm}} \\ e_{Y_{lm}} \\ e_{Z_{lm}} \end{pmatrix} = \sqrt{\sigma_p^2} \begin{pmatrix} a_{X_{lm}} \\ a_{Y_{lm}} \\ a_{Z_{lm}} \end{pmatrix} = \sigma_p \begin{pmatrix} a_{X_{lm}} \\ a_{Y_{lm}} \\ a_{Z_{lm}} \end{pmatrix} \quad (4.24)$$

donde σ_p^2 es la varianza del error de posición general y σ_p es la respectiva desviación estándar. La varianza puede tomar valores de 0 a 100 centímetros cuadrados y por lo tanto, la desviación estándar puede tomar valores de 0 a 10 centímetros, ya que esta es la raíz cuadrada de la varianza. Es importante destacar que el valor de la varianza y de la desviación estándar, se mantiene constante para el cálculo de todos los vectores de error de posición \mathbf{e}_{pm} , es decir, el valor de la varianza y de la desviación estándar no se varia cuando se lleva a cabo el proceso de introducir el error de posición a la nube de puntos después del movimiento. Por otro lado, las variables dadas por $a_{X_{lm}}$, $a_{Y_{lm}}$ y $a_{Z_{lm}}$ son números aleatorios que se obtienen a partir de una distribución gaussiana con media cero y varianza unitaria. Para ello, se utilizó una rutina conocida como *gasdev*, que se extrajo del libro llamado *Numerical Recipes* (Press et al., 2007), la cual genera un número aleatorio gaussiano cada vez que se llama, de tal forma que la expresión 4.24 se puede expresar en términos esta, de la siguiente forma:

$$\mathbf{e}_{pm} = \begin{pmatrix} e_{X_{lm}} \\ e_{Y_{lm}} \\ e_{Z_{lm}} \end{pmatrix} = \sigma_p \begin{pmatrix} a_{X_{lm}} \\ a_{Y_{lm}} \\ a_{Z_{lm}} \end{pmatrix} = \sigma_p \begin{pmatrix} \text{gasdev}() \\ \text{gasdev}() \\ \text{gasdev}() \end{pmatrix} \quad (4.25)$$

Entonces, se va recorriendo la nube de puntos después del movimiento y por cada punto $\mathbf{H}_{m,t_{k+1}}$ se calcula el respectivo vector del error de posición \mathbf{e}_m mediante la expresión 4.25 que genera un vector de error aleatorio gaussiano y se adiciona a las posiciones tridimensionales del punto $\mathbf{H}_{m,t_{k+1}}$ empleando la expresión 4.23. Es de esta forma como se introduce un error de posición a la nube de puntos después del movimiento, de tal forma que ahora cada uno de los puntos $\mathbf{H}_{m,t_{k+1}}$ tienen un error adicionado.

En un cuarto paso, se estiman los parámetros de movimiento, encontrando aquella traslación y rotación con la que se tiene que mover la nube de puntos de prueba antes del movimiento para que esta coincida con la nube de puntos después del movimiento, la cual tiene un error de posición que fue introducido de forma sintética y controlada. Para ello, se utiliza el procedimiento descrito en la sección 3.4.1 del capítulo 3, que se basa en la descomposición en valores singulares de una matriz 3x3 para encontrar la mejor solución del problema

de minimización de un error cuadrático medio que depende de las correspondencias temporales 3D entre las nubes de puntos y que además, es una versión ligeramente mejorada del proceso que fue utilizado por Matthies para estimar el movimiento de un rover entre dos instantes de tiempos consecutivos. De esta forma se obtiene el vector de traslación $\Delta \mathbf{T}'$ y la matriz de rotación $\Delta \mathbf{R}'$, que son la solución del problema de minimización y que garantizan la mejor coincidencia entre la nube de puntos antes del movimiento rotada y trasladada con dichos parámetros y la nube de puntos después del movimiento con error. Además, el vector de rotación $\Delta \mathbf{W}'$ se obtiene a partir de las entradas de la matriz de rotación $\Delta \mathbf{R}'$, mediante el método del seno inverso, usando las ecuaciones de la 2.18 a la 2.19, de la sección 2.1.4 del capítulo 2.

Es importante destacar que para esta prueba en particular y para efectos del procedimiento de estimación de movimiento por descomposición en valores singulares de una matriz, se consideraron las nubes de puntos completas en lugar de subconjuntos dados por $\mathbf{Q}_{ls,tk}$ y $\mathbf{Q}_{ls,tk+1}$ de las respectivas nubes, que es equivalente a considerar que dichos subconjuntos son exactamente iguales a las nubes de puntos, es decir, que $\mathbf{Q}_{ls,tk} = \mathbf{H}_{i,tk}$ y $\mathbf{Q}_{ls,tk+1} = \mathbf{H}_{m,tk+1}$, de tal forma que $s = i = m$ y $C_q = C_{N_{t_k}} = C_{N_{t_{k+1}}}$, que es el tamaño de los subconjuntos.

Posteriormente, en el quinto paso, se procede a calcular el error absoluto de rotación, así como el error absoluto de traslación, mediante una simple resta vectorial entre los vectores de rotación y traslación reales ($\Delta \mathbf{W}$ y $\Delta \mathbf{T}$) y los vectores de rotación y traslación estimados en el paso anterior ($\Delta \mathbf{W}'$ y $\Delta \mathbf{T}'$), respectivamente, tal y como se muestra en las siguientes ecuaciones:

$$\mathbf{e}_{W,u} = \text{abs}(\Delta \mathbf{W}' - \Delta \mathbf{W}) \quad (4.26)$$

$$\mathbf{e}_{T,u} = \text{abs}(\Delta \mathbf{T}' - \Delta \mathbf{T}) \quad (4.27)$$

donde $\mathbf{e}_{W,u}$ es un vector del error absoluto de rotación, cuyas componentes se representan como $(e_{W_{X_l,u}}, e_{W_{Y_l,u}}, e_{W_{Z_l,u}})^T$ que son los errores de rotación escalar alrededor de X_l , Y_l y Z_l , respectivamente. Análogamente, $\mathbf{e}_{T,u}$ es el vector del error absoluto de traslación, cuyas componentes se representan como $(e_{T_{X_l,u}}, e_{T_{Y_l,u}}, e_{T_{Z_l,u}})^T$ que son los errores de traslación escalar a lo largo de X_l , Y_l y Z_l , respectivamente. La variable u se usa para representar el u -ésimo error de traslación y rotación, cuyo valor va desde 1 hasta N_e , que es la cantidad de veces que se van a iterar este paso junto con los pasos anteriores, lo cual se explicara mejor en el siguiente paso. Además, $\text{abs}()$ es una función que representa el proceso de obtener el valor absoluto de las componentes de un vector.

Luego de que se calcula el u -ésimo error absoluto de rotación y traslación, mediante las expresiones 4.26 y 4.27, respectivamente, se procede a calcular un

tercer error que corresponde al error de coincidencia entre las nubes de puntos, el cual es la media de la distancia euclidiana entre los puntos \mathbf{H}_{i,t_k} de la nube de puntos antes del movimiento, rotados y trasladados con los parámetros de movimiento estimados ($\Delta\mathbf{R}'$ y $\Delta\mathbf{T}'$) y los puntos correspondientes $\mathbf{H}_{m=i,t_{k+1}}$ de la nube de puntos después del movimiento. Matemáticamente, el u -ésimo error de coincidencia se denota como $e_{c,u}$ y se calcula con la siguiente ecuación:

$$e_{c,u} = \frac{1}{C_{N_{t_k}}} \sum_{i=1}^{C_{N_{t_k}}} |\mathbf{H}_{m=i,t_{k+1}} - (\Delta\mathbf{R}' \cdot \mathbf{H}_{i,t_k} + \Delta\mathbf{T}')|^2 \quad (4.28)$$

En el sexto paso, se iteran todos los pasos anteriores una cantidad de N_e veces, a excepción del primer paso, el cual se debe ejecutar una sola vez, que es leer y guardar la nube de puntos de prueba antes del movimiento. De forma resumida, los pasos que se iteran un total de N_e veces son los siguientes: (2) se elige de forma aleatoria nuevos valores para las componentes del vector de rotación $\Delta\mathbf{W}$ y el vector de traslación $\Delta\mathbf{T}$, mientras que la matriz de rotación $\Delta\mathbf{R}$ se obtiene a partir de $\Delta\mathbf{W}$, luego a partir de $\Delta\mathbf{T}$ y $\Delta\mathbf{R}$ se genera una nueva nube de puntos después del movimiento, al aplicarle la ecuación de pose a la nube antes del movimiento; (3) se adiciona un error de posición aleatorio gaussiano, de media cero y varianza σ_p^2 , a la nube de puntos después del movimiento; (4) se estiman los parámetros de movimiento, el vector de traslación $\Delta\mathbf{T}'$ y la matriz de rotación $\Delta\mathbf{R}'$, mientras que el vector de rotación $\Delta\mathbf{W}'$ se obtiene a partir de $\Delta\mathbf{R}'$; (5) se calculan los respectivos errores absolutos de traslación y rotación, $\mathbf{e}_{W,u}$ y $\mathbf{e}_{T,u}$, así como el respectivo error de coincidencia $e_{c,u}$ entre las nubes de puntos. Cabe destacar que por cada iteración se guardan los valores de los tres errores anteriores, de tal forma que luego de la última iteración se conocen los valores de $\mathbf{e}_{T,u}$, $\mathbf{e}_{W,u}$ y $e_{c,u}$ asociados a cada iteración. Además, el valor de N_e se estableció como 1000.

En el séptimo paso, luego de que haya finalizado la última iteración, se calcula por cada tipo de error, de rotación, de traslación y de coincidencia, dos métricas que cuantifican que tan preciso y efectivo es el proceso de estimación de movimiento por descomposición en valores singulares, a partir de los diferentes valores de cada error, $\mathbf{e}_{W,u}$, $\mathbf{e}_{T,u}$ y $e_{c,u}$, que se obtuvieron para cada iteración y que forman conjuntos de valores de tamaño N_e , que es la cantidad total de iteraciones.

La primera métrica es la media del error que se puede definir como el valor central o valor promedio de un conjunto formado por diferentes valores de dicho error y se calcula para el error de rotación, traslación y de coincidencia (Borovkov, 2009). En este caso las unidades de la media son centímetros. Matemáticamente, la media del error absoluto de rotación se denota como μ_{e_W} , la del error absoluto de traslación como μ_{e_T} y la del error de coincidencia entre nubes de puntos como μ_{e_c} y se calculan de la siguiente manera:

$$\boldsymbol{\mu}_{e_W} = \frac{1}{N_e} \sum_{u=1}^{N_e} \mathbf{e}_{W,u} \quad (4.29)$$

$$\boldsymbol{\mu}_{e_T} = \frac{1}{N_e} \sum_{u=1}^{N_e} \mathbf{e}_{T,u} \quad (4.30)$$

$$\mu_{e_c} = \frac{1}{N_e} \sum_{u=1}^{N_e} e_{c,u} \quad (4.31)$$

Es importante notar que la media del error absoluto de rotación es un vector cuyas componentes se representan como $(\mu_{e_{W_{X_l}}}, \mu_{e_{W_{Y_l}}}, \mu_{e_{W_{Z_l}}})^T$ y son las medias del error de rotación escalar alrededor de los tres ejes principales de la cámara izquierda, dados por (X_l, Y_l, Z_l) . Análogamente, la media del error absoluto de traslación es un vector dado por $(\mu_{e_{T_{X_l}}}, \mu_{e_{T_{Y_l}}}, \mu_{e_{T_{Z_l}}})^T$, que son las medias del error de traslación escalar a lo largo de los ejes X_l, Y_l y Z_l .

La segunda métrica es la varianza del error que se puede definir una medida de dispersión que representan la variabilidad de los diferentes valores de dicho error respecto a su media y se calcula para el error de rotación, traslación y de coincidencia. En este caso las unidades de la varianza son centímetros cuadrados (Borovkov, 2009). Matemáticamente, la varianza del error absoluto de rotación se denota como $\boldsymbol{\sigma}_{e_W}^2$, la del error absoluto de traslación como $\boldsymbol{\sigma}_{e_T}^2$ y la del error de coincidencia entre nubes de puntos como $\sigma_{e_c}^2$ y se calculan de la siguiente manera:

$$\boldsymbol{\sigma}_{e_W}^2 = \frac{1}{N_e} \left(\sum_{u=1}^{N_e} \mathbf{e}_{W,u}^2 \right) - \boldsymbol{\mu}_{e_W}^2 \quad (4.32)$$

$$\boldsymbol{\sigma}_{e_T}^2 = \frac{1}{N_e} \left(\sum_{u=1}^{N_e} \mathbf{e}_{T,u}^2 \right) - \boldsymbol{\mu}_{e_T}^2 \quad (4.33)$$

$$\sigma_{e_c}^2 = \frac{1}{N_e} \left(\sum_{u=1}^{N_e} e_{c,u}^2 \right) - \mu_{e_c}^2 \quad (4.34)$$

Cabe destacar que la varianza del error absoluto de rotación es un vector al igual que su media y sus componentes son $(\sigma_{e_{W_{X_l}}}^2, \sigma_{e_{W_{Y_l}}}^2, \sigma_{e_{W_{Z_l}}}^2)^T$, que representa la varianza de dicho error alrededor de los ejes X_l, Y_l y Z_l . Análogamente, la varianza del error absoluto de traslación también es vector dado por $(\sigma_{e_{T_{X_l}}}^2, \sigma_{e_{T_{Y_l}}}^2, \sigma_{e_{T_{Z_l}}}^2)^T$, que son las varianzas de dicho error a lo largo de X_l, Y_l y Z_l .

Entonces, con las ecuaciones de la 4.29 a la 4.32 se calcula la media y la varianza del error absoluto de rotación y traslación en X_l , Y_l y Z_l , así como la media y la varianza del error de coincidencia. Además, cada una de las variables anteriores se guardan en un archivo que tiene el nombre de la respectiva variable, cuyo formato general se detallara más adelante. Por ahora solo es importante destacar que las medias y varianzas se guardan en archivos externos y que son un total de 14 archivos, 6 para las medias y varianzas del error de rotación en los tres ejes, otros 6 para las medias y varianzas del error de traslación en los tres ejes y otros 2 para la media y varianza del error de coincidencia.

En el octavo y último paso, se repite el proceso descrito desde el segundo paso hasta el séptimo paso, es decir, desde que se genera la segunda nube de puntos después del movimiento hasta que se calculan y se almacenan las medias y varianzas de los errores. Para cada repetición se varia el valor de la varianza σ_p^2 del error de posición, de tal forma que el valor de esta se empieza en 0 y se va aumentando en un valor igual a V_{σ^2} cada vez que se da una nueva repetición y así sucesivamente hasta que se llega a un valor máximo de dicha varianza, el cual es conocido y está dado por σ_{max}^2 . Lo anterior se puede resumir de la siguiente manera:

$$\begin{aligned} & \sigma_p^2 = 0 \\ \text{Mientras que } & (\sigma_p^2 \leq \sigma_{max}^2) : \\ & \implies \text{Repetir proceso del paso 2 al 7} \\ & \implies \sigma_p^2 = \sigma_p^2 + V_{\sigma^2} \end{aligned} \tag{4.35}$$

Además, la cantidad de veces que se repite el proceso entre el segundo y séptimo paso es igual a $V_{\sigma^2}^{-1} \cdot \sigma_{max}^2 + 1$, que se obtiene al dividir el valor máximo de la varianza entre el valor que se le adiciona cada iteración y sumarle uno para tomar en consideración la primera vez que se ejecuta el proceso (cuando $\sigma_p^2 = 0$). En este caso, el valor de σ_{max}^2 se eligió como 100 de tal forma que el valor máximo de la desviación estándar del error de posición fuera de 10 centímetros y además, el valor de V_{σ^2} se eligió como 0.5, lo cual implica que el proceso del paso dos al siete se repite un total de 201 veces ($0,5^{-1} \cdot 100 + 1 = 201$).

La prueba termina justo después de que finaliza la repetición número 201 y la salida son los 14 archivos que contienen los valores de cada una de las medias y varianzas de los errores. Específicamente, cada uno de esos archivos contiene un vector vertical con una longitud igual a 201, cuyas entradas son los valores de la respectiva media o varianza, los cuales se obtuvieron para cada una de las 201 repeticiones del proceso comprendido entre el paso dos y siete. Además, los archivos tienen el nombre de la respectiva media o varianza y son de tipo *.m* que es una extensión utilizada por el software conocido como

MATLAB, de tal forma que si dicho software lee alguno de los archivos, pueda reconocer el vector que contienen. Entonces, por ejemplo, para el caso de la media del error de traslación en X_l , el nombre del archivo correspondiente es *media_error_traslacion_Xl.m* y el formato general es el siguiente:

```
v_media_error_traslacion_Xl=[valor 1
valor 2
valor 3
valor 4
valor 5
.
.
.
.
valor 201
];
```

Es de forma totalmente análoga para el resto de los archivos, donde lo único que varía entre un archivo y otro es el nombre del archivo en sí y el nombre del vector que contiene dentro, que es igual al nombre del archivo pero con una letra *v* al principio, como se vio en el ejemplo anterior. Obviamente también varían los valores de las entradas del vector, pero se mantiene la longitud de 201 y el formato general.

Una vez que se obtienen las salidas de la prueba, se procede a obtener los resultados finales que son las gráficas de todas las medias y varianzas de los errores de traslación, rotación y coincidencia, contra la varianza del error de posición. Para ello, se realizó un pequeño programa en MATLAB que primero lee los 14 archivos con extensión *.m*, que son los vectores de todas las medias y varianzas de los errores, tal y como se detalló anteriormente, y que luego gráfica dichos vectores contra el vector de la varianza del error de posición, el cual va desde 0 hasta 100 en pasos de 0.5. De esta forma se obtuvieron un total de 14 gráficas: 6 gráficas de la media y varianza del error absoluto de rotación en X_l , Y_l y Z_l contra la varianza del error de posición; 6 gráficas de la media y varianza del error absoluto de traslación en X_l , Y_l y Z_l contra la varianza del error de posición; y 2 gráficas de la media y la varianza del error de coincidencia entre nubes contra la varianza del error de posición.

Por último, los parámetros de control de esta segunda prueba, junto con sus respectivos valores, los cuales se mencionaron, se definieron y se eligieron o calcularon a lo largo de toda esta sección, se resumen a continuación:

```

Límites:
  Limite_de_las_componentes_de_traslación(lT): 18
  Limite_de_las_componentes_de_rotación(lW): 10

Iteraciones:
  Numero_de_iteraciones(N_e): 1000

Varianza del error de posición:
  Varianza_maxima_(sigma2_max): 100
  Aumento_de_la_varianza_por_cada_repetición(V_sigma2): 0.5

```

Al igual que con la primera prueba, dichos parámetros y sus respectivos valores, se establecieron de forma externa, mediante un archivo de texto que tiene exactamente el mismo formato de la lista anterior y que se lee en el primer paso de la prueba. Además, cabe aclarar que el número de iteraciones son las veces que se itera el proceso que va desde que se genera una nube de puntos hasta que se calculan los errores, mientras que el número de repeticiones, que se calcula con los dos últimos parámetros de control, son las veces que se repiten las iteraciones anteriores junto con el cálculo de las medias y varianzas de los errores.

4.2.3 Prueba III: algoritmo completo excluyendo la parte de ROS

El fin de esta tercera y última prueba es probar el correcto funcionamiento y determinar la precisión del algoritmo de odometría visual estereoscópico implementado, pero excluyendo todos los procesos que estén relacionados con nodos y tópicos de ROS. Específicamente, el algoritmo consiste en la iteración del proceso de establecimiento de correspondencias por el coeficiente de correlación pseudo normalizada y del proceso de estimación de movimiento por descomposición en valores singulares, mediante un procedimiento conocido como RANSAC que se utiliza para estimar los parámetros de movimiento tridimensionales entre dos instantes de tiempo sucesivos, a partir de correspondencias temporales clasificadas como típicas, excluyendo aquellas que son atípicas, de tal forma que se obtiene una traslación y una rotación que describen el movimiento entre los dos instantes con una mayor exactitud. Esta prueba tiene un total de nueve pasos, que se basan principalmente en ciertos pasos de las pruebas anteriores.

En un primer paso, se lee la imagen de intensidad izquierda de prueba antes del movimiento llamada *intensity1.bmp* y se guardan los valores de intensidad I_{l_j, t_k} de cada uno de los píxeles $\mathbf{p}_{l_j, t_k}(c_{l_j}, r_{l_j})$ de dicha imagen. Además, su ancho w es 640 y su largo h es 480.

Seguidamente, se lee el archivo de texto llamado *Nube1.txt* que contiene la nube de puntos de prueba antes del movimiento. Solo que en este caso no se van a guardar las posiciones tridimensionales de los puntos que conforman dicha nube, debido a que en esta prueba no se van a utilizar nubes de puntos tridimensionales, lo cual se explicara más adelante. En cambio, a partir de la lectura de dicho archivo de texto se van a identificar los píxeles de la imagen de prueba que corresponden a puntos característicos, que en este caso son aquellos que píxeles que tienen una posición tridimensional asociada. Para ello, se va leyendo cada elemento de la lista que contiene el archivo y si el elemento número j , que está asociado con el píxel \mathbf{p}_{l_j, t_k} , no tiene un valor tipo NaN, es decir, es una posición tridimensional válida, entonces el píxel es un punto característico dado por $\mathbf{h}_{l_i, t_k}(c_{l_i}, r_{l_i})$, donde el valor de i es igual al valor de j menos la cantidad de elementos anteriores con un valor tipo NaN. De esta forma se obtiene el subconjunto formado por puntos característicos \mathbf{h}_{l_i, t_k} que se encuentra en la imagen de prueba. Además, la cantidad de puntos \mathbf{h}_{l_i, t_k} , dada por $C_{N_{t_k}}$, es conocida, ya que es la cantidad de elementos que no tienen un valor NaN.

El segundo paso es exactamente igual al segundo paso de la primera prueba, en el cual se genera una imagen de intensidad izquierda después del movimiento de manera sintética, al aplicarle una transformación euclidiana a la imagen izquierda antes del movimiento, que equivale a rotarla un ángulo de θ grados alrededor de un eje perpendicular a la imagen que pasa por su centro y trasladarla t_c píxeles a lo largo del eje horizontal y t_r píxeles a lo largo del eje vertical.

Entonces, se encuentra de forma totalmente aleatoria los parámetros de movimiento bidimensionales, el ángulo de rotación θ y las componentes de la traslación $(t_c, t_r)^T$, dentro de los siguientes intervalos:

$$1 \leq \theta \leq l_\theta \quad (4.36)$$

$$1 \leq t_c, t_r \leq l_t \quad (4.37)$$

En este caso, el límite l_θ del ángulo de rotación es igual a 10 grados, exactamente igual al valor del límite de rotación de la segunda prueba, que se deriva de los resultados de la primera prueba que se discutirán más adelante en el siguiente capítulo. Mientras que el límite l_t de las componentes de traslación es igual a 120 píxeles, valor que se calculó en la prueba número uno, tal que en el peor de los casos se perdiera como máximo un 40 % de la imagen de prueba luego de la transformación.

Luego, a partir de los valores de los parámetros de movimiento 2D elegidos aleatoriamente, se calcula la matriz de transformación euclidiana \mathbf{K} , mediante la siguiente expresión:

$$\mathbf{K} = \begin{pmatrix} \cos(\theta) & -\sin(\theta) & [1 - \cos(\theta)]\frac{w}{2} + \sin(\theta)\frac{h}{2} + t_c \\ \sin(\theta) & \cos(\theta) & [1 - \cos(\theta)]\frac{h}{2} - \sin(\theta)\frac{w}{2} + t_r \end{pmatrix} \quad (4.38)$$

Seguidamente, se utiliza la función de opencv conocida como *warpAffine*, la cual recibe como entrada la imagen de prueba antes del movimiento y la matriz \mathbf{K} , para generar como salida la imagen de intensidad izquierda después del movimiento, al aplicarle la respectiva transformación euclidiana a la imagen de prueba de entrada, que corresponde a una rotación dada por θ y una traslación dada por $(t_c, t_r)^T$.

Es importante aclarar que los píxeles de la imagen después del movimiento se denotan como $\mathbf{p}_{ln,t_{k+1}}(c_{ln}, r_{ln})$ y sus valores de intensidad como $I_{ln,t_{k+1}}$. Además, los puntos característicos $\mathbf{h}_{lm,t_{k+1}}(c_{lm}, r_{lm})$ en dicha imagen se pueden obtener a partir de los puntos característicos \mathbf{h}_{li,t_k} transformados, mediante la siguiente expresión:

$$\mathbf{h}_{lm,t_{k+1}} = \text{round}(\mathbf{K} \cdot \mathbf{h}_{li,t_k}) \quad (4.39)$$

y excluyendo aquellos puntos transformados cuyas componentes se salen de una imagen de 640x480, donde $\text{round}()$ es la función que representa el proceso de redondeo de las componentes de un vector. Además, la cantidad de puntos característicos $\mathbf{h}_{lm,t_{k+1}}$, dada por $C_{N_{t_{k+1}}}$, es conocida, ya que es la cantidad de puntos \mathbf{h}_{li,t_k} transformados que no se salen de la imagen, lo cual implica que es menor a $C_{N_{t_k}}$. Así mismo, para todo valor de m se guarda el respectivo valor de i del punto \mathbf{h}_{li,t_k} transformado valido, en la entrada número m de un vector dado por \mathbf{v}_c , tal que $(\mathbf{h}_{li=\mathbf{v}_c[m],t_k}, \mathbf{h}_{lm,t_{k+1}})$ sea una correspondencia temporal 2D.

En el tercer paso, se introduce un ruido modelado como gaussiano aleatorio, de media cero y varianza conocida, a la imagen izquierda después del movimiento, es decir, a sus valores de intensidad $I_{ln,t_{k+1}}$; de forma análoga al tercer paso de la segunda prueba en el que introdujo un error de posición a la nube después del movimiento. El fin de esto es simular de forma controlada el ruido que perturba a las imágenes cuando el algoritmo se corre o se prueba en tiempo real.

Matemáticamente, el ruido es introducido a la imagen izquierda después del movimiento mediante una simple adición a cada uno de los valores de intensidad $I_{ln,t_{k+1}}$ de la imagen, tal y como se muestra en la siguiente ecuación:

$$I_{n,t_{k+1}} = \text{round}(I_{n,t_{k+1}} + r_n) \quad (4.40)$$

donde r_n es el ruido gaussiano que se adiciona a cada uno de los respectivos valores de intensidad $I_{n,t_{k+1}}$ que conforman la imagen izquierda después del movimiento y $\text{round}()$ es la función que representa el proceso de redondeo

del valor de intensidad con ruido, que es necesario debido a que los valores de intensidad siempre son valores enteros, es decir, un valor de intensidad de un píxel no puede tener un valor decimal, por ejemplo, de 35.2, solo pueden ser números enteros de 0 a 255. Además, el valor de cada r_n se calcula de la siguiente manera:

$$r_n = \sqrt{\sigma_r^2} \cdot a_r = \sigma_r \cdot a_r = \sigma_r \cdot \text{gasdev}() \quad (4.41)$$

donde σ_r^2 es la varianza del ruido gaussiano y σ_r es la respectiva desviación estándar, que se mantienen constantes para el cálculo de todos los r_n . Mientras que la variable a_r es un número aleatorio que se obtiene a partir de una distribución gaussiana con media cero y varianza unitaria, mediante la función $\text{gasdev}()$ del libro de *Numerical Recipes*. Además, al igual que en la segunda prueba, la varianza puede tomar valores de 0 a 100 y por lo tanto, la desviación estándar de 0 a 10.

Entonces, se va recorriendo la imagen izquierda después del movimiento y por cada píxel $\mathbf{p}_{ln,t_{k+1}}$ se calcula el valor del respectivo ruido gaussiano r_n mediante la expresión 4.41 y se adiciona al respectivo valor de intensidad dado por $I_{n,t_{k+1}}$, empleando la expresión 4.40. De esa manera se introduce ruido a la imagen después del movimiento, tal que ahora cada valor de intensidad $I_{n,t_{k+1}}$ esta perturbado por un ruido gaussiano.

En el cuarto paso, se obtienen las nubes de puntos antes y después del movimiento, de forma sintética. Como se mencionó anteriormente, no se van a utilizar nubes de puntos tridimensionales, en cambio, se van a utilizar nubes de puntos bidimensionales para efectos de esta prueba en particular. Una nube de puntos bidimensional se puede definir de manera muy sencilla como un conjunto de puntos bidimensionales, dados de forma general como $\mathbf{p}_i = (x_i, y_i)^T$, que es equivalente a decir que es conjunto de puntos tridimensionales dados por $\mathbf{P}_i = (x_i, y_i, 0)^T$ que se encuentra sobre el plano $Z = 0$.

En este caso, se va a suponer que la nube de puntos antes del movimiento se encuentra sobre el plano de la cámara izquierda, el cual se va tomar como si fuera el plano $Z_l = 0$, de tal forma que el conjunto de puntos $\mathbf{H}_{li,t_k}(X_{li}, Y_{li}, Z_{li})$ que conforman dicha nube, sea exactamente igual al subconjunto de puntos característicos \mathbf{h}_{li,t_k} en la imagen antes del movimiento, como se muestra en la siguiente expresión:

$$\mathbf{H}_{li,t_k} = \mathbf{h}_{li,t_k} \implies \begin{pmatrix} X_{li} \\ Y_{li} \\ Z_{li} \end{pmatrix} = \begin{pmatrix} c_{li} \\ r_{li} \\ 0 \end{pmatrix} \quad (4.42)$$

Así mismo, se va a suponer que la nube de puntos después del movimiento se encuentra sobre el plano de la cámara izquierda, que se toma como $Z_l = 0$, de tal forma que el conjunto de puntos $\mathbf{H}_{lm,t_{k+1}}(X_{lm}, Y_{lm}, Z_{lm})$ que conforman

dicha nube sea exactamente igual al subconjunto de puntos característicos \mathbf{h}_{lm,t_k} en la imagen después del movimiento:

$$\mathbf{H}_{lm,t_{k+1}} = \mathbf{h}_{lm,t_{k+1}} \implies \begin{pmatrix} X_{lm} \\ Y_{lm} \\ Z_{lm} \end{pmatrix} = \begin{pmatrix} c_{lm} \\ r_{lm} \\ 0 \end{pmatrix} \quad (4.43)$$

Es de esta forma como se obtienen las nubes de puntos antes y después del movimiento, pero solo para efectos de esta prueba en particular. Además, la razón por la cual se obtienen de esa forma, a partir de puntos bidimensionales que son los puntos característicos en la respectiva imagen izquierda, es para facilitar la ejecución y planteamiento de la prueba, debido que a pesar de que se conoce la nube de puntos de prueba antes del movimiento, que se encuentra en el archivo *Nube1.txt*, el proceso para obtener la respectiva nube de puntos después del movimiento es bastante complicado, ya que se tendría que encontrar los parámetros de movimiento tridimensionales ($\Delta T_{X_l}, \Delta, T_{Y_l}, \Delta T_{Z_l}, \Delta W_{X_l}, \Delta W_{Y_l}, \Delta W_{Z_l}$) equivalentes a los parámetros de movimiento bidimensionales (t_c, t_r, θ) con los que se transformó la imagen de prueba antes del movimiento, para luego poder generar la nube después del movimiento rotando y trasladando la nube antes del movimiento. En cambio, en este caso se obtienen las nubes de puntos sin tener que encontrar los parámetros de movimiento tridimensionales equivalentes y además, se pueden relacionar entre sí a partir de los parámetros de movimiento bidimensionales, de la siguiente manera:

$$\mathbf{H}_{m,t_{k+1}} = \Delta \mathbf{R} \cdot \mathbf{H}_{i,t_k} + \Delta \mathbf{T} \quad (4.44)$$

donde $\Delta \mathbf{T}$ es el vector de traslación dado por $(\Delta T_{X_l}, \Delta, T_{Y_l}, \Delta T_{Z_l})^T = (t_c, t_r, 0)^T$ y $\Delta \mathbf{R}$ es la matriz de rotación, cuyas entradas se obtienen a partir de las componentes del vector de rotación $\Delta \mathbf{W}$ dado por $(\Delta W_{X_l}, \Delta W_{Y_l}, \Delta W_{Z_l})^T = (0, 0, \theta)^T$.

Es importante destacar que el tamaño de la nube de puntos antes del movimiento es igual $C_{N_{t_{k+1}}}$ que es la cantidad de puntos característicos en la imagen antes del movimiento. Análogamente, el tamaño de la nube de puntos después del movimiento es igual $C_{N_{t_k}}$ que es la cantidad de puntos característicos en la imagen después del movimiento. Además, las correspondencias temporales 3D son conocidas y están dadas por $(\mathbf{H}_{li=\mathbf{v}_c[m],t_k}, \mathbf{H}_{lm,t_{k+1}})$, que son exactamente iguales a las correspondencias temporales 2D dadas por $(\mathbf{h}_{li=\mathbf{v}_c[m],t_k}, \mathbf{h}_{lm,t_{k+1}})$.

En el quinto paso se ejecuta el algoritmo de odometría visual estereoscópico implementado, descrito en el capítulo 3, pero excluyendo los procesos basados en ROS. El algoritmo sin ROS consiste en la iteración del proceso de establecimiento de correspondencias temporales entre las imágenes y nubes de

puntos antes y después del movimiento, usando el coeficiente de correlación pseudo normalizada y del proceso de estimación de movimiento minimizando un error cuadrático medio que depende de las correspondencias temporales 3D, empleando el método de descomposición de valores singulares. Dicha iteración se lleva a cabo mediante procedimiento conocido como RANSAC para estimar los parámetros de movimiento (traslación y rotación) a partir de correspondencias temporales que fueron clasificadas como típicas, es decir, excluyendo correspondencias atípicas. En este caso, dicho algoritmo recibe como entrada las imágenes y nubes de puntos antes y después del movimiento que se obtuvieron en los pasos anteriores y genera como salida los parámetros de movimiento bidimensionales estimados, es decir, el vector de traslación $\Delta \mathbf{T}'$ que en este caso está dado por $(\Delta T'_{X_i}, \Delta, T'_{Y_i}, \Delta T'_{Z_i})^T = (t'_c, t'_r, 0)^T$ y la matriz de rotación $\Delta \mathbf{R}'$, que a partir de sus entradas se obtiene el respectivo vector de rotación $\Delta \mathbf{W}'$ que esta dado por $(\Delta W'_{X_i}, \Delta W'_{Y_i}, \Delta W'_{Z_i})^T = (0, 0, \theta')^T$.

Además, se utilizaron los siguientes valores para los parámetros de control del algoritmo de odometría visual estereoscópico implementado:

Imagen y ventanas:

```
Ancho_de_la_imagen(w): 640
Largo_de_la_imagen(h): 480
Ancho_de_la_ventana_de_correlacion(wc): 11
Largo_de_la_ventana_de_correlacion(hc): 11
Ancho_de_la_ventana_de_búsqueda(wb): 275
Largo_de_la_ventana_de_búsqueda(hb): 275
```

Coefficiente de similitud y algoritmo de RANSAC:

```
Coefficiente_de_similitud_minimo(cpn_min): 0.9
Consenso aceptable_o_cantidad_de_correspondencias(Cq): 6
Umbral_de_distancia(dq_max): 0.35
Numero_de_iteraciones(Nq): 50
```

Cabe aclarar que los valores del ancho y alto de la ventana de correlación y de la ventana de búsqueda son los mismos valores que se utilizaron en la primera prueba, que se eligieron mediante prueba y error para la ventana de correlación y para la ventana de búsqueda se calcularon de tal forma que en el peor de los casos la mejor coincidencia de un píxel estuviera dentro de la ventana. El valor del coeficiente de similitud mínimo es igual al valor que se utilizó en la primera prueba y que se eligió por prueba y error. Los parámetros del procedimiento de RANSAC se eligieron mediante prueba y error, a excepción del umbral de distancia que se obtuvo al dividir entre 2 el valor de la distancia mínima utilizada en la primera prueba, que era 0.7. Se pudo

haber utilizado un valor de 0.7 para dicho umbral de distancia, pero utilizo la mitad de dicho valor para obtener una mayor exactitud en los parámetros de movimiento estimados. Además, dichos parámetros de control y sus respectivos valores se establecen de forma externa a la prueba, con archivo de texto que es leído en el primer paso de la prueba y que tiene el mismo formato que tiene lista mostrada anteriormente.

En el sexto paso, se calcula el error absoluto de rotación alrededor de Z_l , dado por $e_{W_{Z_l},u}$, así como el error absoluto de traslación a lo largo de X_l y Y_l , dados por $e_{T_{X_l},u}$ y $e_{T_{Y_l},u}$, mediante una simple resta entre las respectivas componentes de los vectores de movimiento reales ($\Delta\mathbf{W}$ y $\Delta\mathbf{T}$) y las componentes de los vectores de movimiento estimados ($\Delta\mathbf{W}'$ y $\Delta\mathbf{T}'$), de la siguiente manera:

$$e_{W_{Z_l},u} = \text{abs}(\Delta W'_{Y_l} - \Delta W_{Y_l}) = \text{abs}(\theta' - \theta) \quad (4.45)$$

$$e_{T_{X_l},u} = \text{abs}(\Delta T'_{X_l} - \Delta T_{X_l}) = \text{abs}(t'_c - t_c) \quad (4.46)$$

$$e_{T_{Y_l},u} = \text{abs}(\Delta T'_{Y_l} - \Delta T_{Y_l}) = \text{abs}(t'_r - t_r) \quad (4.47)$$

Es importante notar que a diferencia de la segunda prueba, las nubes de puntos son bidimensionales y por lo tanto, los parámetros de movimiento reales y estimados también lo son, lo que quiere decir que el error de rotación alrededor de X_l y Y_l y el error de traslación a lo largo de Z_l no existen. Además, la variable u se usa para representar el u -ésimo error de traslación en X_l y Y_l y de rotación en Z_l , cuyo valor va desde 1 hasta N_e , que es la cantidad de veces que se va a iterar este paso y los anteriores, proceso que se explicara más adelante. Mientras que $\text{abs}()$ es una función que representa el proceso de obtener el valor absoluto de un número.

Seguidamente, se calcula el error de coincidencia entre las nubes de puntos antes y después del movimiento, de forma análoga al quinto paso de la segunda prueba. Dicho error es la media de la distancia euclidiana entre los puntos $\mathbf{H}_{m,t_{k+1}}$ de la nube después del movimiento y los puntos correspondientes $\mathbf{H}_{i=v[m],t_k}$ de la segunda nube de puntos después del movimiento, rotados y trasladados con $\Delta\mathbf{R}'$ y $\Delta\mathbf{T}'$. Entonces, el u -ésimo error de coincidencia se denota como $e_{c,u}$ y se calcula así:

$$e_{c,u} = \frac{1}{C_{N_{t_{k+1}}}} \sum_{m=1}^{C_{N_{t_{k+1}}}} |\mathbf{H}_{m,t_{k+1}} - (\Delta\mathbf{R}' \cdot \mathbf{H}_{i=v_c[m],t_k} + \Delta\mathbf{T}')|^2 \quad (4.48)$$

En el séptimo paso es muy parecido al sexto de la segunda prueba, donde se iteran todos los pasos anteriores una cantidad de N_e veces, a excepción del

primer paso. Los pasos que se iteran son los siguientes: (2) se elige aleatoriamente nuevos valores para los parámetros de movimiento bidimensionales, $\Delta \mathbf{T} = (t_c, t_r, 0)^T$ y $\Delta \mathbf{W}' = (0, 0, \theta)^T$, y a partir de estos se calcula la matriz de transformación euclidiana \mathbf{K} , para luego generar una nueva imagen después del movimiento, mediante la función *warpaffine*, que realiza una transformación euclidiana a la imagen antes del movimiento; (3) se introduce un ruido aleatorio gaussiano, de media cero y varianza σ_r^2 , a la imagen después del movimiento; (4) se generan las nubes de puntos antes y después del movimiento a partir de los subconjuntos de puntos característicos de la respectiva imagen; (5) se ejecuta el algoritmo de odometría visual estereoscópico, excluyendo los procesos de ROS, para obtener los parámetros de movimiento estimados, $\Delta \mathbf{T}' = (t'_c, t'_r, 0)^T$ y $\Delta \mathbf{R}'$, a partir de correspondencias temporales 3D clasificadas como típicas y además, se obtiene el vector $\Delta \mathbf{W}' = (0, 0, \theta')^T$ a partir de las entradas de $\Delta \mathbf{R}'$; (6) se calculan los errores de traslación $e_{T_{X_l},u}$ y $e_{T_{Y_l},u}$ y el error de rotación $e_{W_{Z_l},u}$, así como el error de coincidencia $e_{c,u}$. Además, al final de cada iteración se guardan los valores de los cuatro errores anteriores, así mismo, N_e se eligió como 1000.

El octavo paso se lleva a cabo de forma análoga al séptimo paso de la segunda prueba, se calcula la media y la varianza del error de traslación en X_l y en Y_l , del error de rotación en Z_l y del error de coincidencia, a partir de los diferentes valores de cada error, $e_{T_{X_l},u}$, $e_{T_{Y_l},u}$, $e_{W_{Z_l},u}$ y $e_{c,u}$, que se obtuvieron para cada iteración y que forman un conjunto de valores de tamaño N_e .

La media del error de traslación en X_l se denota como $\mu_{e_{T_{X_l}}}$, la del error de traslación en Y_l como $\mu_{e_{T_{Y_l}}}$, la de error de rotación en Z_l como $\mu_{e_{W_{Z_l}}}$ y la del error de coincidencia como μ_{e_c} y se calculan de la siguiente forma:

$$\mu_{e_{T_{X_l}}} = \frac{1}{N_e} \sum_{u=1}^{N_e} e_{T_{X_l},u} \quad (4.49)$$

$$\mu_{e_{T_{Y_l}}} = \frac{1}{N_e} \sum_{u=1}^{N_e} e_{T_{Y_l},u} \quad (4.50)$$

$$\mu_{e_{W_{Z_l}}} = \frac{1}{N_e} \sum_{u=1}^{N_e} e_{W_{Z_l},u} \quad (4.51)$$

$$\mu_{e_c} = \frac{1}{N_e} \sum_{u=1}^{N_e} e_{c,u} \quad (4.52)$$

Mientras que la varianza del error de traslación en X_l se denota como $\sigma_{e_{T_{X_l}}}^2$, la del error de traslación en Y_l como $\sigma_{e_{T_{Y_l}}}^2$, la de error de rotación en Z_l

como $\sigma_{e_{WZ_l}}^2$ y la del error de coincidencia como $\sigma_{e_c}^2$ y se calculan de la siguiente forma:

$$\sigma_{e_{TX_l}}^2 = \frac{1}{N_e} \left(\sum_{u=1}^{N_e} e_{TX_l,u}^2 \right) - \mu_{e_{TX_l}}^2 \quad (4.53)$$

$$\sigma_{e_{TY_l}}^2 = \frac{1}{N_e} \left(\sum_{u=1}^{N_e} e_{TY_l,u}^2 \right) - \mu_{e_{TY_l}}^2 \quad (4.54)$$

$$\sigma_{e_{WZ_l}}^2 = \frac{1}{N_e} \left(\sum_{u=1}^{N_e} e_{WZ_l,u}^2 \right) - \mu_{e_{WZ_l}}^2 \quad (4.55)$$

$$\sigma_{e_c}^2 = \frac{1}{N_e} \left(\sum_{u=1}^{N_e} e_{c,u}^2 \right) - \mu_{e_c}^2 \quad (4.56)$$

Cada una de las medias y las varianzas que se calculan con las ecuaciones de 4.49 a la 4.56, se guardan en un archivo tipo *.m* que tiene el nombre de la respectiva media o varianza y que tienen exactamente el mismo formato que los archivos utilizados en la segunda prueba. Dicho formato es el de un vector vertical que puede ser reconocido por MATLAB si el archivo es leído por este, tal y como se detalló al final de la sección anterior. Además, en este caso la cantidad total de archivos es 8, en lugar de 14 como en la segunda prueba, debido a que el error de traslación en Z_l y los errores de rotación en X_l y Y_l no existen y por lo tanto, sus medias y varianzas tampoco.

En el noveno y último paso, se repite todo el proceso que va desde el segundo paso hasta el paso anterior, es decir, desde que se genera la imagen después del movimiento hasta que se calculan y se guardan las medias y varianzas de los errores. Para cada repetición se varia la desviación estándar σ_r del ruido que se adiciona a la imagen después del movimiento, a diferencia del octavo paso de la segunda prueba en el que se varia la varianza del error de posición. El valor de σ_r se varia de tal forma que empiece en 0 y se va aumentando en un valor igual a V_σ cada vez que termina una repetición y así consecutivamente hasta que se llega a un valor máximo de dicha desviación, dado por σ_{max} . Este paso se puede resumir de la siguiente manera:

$$\begin{aligned} & \sigma_r = 0 \\ & \text{Mientras que } (\sigma_r \leq \sigma_{max}) : \\ & \implies \text{Repetir proceso del paso 2 al 8} \\ & \implies \sigma_r = \sigma_r + V_\sigma \end{aligned} \quad (4.57)$$

Además, de forma análoga a la segunda prueba, la cantidad de veces que se repiten los pasos del dos al ocho es igual a $V_\sigma^{-1} \cdot \sigma_{max} + 1$. El valor de

σ_{max} se eligió como 10 de tal forma que el valor máximo de la varianza sea 100, mientras que el valor de V_σ se eligió como 0.5, lo que implica que el proceso comprendido entre el paso dos y ocho se repite un total de 21 veces ($0,5^{-1} \cdot 10 + 1 = 21$).

Es importante aclarar que la razón por la cual en esta prueba se varia la desviación estándar y no la varianza, en comparación con la segunda prueba, es debido a que si se varia la varianza de 0 a 100 en pasos de 0.5, la media de la diferencia del valor de la desviación estándar (raíz cuadrada de la varianza) entre repeticiones consecutivas sería de aproximadamente 0.1, y se obtendrían prácticamente los mismos resultados (medias y varianzas de los errores) para cada dos, tres o cuatro repeticiones seguidas. Lo anterior se debe a que los valores de intensidad de los píxeles de la imagen después del movimiento, se tienen que redondear luego de que se les adiciona ruido (ecuación 4.40) y ese redondeo prácticamente elimina lo único que diferencia a las repeticiones consecutivas, es decir, esa pequeña diferencia que existe entre el valor de la desviación estándar de repeticiones sucesivas se ve suprimida por el redondeo, lo que implica que la imagen después del movimiento con ruido es prácticamente la misma para cada dos, tres o cuatro repeticiones. Es por eso que se decidió variar la desviación estándar de 0 a 10 en pasos de 0.5, para asegurarse que la diferencia de su valor entre repeticiones consecutivas siempre sea de 0.5, de tal forma que las imágenes después del movimiento con ruido si se diferencien entre dos repeticiones seguidas y se obtengan resultados diferentes.

Entonces, la prueba finaliza luego de que se haya ejecutado la repetición número 21 y su salida son los 8 archivos con extensión *.m*, los cuales tienen un vector vertical de longitud igual a 21, de tal forma que sus entradas son los valores de las respectivas medias y varianzas de los errores que se obtuvieron para cada una de las 21 repeticiones. Además, dichos vectores pueden ser reconocidos por el software coincido como MATLAB cuando este lea los archivos.

Una vez que se obtienen las salidas de la prueba, se procede a obtener los resultados finales que son las gráficas de todas las medias y varianzas de los errores contra la varianza del ruido introducido en la imagen después del movimiento. Para ello, se utilizó una versión modificada del programa de MATLAB que se empleó en la segunda prueba para obtener las gráficas, de tal forma que lee los 8 archivos con extensión *.m*, que son los vectores de las medias y varianzas de los errores y los gráfica contra el vector de la varianza del ruido introducido en la imagen después del movimiento, el cual se obtiene al elevar al cuadrado un vector que va de 0 a 10 en pasos de 0.5 (la varianza es el cuadrado de la desviación). De esta forma se obtuvieron un total de 8 gráficas: 4 gráficas de la media y la varianza del error de traslación en X_l y Y_l contra la varianza del ruido; 2 gráficas para la media y varianza del error de rotación en Z_l contra la varianza del ruido; y gráficas de la media y la varianza

del error de coincidencia entre nubes contra la varianza del ruido.

Por último, los parámetros de control de esta tercera prueba se resumen a continuación, junto con sus valores:

```
Limites:
  Limite_de_las_componentes_de_traslación(lT): 120
  Limite_de_las_componentes_de_rotación(lW): 10

Iteraciones:
  Numero_de_iteraciones(N_e): 1000

Desviacion estandar del ruido gaussiano:
  Desviacion_maxima_(sigma_max): 100
  Aumento_de_la_desviacion_por_cada_repetición(V_sigma): 0.5
```

Dichos parámetros y sus respectivos valores, se establecieron de forma externa, mediante un archivo de texto que tiene exactamente el mismo formato de la lista anterior y que se lee en el primer paso de la prueba, de forma análoga a los parámetros de control que se utilizaron para el algoritmo de odometría visual estereoscópico ejecutado en esta prueba.

5 Resultados de las pruebas y análisis

5.1. Resultados de la prueba I

Los resultados de los 19 casos que se llevaron a cabo en la primera prueba, los cuales se plantearon de acuerdo a todas las posibles combinaciones de los parámetros de movimiento 2D (t_c, t_r, θ) usados para transformar la imagen de entrada de prueba y de acuerdo a los distintos valores que puede tomar el límite de rotación ($l_\theta=0, 5, 10, 15$ o 20 grados), se resumen desde la tabla 5.1 hasta la tabla 5.5, en las cuales se muestra para cada caso las cantidades de correspondencias clasificadas como falsos negativos (C_{FN}), verdaderos negativos (C_{VN}), falsos positivos (C_{FP}) y verdaderos positivos (C_{VP}), así como los valores de las respectivas métricas que se obtienen a partir de dichas cantidades, que son la precisión (P_m), la sensibilidad (S_m) y el valor F_1 .

Casos		Cantidades				Métricas (%)		
#	Parámetros	C_{FN}	C_{VN}	C_{FP}	C_{VP}	P_m	S_m	F_1
1	t_c	135	13729	5107	181029	97.3	99.9	98.6
2	t_r	181	17786	7290	174743	96	99.9	97.9
3	t_c, t_r	277	29628	11922	158173	93	99.8	96.3

Cuadro 5.1: Resultados de los casos con un límite de rotación igual a 0 grados ($l_\theta = 0^\circ$).

Casos		Cantidades				Métricas (%)		
#	Parámetros	C_{FN}	C_{VN}	C_{FP}	C_{VP}	P_m	S_m	F_1
4	θ	5300	2345	6430	185925	96.6	97.2	96.9
5	t_c, θ	4873	14842	10545	169740	94.1	97.2	95.6
6	t_r, θ	4834	18488	12106	164572	93.1	97.1	95.1
7	t_c, t_r, θ	4299	29452	16278	149971	90.2	97.2	93.6

Cuadro 5.2: Resultados de los casos con un límite de rotación igual a 5 grados ($l_\theta = 5^\circ$).

Casos		Cantidades				Métricas (%)		
#	Parámetros	C _{FN}	C _{VN}	C _{FP}	C _{VP}	P _m	S _m	F ₁
8	θ	21120	5343	16915	156622	90.2	88.1	89.2
9	t_c, θ	19291	17107	20604	142998	87.4	88.1	87.8
10	t_r, θ	18625	20943	21754	138678	86.4	88.1	87.3
11	t_c, t_r, θ	17008	31331	25301	126360	83.3	88.1	85.7

Cuadro 5.3: Resultados de los casos con un límite de rotación igual a 10 grados ($l_\theta = 10^\circ$).

Casos		Cantidades				Métricas (%)		
#	Parámetros	C _{FN}	C _{VN}	C _{FP}	C _{VP}	P _m	S _m	F ₁
12	θ	40334	11471	31851	116344	78.5	74.3	76.3
13	t_c, θ	36902	22058	33949	107091	75.9	74.4	75.1
14	t_r, θ	36060	24989	35026	103925	74.8	74.2	74.5
15	t_c, t_r, θ	32708	34701	37524	95067	71.7	74.4	73.0

Cuadro 5.4: Resultados de los casos con un límite de rotación igual a 15 grados ($l_\theta = 15^\circ$).

Casos		Cantidades				Métricas (%)		
#	Parámetros	C _{FN}	C _{VN}	C _{FP}	C _{VP}	P _m	S _m	F ₁
16	θ	40761	28386	42133	88720	67.8	68.5	68.2
17	t_c, θ	37537	37090	43816	81557	65.1	68.5	66.7
18	t_r, θ	36731	39340	44437	79492	64.1	68.4	66.2
19	t_c, t_r, θ	34012	47038	46173	72777	61.2	68.2	64.5

Cuadro 5.5: Resultados de los casos con un límite de rotación igual a 20 grados ($l_\theta = 20^\circ$).

Cabe destacar que los casos y por lo tanto, sus resultados, se agruparon de acuerdo al valor del límite de rotación, ya que es más fácil analizar y comparar los resultados de varios grupos conformados por casos que comparten algo en común, en lugar de los resultados de cada caso de forma individual. Entonces, como el límite de rotación puede tomar cinco valores diferentes, hay un total de cinco grupos de casos y cada uno tiene asociado una tabla con los respectivos resultados, de la siguiente manera: la tabla 5.1 resume los resultados del primer

grupo de casos que es cuando el límite de rotación es igual a 0 grados, la tabla 5.2 los del segundo grupo que es cuando el límite de rotación es igual a 5 grados, la tabla 5.3 los del tercer grupo que es cuando el límite de rotación es igual a 10 grados, la tabla 5.4 los del cuarto grupo que es cuando el límite de rotación es igual a 15 grados y finalmente, la tabla 5.5 los del quinto y último grupo que es cuando el límite de rotación es igual a 20 grados. Además, cada grupo está conformado por un total de cuatro casos, a excepción del primer grupo que solamente tiene tres casos, debido a que en dicho grupo no existe el caso donde solamente se toma en cuenta el ángulo de rotación, ya que su límite es de 0 grados.

Es importante aclarar que para cada tabla, si en la columna correspondiente a los parámetros de un caso en particular se muestra la componente de traslación horizontal (t_c) o de traslación vertical (t_r), quiere decir que dicha componente si se tomó en cuenta para ese caso, es decir, si hubo traslación vertical o horizontal y su límite fue de 120 píxeles. Así mismo, si se muestra el ángulo de rotación (θ), quiere decir que si se tomó cuenta para ese caso, es decir, si hubo rotación y que su límite fue igual al especificado en la descripción de la tabla, que varía de acuerdo al respectivo grupo. Además, si alguno de los parámetros anteriores no se muestra, es debido a que no se tomó en cuenta para ese caso y por lo tanto, su límite es igual a 0, lo que implica que no hubo rotación o traslación vertical o traslación horizontal.

Primero, se analizan los resultados de cada grupo, es decir, se analizan los datos de cada tabla, principalmente los valores de las tres métricas y se van comparando con los resultados del respectivo grupo anterior. Posteriormente, se realiza un análisis de los resultados generales que se pueden inferir o deducir de los valores de las métricas de los cinco grupos, es decir, de aquellos resultados implícitos que se repiten en cada grupo y que por lo tanto, se pueden generalizar.

En la tabla 5.1 se muestran los resultados del primer grupo formado por aquellos casos en los que no se tomó en cuenta la rotación. En dicha tabla, se puede apreciar que en el peor de los casos, en el que se tomaron en cuenta las dos componentes de traslación, se obtuvo una precisión de 93 %, una sensibilidad de 99.8 % y un valor F_1 de 96.3 %, los cuales son unos resultados excelentes debido a que son porcentajes bastante altos que se encuentran por encima del 90 %, junto con los valores de las métricas de los demás casos que fueron incluso mayores. Estos resultados tan buenos demuestran que el proceso de establecimiento de correspondencias temporales funciona bastante bien en aquellas situaciones donde no hay rotación del todo, como cuando el rover se mueve solamente en línea recta, ya que en el peor de los casos, cuando el proceso considera una correspondencia establecida como correcta, está en lo cierto un 93 % de las veces o dicho de otra forma, el proceso se equivoca tan solo un 7 % de todas las veces que considera una correspondencia establecida

como correcta y además, el proceso considera como correctas un 99,8% de todas las correspondencias establecidas que verdaderamente lo son o dicho de otra forma, el proceso considera erróneamente como incorrectas tan solo un 0,2% de las correspondencias establecidas que verdaderamente son correctas.

En la tabla 5.2 se presentan los resultados del segundo grupo formado por los casos en los que si se consideró la rotación, pero con un límite de 5 grados. En este grupo, el peor de los casos sigue siendo aquel en el que se tomaron en cuenta la mayor cantidad de parámetros de movimiento, que son un total de tres, las dos componentes de traslación y el ángulo de rotación, en el cual se obtuvo una precisión de 90,2%, una sensibilidad de 97,2% y un valor F_1 de 96,3%, que siguen siendo resultados excelentes ya que se encuentran por encima del 90%, al igual que en el peor caso del primer grupo. Así mismo, las métricas obtenidas para el resto de los casos, en los que se consideró una menor cantidad de parámetros, son mayores a las obtenidas en el peor caso, como es de esperarse. Estos resultados demuestran que el proceso de establecimiento de correspondencias no solamente funciona bastante bien en aquellas situaciones donde no hay rotación, sino que también en aquellas donde si hubo una rotación pero de hasta 5 grados como máximo, es decir, el funcionamiento del proceso es bastante bueno cuando el rover experimenta una rotación muy pequeña o nula, a parte de una traslación horizontal o vertical o ambas, ya que en la peor situación posible, cuando el proceso considera una correspondencia establecida como correcta, está en lo cierto un 90,2% de las veces y además, considera como correctas un 97,2% de todas las correspondencias establecidas que verdaderamente lo son. Además, si se compara una a una las métricas de los últimos tres casos de la tabla 5.2, excluyendo el caso donde solo hubo rotación, con los respectivos casos de la tabla 5.1, se puede apreciar que las métricas solamente disminuyeron en aproximadamente un 3%, que es una disminución bastante pequeña y confirma aún más lo dicho anteriormente de que el proceso funciona bastante bien para ambas situaciones, sin o con rotación de hasta 5 grados, ya que las métricas difieren en un valor muy pequeño y son mayores a 90%.

Los resultados del tercer grupo formado por los casos en los que se utilizó un límite de rotación de 10 grados, se muestran en la tabla 5.3. Al igual que en el grupo anterior, el peor caso se da cuando se consideran los tres parámetros de movimiento, en el cual se logró una precisión de 83,3%, una sensibilidad de 88,1% y un valor F_1 de 85,7%, los cuales dejan de ser resultados excelentes, como los obtenidos en los grupos anteriores y pasan a ser resultados catalogados como buenos, debido a que se encuentran por debajo del 90%, pero por encima del 80%, al igual que las métricas de los demás casos del grupo. Lo anterior se debe a que hubo una disminución de aproximadamente un 8% de las métricas, que se deduce al comparar las métricas de los casos de la tabla 5.3, con las de los respectivos casos de la tabla 5.2 y además, es mayor

al doble de la disminución que hubo entre las métricas del primer grupo y el segundo grupo, que fue de un 3%. Esto evidencia una clara degradación en el funcionamiento del proceso de establecimiento de correspondencias temporales cuando el límite de rotación se aumenta de 5 grados a 10 grados que es el doble, es decir, cuando el límite de rotación deja de ser tan pequeño como 5 grados, de tal forma que ahora en la peor situación, cuando el proceso considera una correspondencia establecida como correcta, está en lo cierto un 83,3% de las veces en lugar de 90,2% y además, considera como correctas un 88,1% de todas las correspondencias establecidas que verdaderamente lo son, en lugar de un 97,2%. Entonces, se podría decir que en aquellas situaciones donde la rotación máxima es de 10 grados, el funcionamiento del proceso es bueno, pero no tan bueno como en aquellas situaciones donde la rotación es pequeña (menor a 5 grados) o nula, debido a que las métricas disminuyeron un 8% y son menores a 90%.

Los resultados del cuarto grupo formado por aquellos casos en los que se usó un límite de 15 grados, se resumen en la tabla 5.4. Se puede apreciar que en el peor de los casos de este grupo, se obtuvo una precisión de 71,7%, una sensibilidad de 74,4% y un valor F_1 de 73%, los cuales son resultados regulares, es decir, que no son ni buenos ni malos, a causa de que son menores a 80% pero mayores a 70%, que también se cumple con las métricas de los demás casos de este grupo. Lo que paso es que hubo una disminución en los valores de las métricas, el cual fue esta vez de aproximadamente un 13%, un 5% más que la disminución de las métricas del tercer grupo y que se obtiene al comparar las métricas de los casos de la tabla 5.3, con las de los respectivos casos de la tabla 5.4. Lo anterior implica que el funcionamiento del proceso de establecimiento de correspondencias se degradó una vez más, lo cual es de esperarse debido a que el límite de rotación se aumentó esta vez de 10 grados a 15 grados y por lo tanto, ahora en la peor situación, cuando el proceso considera una correspondencia establecida como correcta, está en lo cierto un 71,7% de las veces en lugar de 83,3% y además, considera como correctas un 74,4% de todas las correspondencias establecidas que verdaderamente lo son, en lugar de un 88,1%. Entonces, cuando la rotación máxima es de 15 grados el funcionamiento del proceso pasa a ser aceptable y deja de ser bueno como cuando la rotación máxima es de 10 grados, ya que las métricas estuvieron por debajo de 80% producto de una disminución de 13%.

En la tabla 5.5 se presentan los resultados del quinto y último grupo constituido por aquellos casos en los que se empleó un límite de rotación de 20 grados. En dicha tabla se puede apreciar que los valores de todas las métricas de los cuatro casos estuvieron por debajo del 70% y por encima del 60%, los cuales en realidad son resultados malos, si se comparan con los resultados de los otros grupos. Esta vez hubo una disminución de las métricas de aproximadamente un $8 \pm 2\%$, al comparar las métricas de los casos de la tabla 5.4,

con las de los respectivos casos de la tabla 5.5 y que de forma inesperada fue menor a la disminución de las métricas del tercer grupo, pero que aun así basto para que los valores de las métricas bajaran a menos de un 70 %. Además, como es de esperarse, el funcionamiento del proceso de establecimiento de correspondencias se degrado una tercera vez, debido a que se aumentó el límite de rotación otros 5 grados, es decir, se aumentó de 15 grados a 20 grados. Entonces, cuando la rotación llega hasta los 20 grados, el funcionamiento del proceso pasa de ser aceptable a ser deficiente o prácticamente inaceptable, ya que los valores de las métricas son menores a 70 %.

Ahora surge la siguiente pregunta: de acuerdo a los resultados anteriores, ¿cuál debería de ser el límite de rotación máximo permitido? Para responder esa pregunta primero hay que aclarar que no se va a tolerar un funcionamiento aceptable o deficiente del proceso de establecimiento de correspondencias temporales, ya que eso podría afectar al resto de los procesos primordiales que conforman al algoritmo de odometría visual estereoscópico implementado y por lo tanto, en el peor de los casos disminuir o afectar la exactitud de los parámetros de movimiento tridimensionales estimados, que es la salida final del algoritmo e inclusive aumentar el tiempo de ejecución. Es por esa razón que el límite de rotación máximo permitido, es decir, la máxima rotación que puede experimentar el rover entre dos instantes de tiempo consecutivos, debe de ser de 10 grados, esto para asegurarse de que el proceso de establecimiento de correspondencias temporales vaya a tener un buen funcionamiento en la peor situación posible, es decir, que en el peor de los casos cuando el proceso considere una correspondencia establecida como correcta, que este en lo cierto un 80 % o más de las veces y además, que considere como correctas un 80 % o más de todas las correspondencias establecidas que verdaderamente lo son. O dicho de otra manera, para garantizar un funcionamiento bueno o excelente del proceso, el límite de rotación tiene que ser igual o menor a 10 grados, debido a que la precisión y la sensibilidad en esos casos es mayor a 80 %. Además, si en algún caso particular la rotación del rover es mayor a 10 grados, no quiere decir que el proceso de establecimiento de correspondencias no vaya a funcionar del todo, simplemente lo que va a pasar es que la probabilidad de establecer una correspondencia erróneamente va a aumentar y por lo tanto, puede que en el peor de los casos disminuya la exactitud de los parámetros de movimiento estimados por el algoritmo, debido que el procedimiento basado en RANSAC no pudo excluir todas las correspondencias erróneas. Aunque lo más probable es que si pueda excluir la mayoría de las correspondencias erróneas, pero que le tome más tiempo del que debería y por lo tanto, el tiempo de estimación aumentaría considerablemente, que es algo que se desea evitar.

Es importante aclarar la razón principal por la cual la precisión, la sensibilidad y el valor F_1 disminuyen de un grupo a otro conforme se aumenta el límite de rotación, de tal forma que el funcionamiento del proceso se va

degradando cada vez más, tal y como se analizó en los párrafos anteriores. Lo que pasa es que cuando hubo una rotación, nunca se va poder rastrear correctamente en la imagen después del movimiento, la ventana de correlación que se establece en la imagen antes del movimiento, alrededor de un punto característico. Lo anterior se debe a que la ventana de correlación en la imagen después del movimiento que verdaderamente coincide de forma exacta con la ventana de correlación en la imagen antes del movimiento, se encuentra rotada, de la misma forma en como la imagen después del movimiento se encuentra rotada respecto a la imagen antes del movimiento. Y resulta que cuando el proceso está buscando la ventana más similar en la imagen después del movimiento, en ningún momento compara la ventana de correlación en la imagen antes del movimiento, con una ventana de correlación en la imagen después del movimiento que este rotada, siempre la compara con ventanas de correlación que tienen la misma rotación de 0 grados. Esto ocasiona que la ventana en la imagen después del movimiento que el proceso determino como la que tiene la mayor similitud con la ventana antes del movimiento, nunca va a ser la ventana verdaderamente correcta y entonces, siempre va a existir un error que va a ir aumentando conforme aumenta la rotación, lo cual implica que cada vez se van a establecer más correspondencias erróneas que se van a considerar como correctas o también, se van a establecer más correspondencias verdaderamente correctas que fueron excluidas debido a que el coeficiente de similitud fue menor a 0.9, producto de que las ventanas no son totalmente similares.

Por otro lado, de la tabla 5.1 a la tabla 5.5, se puede notar que para todos los casos el valor de la precisión es más bajo que el valor de la sensibilidad, a excepción de los casos 8, 12 y 13. Lo anterior quiere decir que el proceso de establecimiento de correspondencias es más propenso a cometer falsos positivos (correspondencias establecidas consideradas falsamente como correctas) que a cometer falsos negativos (correspondencias establecidas consideradas falsamente como incorrectas). Esto en realidad es un resultado no deseado debido a que es mejor tener más falsos negativos que falsos positivos, es decir, es mejor que el proceso se equivoque más a la hora de considerar correspondencias como incorrectas, ya que simplemente se estarían rechazando correspondencias verdaderamente correctas, lo cual no tiene ninguna repercusión de peso, en lugar de que se equivoque más cuando considera correspondencias como correctas, lo cual aumenta la posibilidad de que se dejen pasar correspondencias erróneas que pueden comprometer los posteriores procesos del algoritmo de odometría visual estereoscópico. Aunque en realidad no es un problema tan grave ya que este proceso aporta solamente una primera etapa para el rechazo de correspondencias incorrectas y como se mencionó anteriormente, es el proceso basado en RANSAC el que verdaderamente se encarga de rechazarlas. Entonces, al final no es tan malo que se dejen pasar ciertas correspondencias incorrectas, ya que el método de RANSAC va a terminar excluyéndolas. Lo que si puede pasar,

como también se mencionó anteriormente, es que entre más correspondencias incorrectas más tiempo le va a tomar a dicho método excluirlas y por lo tanto, el tiempo de ejecución va a aumentar, pero eso se evita al establecer el límite de la rotación como 10 grados, para garantizar que la posibilidad de que se dejen pasar correspondencias erróneas sea relativamente baja.

Otro resultado importante que se puede inferir de la tabla 5.1 a la tabla 5.5 es que para un mismo grupo de casos, la precisión decrece alrededor de un 3% conforme se consideran más parámetros de movimiento, es decir, decrece en un 3% de un caso donde solo hubo rotación a un caso donde hubo rotación y traslación vertical u horizontal y luego decrece otro 3% de un caso como el anterior a un caso donde hubo rotación y traslación en los dos ejes. Lo anterior es un resultado no deseado pero si esperado, ya que tiene mucho sentido que al proceso le cueste cada vez más establecer una correspondencia verdaderamente correcta conforme se consideran más parámetros de movimiento, debido a que la imagen después del movimiento va a diferir cada vez más de la imagen antes del movimiento y por lo tanto, el rastreo en la imagen después del movimiento, de la ventana de correlación que se establece en la imagen antes del movimiento, va a ser cada vez más complicado, debido que la ventana de correlación en la imagen después del movimiento que verdaderamente coincide con la ventana de correlación en la imagen antes del movimiento, va a estar rotada y cada vez más lejos de su posición original. Así mismo, se puede inferir que la sensibilidad se mantiene prácticamente constante para los casos de un mismo grupo, lo cual quiere decir que el proceso siempre es capaz de considerar como correctas el mismo porcentaje de todas las correspondencias verdaderas sin importar si hubo rotación, traslación vertical, traslación horizontal o cualquier combinación de estos. Dicho resultado es inesperado y bastante bueno, pero no tan relevante por lo comentado en el párrafo anterior, ya que para que la sensibilidad se mantenga constante la cantidad de falsos negativos tiene que disminuir en la misma proporción con la que disminuye la cantidad de verdaderos positivos (correspondencias establecidas consideradas verdaderamente como correctas) de un caso a otro, mientras que por otro lado, el hecho de que la precisión disminuya implica que la cantidad de falsos positivos aumenta conforme disminuye la cantidad de verdaderos positivos, lo que se hace cada vez más amplia la brecha entre falsos positivos y negativos.

El ultimo resultado que se puede deducir de la tabla 5.1 a la tabla 5.5 es que la precisión de los casos en los que la traslación solamente fue horizontal, es menor a la obtenida en los casos en los que traslación solamente fue vertical, sin importar si hubo rotación o no. Específicamente, alrededor de un 1% menor, lo cual quiere decir que el proceso es mejor estableciendo correspondencias verdaderamente correctas en aquellas situaciones donde la traslación es únicamente horizontal en lugar de vertical. En realidad, lo que se espera es que la precisión sea igual en ambas situaciones, por lo que es un resultado

inesperado pero que tiene una explicación. Lo que pasa es que el límite que se utilizó para las dos componentes de la traslación fue exactamente el mismo, de 120 píxeles, que de cierta forma es como considerar que las imágenes son cuadradas, es decir, que un píxel tiene la misma libertad de movimiento tanto de forma horizontal como vertical, lo cual es falso. Las imágenes usualmente son rectangulares, lo cual quiere decir que un píxel tiene más libertad de movimiento en la horizontal que en la vertical y por lo tanto, el límite de la traslación vertical debería de ser diferente y menor al límite de la traslación horizontal, pero en cambio, se utilizó el mismo límite, lo que generó que la precisión con la traslación vertical fuera menor.

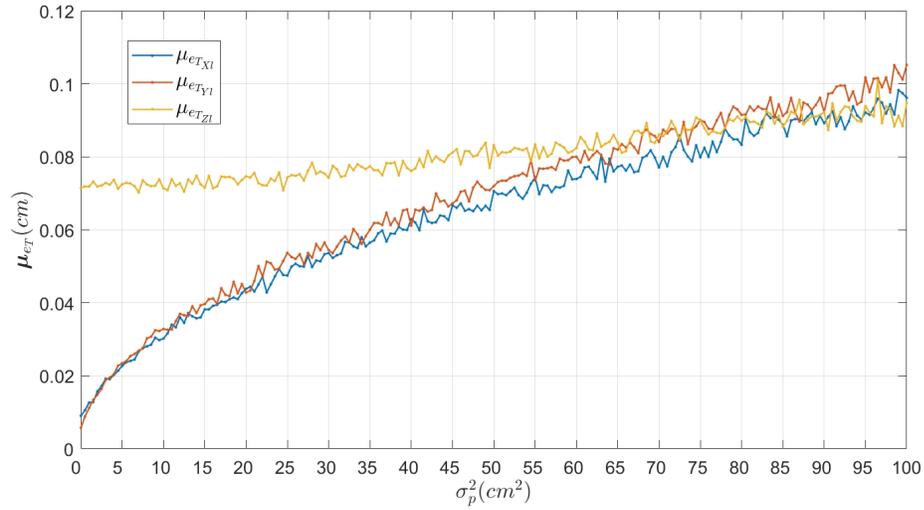
5.2. Resultados de la prueba II

Los resultados de la segunda prueba se resumen desde la figura 5.1 hasta la figura 5.3, en las cuales se muestran las gráficas de las respectivas medias y varianzas del error absoluto de traslación (e_T), del error absoluto de rotación (e_W) y del error de coincidencia entre nubes de puntos (e_c), contra la varianza (σ_p^2) del error de posición aleatorio gaussiano (e_{pm}) que se introdujo a las posiciones tridimensionales de cada uno de los puntos que conforman la nube de puntos después del movimiento y que se varió de 0 a 100 centímetros cuadrados en pasos de 0.5.

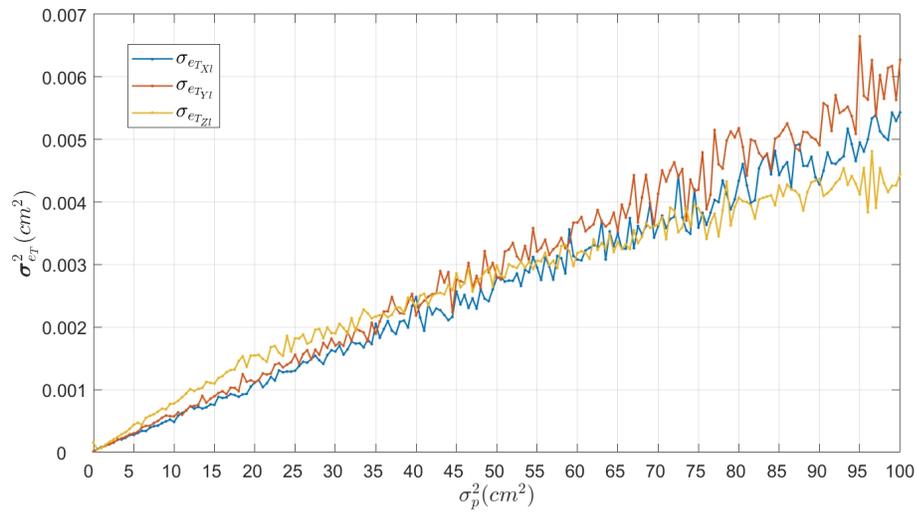
Cabe destacar que para facilitar el análisis de los resultados y su comparación, estos se agruparon de acuerdo al tipo de error, del tal forma que en la figura 5.1 se muestran los resultados relacionados con el error absoluto de traslación que conforman el primer grupo, en la figura 5.2 los del error absoluto de rotación que conforman el segundo grupo y en la figura 5.3 los del error de coincidencia que conforman el tercer y último grupo. Además, dichas figuras están compuestas por dos subfiguras: la subfigura (a) en la cual se muestra la gráfica de la media del respectivo error y la subfigura (b), en la cual se muestra la gráfica de la varianza del respectivo error. Cabe aclarar que como el error absoluto de traslación es un vector tridimensional, es decir, como tiene componentes en X_l , en Y_l y en Z_l , su media y su varianza también y es por ello, que las gráficas de dichas métricas en realidad están conformadas por tres gráficas de diferentes colores que representan las tres componentes respectivas en X_l , en Y_l y en Z_l de la media o varianza. Es de forma totalmente análoga con la media y la varianza del error de rotación.

Es importante recordar que el límite para las tres componentes de la traslación (l_T) fue de 18 centímetros, el cual es equivalente al límite de la traslación bidimensional de 120 píxeles que se utilizó en la primera prueba, así mismo, el límite de las tres componentes de la rotación (l_W) fue de 10 grados, que es el límite de rotación máximo permitido para que el proceso de establecimien-

to de correspondencias tengan un buen funcionamiento, que se dedujo de los resultados de la primera prueba, tal y como se explicó en la sección anterior.

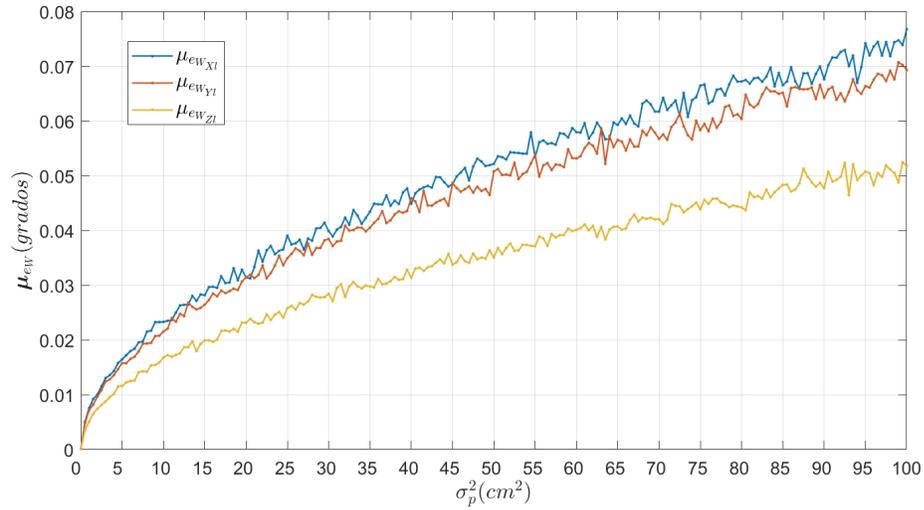


(a) Media del error de traslación en X_l (en azul), media del error de traslación en Y_l (en anaranjado) y media del error de traslación en Z_l (en amarillo).

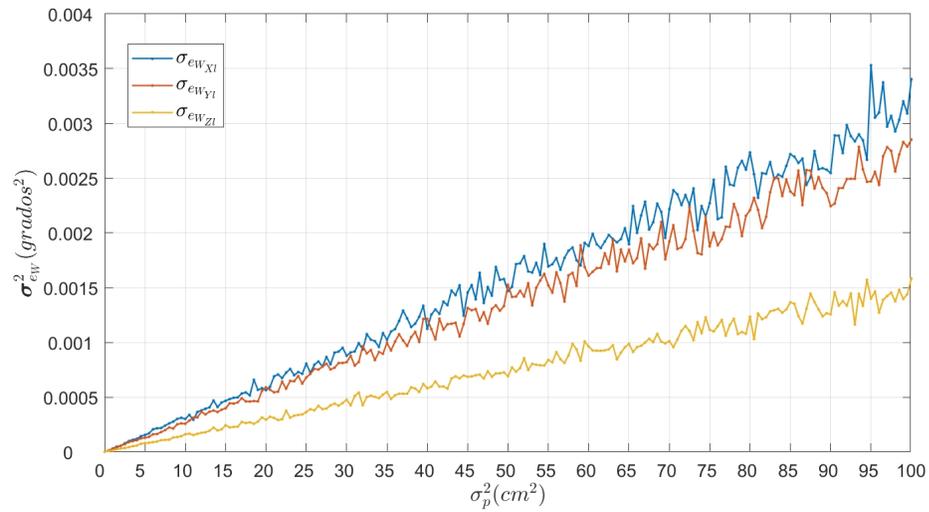


(b) Varianza del error de traslación en X_l (en azul), varianza del error de traslación en Y_l (en anaranjado) y varianza del error de traslación en Z_l (en amarillo).

Figura 5.1: Gráficas de la media y la varianza del error de traslación, contra la varianza del error de posición introducido a la segunda nube de puntos en la segunda prueba.

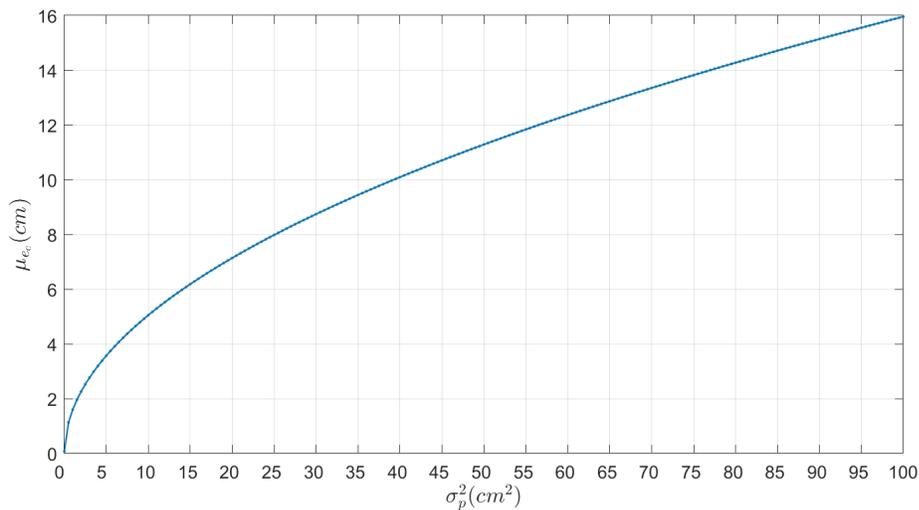


(a) Media del error de rotación en X_l (en azul), media del error de rotación en Y_l (en anaranjado) y media del error de rotación en Z_l (en amarillo).

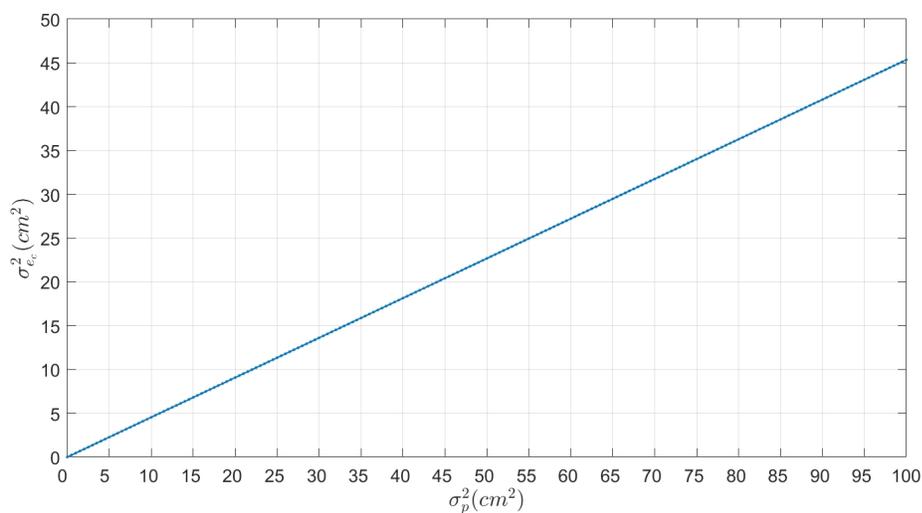


(b) Varianza del error de rotación en X_l (en azul), varianza del error de rotación en Y_l (en anaranjado) y varianza del error de rotación en Z_l (en amarillo).

Figura 5.2: Gráficas de la media y la varianza del error de rotación, contra la varianza del error de posición introducido a la segunda nube de puntos en la segunda prueba.



(a) Media del error de coincidencia.



(b) Varianza del error de coincidencia.

Figura 5.3: Gráficas de la media y la varianza del error de coincidencia entre las nubes de puntos, contra la varianza del error de posición introducido a la segunda nube de puntos en la segunda prueba.

A continuación se analiza y se compara cada grupo de resultados, es decir, las gráficas que se muestran en las figuras 5.1, 5.2 y 5.3, que son los resultados asociados a cada uno de los tres errores calculados en la segunda prueba. Así mismo, se realiza un análisis de los resultados generales que se pueden inferir de las gráficas de los tres grupos, es decir, de aquellos comportamientos

implícitos o explícitos que se repiten en las gráficas de cada error.

En la figura 5.1 se muestra el primer grupo de resultados conformado por aquellas gráficas que están asociadas con el error absoluto de traslación, específicamente, en la subfigura 5.1a se muestran las gráficas de la media del error de traslación en X_l , Y_l y Z_l , mientras que en la subfigura 5.1b se muestran las gráficas de la varianza del error de traslación en X_l , Y_l y Z_l . En la subfigura 5.1a se puede apreciar que la media del error de traslación en X_l , Y_l y Z_l es demasiado pequeña y prácticamente insignificante, debido a que incluso en el peor de los casos, que se da cuando la varianza del error de posición es igual a 100 centímetros cuadrados, el valor máximo de la media es de aproximadamente 0.105 centímetros o 1.05 milímetros en Y_l y es aún menor en X_l y Z_l , lo cual implica que para un mismo valor de la varianza del error de posición, el valor central de los diferentes valores del error de traslación es prácticamente igual a cero. Mientras que la subfigura 5.1b se puede notar que la varianza del error de traslación en X_l , Y_l y Z_l es aún más pequeña que la media, ya que en el peor de los casos, cuando la varianza del error de posición es igual a 100 centímetros cuadrados, el valor máximo de la varianza es de aproximadamente 0.006 centímetros o 0.06 milímetros en Y_l , siendo menor en X_l y Z_l , lo cual implica que para un mismo valor de la varianza del error de posición, la dispersión de los diferentes valores del error de traslación con respecto a la media es demasiado pequeña, es decir, que los valores del error de traslación están demasiado cerca de la media, siendo estos prácticamente iguales. Los resultados descritos anteriormente son excelentes y demuestran que si se las correspondencias temporales 3D son conocidas y correctas, el proceso de estimación de movimiento por descomposición en valores singulares es capaz de estimar el vector de traslación con una exactitud y una precisión demasiado alta, a pesar del error de posición presente en la nube de puntos después del movimiento, ya que la traslación estimada es prácticamente igual a la traslación real, siendo la diferencia entre estas de $1,05 + \sqrt{0,06}$ milímetros en el peor escenario posible.

En la figura 5.2 se muestra el segundo grupo de resultados conformado por aquellas gráficas que están relacionadas con el error absoluto de rotación, específicamente, en la subfigura 5.2a se muestran las gráficas de la media del error de rotación en X_l , Y_l y Z_l , mientras que en la subfigura 5.2b se muestran las gráficas de su varianza en X_l , Y_l y Z_l . En la subfigura 5.2a se puede percibir que la media del error de rotación en X_l , Y_l y Z_l es inclusive más pequeña que la media del error de traslación, de tal forma que en el peor de los casos, el valor máximo de la media es de aproximadamente 0.075 grados en X_l , siendo aún menor para la media en Y_l y en Z_l , lo que significa que el valor central de los diferentes valores del error de rotación para un mismo valor de la varianza del error de posición, es por poco igual cero. Así mismo, en la subfigura 5.2b se observa que la varianza del error de rotación en X_l , Y_l y Z_l es más pequeña que

su media y aun más pequeña que la varianza del error de traslación, debido a que en el peor de los casos, el valor máximo de la varianza es aproximadamente igual a 0.0035 grados en X_l e inferior en Y_l y en Z_l , lo que significa que para un mismo valor de la varianza del error de posición, los diferentes valores del error de rotación son demasiado parecidos a la media, es decir, la dispersión de estos es demasiado pequeña. Al igual que con el error de traslación, los resultados descritos anteriormente para el error de rotación son excelentes y en este caso demuestran que si se conocen las correspondencias temporales 3D, el proceso de estimación de movimiento por descomposición en valores singulares es capaz de estimar el vector de rotación y por lo tanto, la matriz de rotación, con una exactitud y una precisión aún más alta que con la que estima el vector de traslación, esto a pesar del error de posición que afecta la nube de puntos después del movimiento, ya que la rotación estimada es por poco igual a la rotación real, siendo la diferencia entre estas de $0,075 + \sqrt{0,0035}$ grados en el peor escenario posible, que es casi la mitad de la diferencia máxima entre la traslación estimada y real.

Además, si se compara la subfigura 5.1a con la subfigura 5.2a, se puede notar que tanto la media en X_l , Y_l y Z_l del error de traslación, como la media en X_l , Y_l y Z_l del error de rotación, crecen con un comportamiento no lineal, parecido al de una raíz cuadrada, esto conforme aumenta el valor de la varianza del error de posición. Dicho comportamiento es esperado y tiene una explicación bastante simple. Lo que pasa es que el error de posición depende directamente de la desviación estándar, entonces, conforme se aumenta el valor de la varianza, el valor de la desviación estándar también aumenta pero de forma no lineal debido a que se obtiene a partir de la raíz cuadrada de la varianza y por lo tanto, el error de posición también aumenta de forma no lineal. Luego, como el proceso de estimación de movimiento depende directamente de la nube de puntos después del movimiento, a la cual se le adiciona el error de posición, es de esperarse que el comportamiento no lineal de dicho error también aparezca en la traslación y la rotación estimada, de tal forma que conforme aumenta la varianza del error de posición, la traslación y la rotación estimada se alejan cada vez más de la traslación y la rotación real pero de forma no lineal, siguiendo el comportamiento de una raíz cuadrada, lo cual quiere decir que la diferencia entre la rotación y la traslación real y estimada, es decir, el error de traslación y de rotación, va creciendo de forma no lineal y por lo tanto, sus medias igual. Entonces, de forma resumida, lo que sucede es que el error de traslación y el error rotación dependen indirectamente del error de posición, de tal forma que adoptan el mismo comportamiento no lineal que presenta el error de posición cuando se aumenta la varianza y por lo tanto, las medias de estos también. Así mismo, si se compara la subfigura 5.1b con la subfigura 5.2b, se puede notar que tanto la varianza en X_l , Y_l y Z_l del error de traslación, como la varianza en X_l , Y_l y Z_l del error de rotación, crecen

con un comportamiento lineal conforme se aumenta el valor de la varianza del error de posición, el cual también es un comportamiento esperado que tiene una simple explicación. En este caso, lo que sucede es que como el error de traslación y el error de rotación dependen indirectamente del error de posición, tal y como se explicó anteriormente, sus varianzas también lo hacen, de tal forma que adoptan el mismo comportamiento lineal que tiene la varianza del error de posición, ya que esta se aumenta linealmente en pasos de 0.5.

Adicionalmente, en las figuras 5.1a y 5.2a, se puede apreciar que la gráfica de la media y de la varianza del error de traslación y de rotación en Z_l , difiere ligeramente de las respectivas gráficas de la media y la varianza del error de traslación y de rotación en X_l y Y_l . En realidad, lo que se espera es que las gráficas en X_l , Y_l y Z_l sean muy parecidas y que la diferencia entre las tres sea muy pequeña, por lo que es un resultado inesperado, pero que tiene una explicación que surge de la única diferencia que existe entre las posiciones respecto a X_l y Y_l y la posiciones respecto a Z_l , de los puntos que conforman las nubes de puntos. Lo que sucede es que los puntos que conforman la nube de puntos antes del movimiento y por lo tanto, la nube de puntos después del movimiento, se encuentra sobre un mismo plano, es decir, las posiciones respecto a Z_l son muy parecidas o prácticamente iguales, esto es debido a que la nube de puntos de prueba que es la nube antes del movimiento, se obtuvo a partir de las imágenes capturadas por una cámara estereoscópica que se ubicó de tal forma que su plano fuera paralelo a la superficie, es decir, el eje Z_l forma un ángulo de 90 grados con la superficie y por lo tanto, todos los puntos de dicha superficie que están dentro del campo de visión de la cámara, se encuentra a la misma distancia respecto al eje Z_l . Entonces, lo que pasa es que las posiciones respecto a X_l y Y_l de los puntos de las nubes varían considerablemente, mientras que las posiciones respecto a Z_l son muy parecidas, hasta el punto que se podrían considerar como constantes y es esa diferencia entre los comportamientos de las posiciones de las nubes lo que ocasiona indirectamente la ligera diferencia que existe entre la gráfica de la media y la varianza en Z_l y las gráficas de las respectivas medias y varianzas en X_l y Y_l .

Otro comportamiento bastante obvio que se aprecia en las figuras 5.1a y 5.2a, es que las gráficas de las medias y las varianzas del error de traslación y de rotación están compuestas por un montón de altos y bajos de principio a fin y que van siendo cada vez más notables conforme va aumentando el valor de la varianza del error de posición. Un alto se da cuando el valor de la media o la varianza del error de traslación o rotación para un valor dado de la varianza σ_p^2 del error de posición, es mayor que el valor de la respectiva media o varianza para $\sigma_p^2 - 0,5$ y para $\sigma_p^2 + 0,5$, que son los valores inmediatamente a la izquierda y la derecha. Análogamente, un bajo se da cuando el valor de la media o la varianza para un valor dado de la varianza σ_p^2 del error de posición,

es menor que el valor de la respectiva media o varianza para $\sigma_p^2 - 0,5$ y para $\sigma_p^2 + 0,5$. Dichos altos y bajos surgen principalmente debido a que la traslación y rotación se eligen de forma totalmente aleatoria y entonces, hay ocasiones en las que para un σ_p^2 los parámetros de movimiento van a estar muy cerca de sus respectivos límites, en comparación con $\sigma_p^2 - 0,5$ y $\sigma_p^2 + 0,5$, y por lo tanto, se espera que los errores de traslación y rotación sea ligeramente mayores, lo que genera un alto. Análogamente, hay otras ocasiones en las que para un σ_p^2 los parámetros de movimiento van a estar lejos de sus respectivos límites, en comparación con $\sigma_p^2 - 0,5$ y $\sigma_p^2 + 0,5$, y por lo tanto, se espera que los errores de traslación y rotación sea ligeramente menores, lo que genera un bajo. Cabe destacar que en realidad los altos y los bajos son insignificantes, específicamente, del orden de 10^{-3} para las medias y del orden de 10^{-3} para las varianzas, de tal forma que si el rango del eje vertical se aumentara de 0 a 1, por ejemplo, esos altos y bajos prácticamente desaparecerían. Además, es importante mencionar que si en lugar de una sola prueba, se realizaran varias pruebas pero para diferentes intervalos pequeños dentro de los cuales se puede elegir aleatoriamente la traslación y rotación, los altos y bajos desaparecerían para un rango pequeño del eje vertical.

Por otro lado, en la figura 5.3 se muestra el tercer y último grupo de resultados conformado por aquellas gráficas que están relacionadas con el error de coincidencia, específicamente, en la subfigura 5.3a se muestra la gráfica de su media y en la subfigura 5.3b la gráfica de su varianza. En la subfigura 5.3a se puede notar que la media del error de coincidencia es considerablemente grande en comparación con la media del error de traslación o de rotación, ya que en el peor de los casos el valor máximo es de 16 centímetros y a diferencia de los otros errores, no se puede decir que para un mismo valor de la varianza del error de posición, el valor central de los diferentes valores del error de coincidencia es aproximadamente cero, sino que ahora más bien dicho valor central que es la media, va creciendo considerablemente desde 0 hasta 16 centímetros con un comportamiento no lineal parecido al de una raíz cuadrada, esto conforme aumenta la varianza del error de posición. Dicho comportamiento no lineal es esperado ya que el error de coincidencia se calcula a partir de la nube de puntos antes del movimiento y de la nube de puntos después del movimiento, a la que siempre se le adiciona el error de posición, el cual se aumenta de forma no lineal, tal y como se explicó anteriormente. Entonces, se podría decir que el error de coincidencia depende directamente del error de posición. Por otra parte, en la subfigura 5.3b se observa que la varianza del error de coincidencia no es igual de pequeña que la varianza del error de traslación o de rotación, en cambio, es bastante grande, llegando hasta un valor máximo de aproximadamente 46 centímetros cuadrados en el peor caso posible. Además, en comparación con los otros errores, no se puede afirmar que la dispersión de los diferentes valores del error de coincidencia con

respecto a la media es demasiado pequeña, en cambio, la dispersión que es la varianza crece considerablemente desde 0 hasta 46 centímetros cuadrados y lo hace de manera lineal conforme aumenta la varianza del error de posición de 0 a 100. Dicho comportamiento lineal también es de esperarse, debido a que como el error de coincidencia depende directamente del error de posición, la varianza de este también depende de la varianza del error de posición, la cual varía linealmente, como se explicó en párrafos anteriores.

Los resultados anteriores demuestran que a pesar de que el proceso de estimación de movimiento basado en la descomposición en valores singulares es capaz de estimar los parámetros de movimiento tridimensionales, la traslación y la rotación en los tres ejes, con un error que tiende a cero para cualquier valor de la varianza del error de posición gaussiano de media cero, siempre y cuando las correspondencias temporales 3D sean correctas y conocidas, el error de coincidencia entre las nubes de puntos antes y después del movimiento no tiende a cero para cualquier valor de la varianza, en cambio, este va aumentando considerablemente conforme incrementa la varianza del error de posición, así como la dispersión de sus valores. El punto es que como la traslación y la rotación estimada son tan precisas y exactas, prácticamente iguales a la traslación y rotación real, lo lógico sería pensar que las nubes de puntos deberían de coincidir casi que de forma perfecta, pero resulta que eso solamente sucedería si las nubes de puntos no se ven perturbadas por ningún tipo de error, la cual es una situación totalmente irreal e ideal que en este caso solamente se cumple cuando la varianza del error de posición es cero, pero para el resto de los casos siempre va a haber un error de posición presente en la segunda nube de puntos que no permite esta coincidencia de forma exacta con la primera nube de puntos, incluso cuando los parámetros de movimiento son igual a los reales. Entonces, el hecho de que el proceso estime los parámetros de movimiento con un error cercano a cero, no quiere decir que se elimine el error de posición presente en la segunda nube de puntos y por lo tanto, el error de coincidencia tampoco, más bien, quiere decir que el proceso tiene la capacidad de realizar una estimación casi perfecta de los parámetros de movimiento a pesar del error de posición que afecta a la segunda nube de puntos y a pesar de que dicho error no permite que haya una coincidencia perfecta entre las nubes de puntos para cualesquiera parámetros de movimiento. Cabe destacar que lo descrito anteriormente es de esperarse debido a que la estimación se realiza a partir de la minimización de un error cuadrático medio y resulta que dicha estimación tiene una propiedad bastante particular. Dicha propiedad es que cuando el error de posición que afecta a las nubes de puntos después del movimiento es un error gaussiano de media cero y varianza constante para todas las posiciones de las nubes de puntos, la estimación es no sesgada, lo cual quiere decir que si la cantidad de correspondencias temporales 3D tiende a un valor muy grande, entonces se espera que el error de estimación tienda a cero,

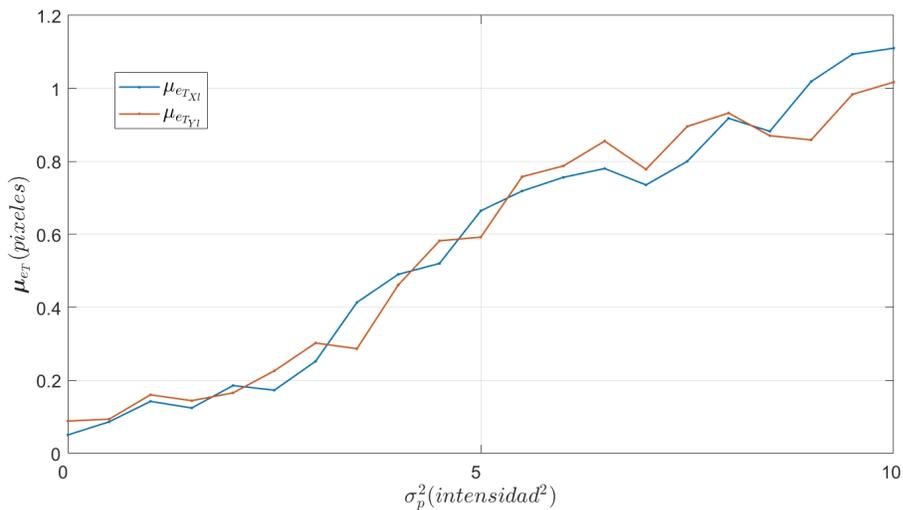
es decir, que los parámetros de movimiento estimados sean lo más cercano a los parámetros de movimiento real, que es justo lo que sucedió en esta prueba. Obviamente, lo anterior está sujeto a que las correspondencias temporales 3D sean correctas. Entonces, los resultados obtenidos en esta prueba demuestran que el proceso de estimación de movimiento tiene un funcionamiento correcto y esperado, ya que cumplen con lo que dicta la teoría.

5.3. Resultados de la prueba III

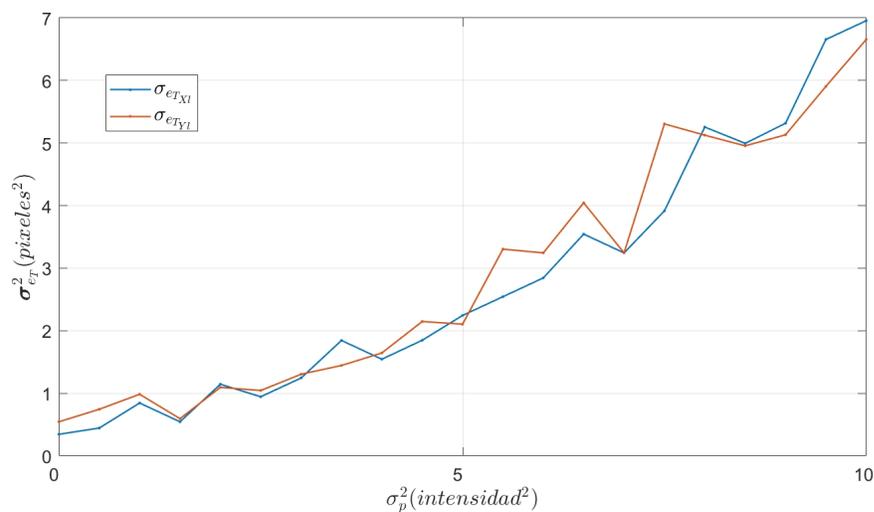
Los resultados de la tercera prueba se resumen de la figura 5.4 a la figura 5.6, en donde se muestran las gráficas de las respectivas medias y varianzas del error absoluto de traslación en X_l y Y_l ($e_{T_{X_l}}$, $e_{T_{Y_l}}$), del error absoluto de rotación en Z_l ($e_{W_{Z_l}}$) y del error de coincidencia entre las nubes de puntos (e_c), contra la varianza (σ_r^2) del ruido gaussiano (r_n) que se adicione a los valores de intensidad de los píxeles de la imagen izquierda después del movimiento y que va de 0 a 100 de intensidad al cuadrado. Es importante aclarar que en esta prueba, a diferencia de la segunda prueba, lo que se varió fue la desviación estándar (σ_r) del ruido de 0 a 10 de intensidad en pasos de 0.5 y por lo tanto, la varianza del ruido se varió indirectamente de 0 a 100 pero en pasos no lineales, ya que esta es la raíz cuadrada de la desviación.

Al igual que con los resultados de la segunda prueba, los resultados de esta tercera prueba se agruparon de acuerdo al tipo de error para facilitar su análisis y comparación. Entonces, en la figura 5.4 se muestran los resultados relacionados con el error absoluto de traslación en X_l y Y_l que componen el primer grupo, en la figura 5.2 los del error absoluto de rotación en Z_l que componen el segundo grupo y en la figura 5.3 los del error de coincidencia que integran el tercer y último grupo. Además, cada figura se divide en la subfigura (a) en la que se muestra la gráfica de la media del respectivo error y en la subfigura (b) en la que se muestra la gráfica de la varianza. Cabe aclarar que como en esta tercera prueba se utilizaron nubes de puntos bidimensionales y por lo tanto, parámetros de movimiento bidimensionales, el error absoluto de traslación es entonces un vector bidimensional con componentes en X_l y Y_l y su media y varianza igual y es por esa razón que las gráficas de dichas métricas están conformadas por dos gráficas de diferente color que representan sus componentes en X_l y Y_l . Mientras que el error de rotación es unidimensional, solamente tiene una componente en Z_l , y por lo tanto, su media y varianza también, es por ello que solo se muestra una gráfica por métrica.

Es importante recordar que en esta prueba el límite de la traslación bidimensional (l_t) fue de 120 píxeles, que equivale a 18 centímetros si las nubes fueran tridimensionales y además, el límite de la rotación unidimensional (l_θ) fue de 10 grados, al igual que en la segunda prueba.



(a) Media del error de traslación en X_l (en azul) y media del error de traslación en Y_l (en anaranjado).



(b) Varianza del error de traslación en X_l (en azul) y varianza del error de traslación en Y_l (en anaranjado).

Figura 5.4: Gráficas de la media y la varianza del error de traslación, contra la varianza del ruido introducido a los valores de intensidad de la segunda imagen izquierda en la tercera prueba.

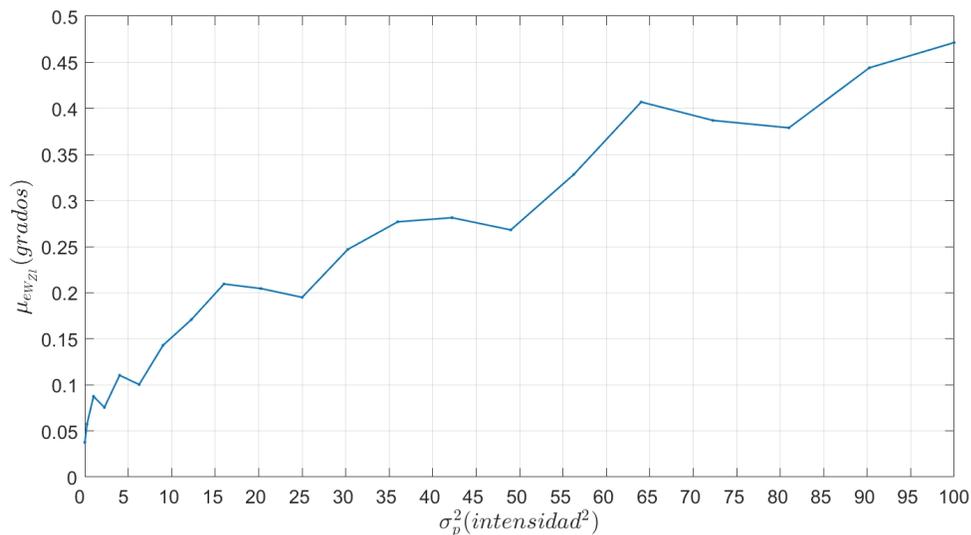
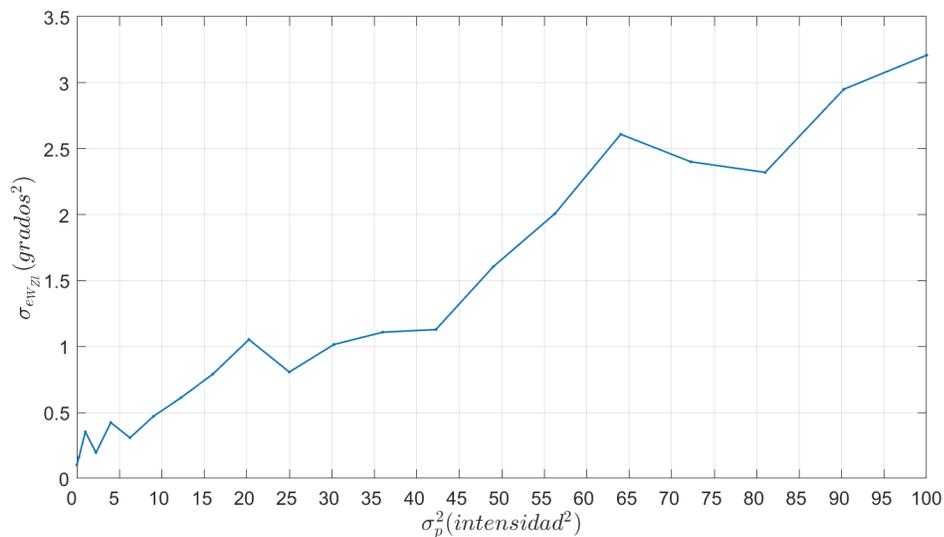
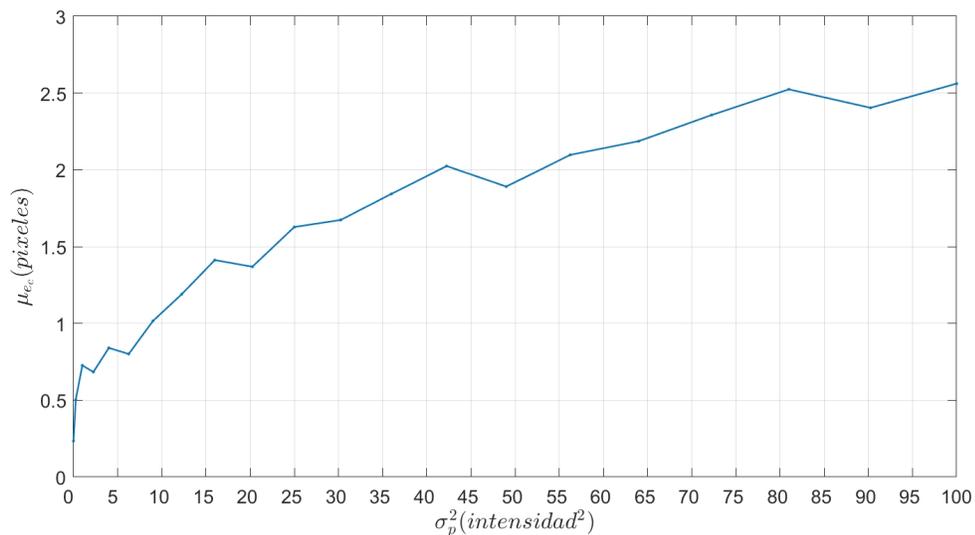
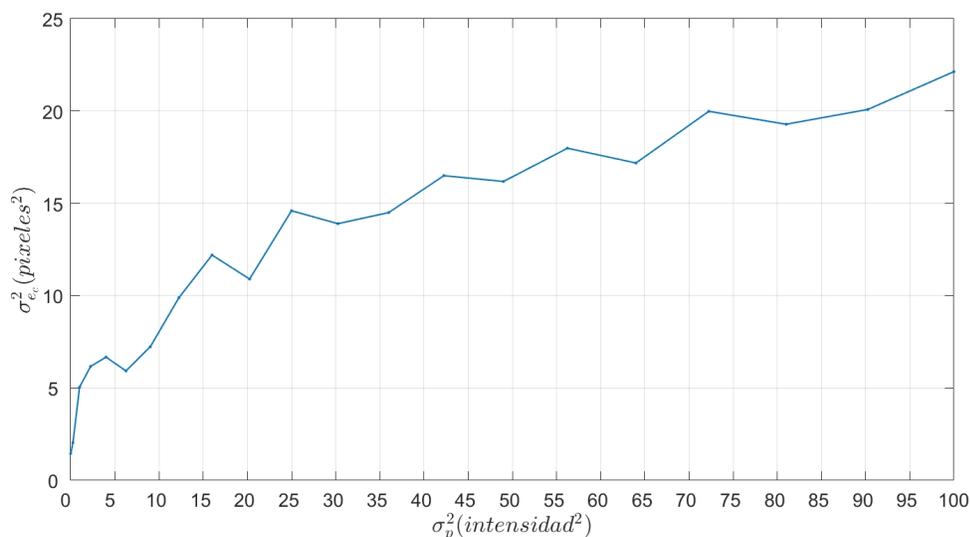
(a) Media del error de rotación en Z_l .(b) Varianza del error de rotación en Z_l .

Figura 5.5: Gráficas de la media y la varianza del error de rotación, contra la varianza del ruido introducido a los valores de intensidad de la segunda imagen izquierda en la tercera prueba.



(a) Media del error de coincidencia.



(b) Varianza del error de coincidencia.

Figura 5.6: Gráficas de la media y la varianza del error de coincidencia entre las nubes de puntos, contra la varianza del ruido introducido a los valores de intensidad de la segunda imagen izquierda en la tercera prueba.

A continuación se analiza y se compara cada grupo de resultados, que son las gráficas asociadas a cada uno de los tres errores y que se muestran en las figuras 5.4, 5.5 y 5.6. Además, se analizan los resultados generales que se pueden inferir de todas las gráficas de los tres grupos.

En la figura 5.4 se presenta el primer grupo de resultados compuesto por las gráficas de la media del error de traslación en X_l y Y_l , que se muestran en la subfigura 5.4a y las gráficas de la varianza del error de traslación, que se muestran en la subfigura 5.4b. En la subfigura 5.4a se puede notar que la media del error de traslación en X_l y en Y_l es relativamente pequeña, pero no se podría decir que es aproximadamente cero, ya que esta crece desde 0 píxeles hasta 1.1 píxeles, conforme aumenta la varianza del ruido gaussiano de intensidad, lo cual es equivalente a decir que el valor central de los diferentes valores del error de traslación va creciendo conforme crece la varianza del ruido, de tal forma que se va alejando cada vez del valor central ideal que sería de 0 píxeles y lo va haciendo con un comportamiento más o menos no lineal. En cuanto a la subfigura 5.4b, se observa que la varianza en X_l y en Y_l es mucho más grande que la respectiva media, específicamente, alrededor de unas 6 o 7 veces el valor de la media, debido a que la varianza crece desde aproximadamente 0 píxeles hasta 7 píxeles, conforme aumente la varianza del ruido de intensidad, lo que significa que la dispersión de los diferentes valores del error de traslación respecto a la media, va creciendo conforme crece la varianza del ruido, tal que los valores se van alejando cada vez más de la respectiva media y lo hacen de una forma aproximadamente lineal. Los resultados anteriores demuestran que la precisión y la exactitud con la que algoritmo de odometría visual estereoscópico implementado estima la traslación, van disminuyendo entre mayor sea el ruido presente en los valores de intensidad de la imagen izquierda después del movimiento, de tal forma que la traslación estimada se aleja cada vez de la traslación real, siendo la diferencia entre estas de aproximadamente $1,1 \pm \sqrt{7}$ píxeles en el peor caso.

En la figura 5.5 se presentan las gráficas asociadas al error de rotación en Z_l , específicamente, en la subfigura 5.5a se muestra la gráfica de su media y en la subfigura 5.5b la gráfica de su varianza. En la subfigura 5.5a se observa que la media del error de rotación en Z_l es más pequeña que la media del error de traslación en X_l y en Y_l , específicamente, alrededor de un 50 % menor, pero aun así no es posible afirmar que la media de la rotación es prácticamente cero, ya que esta crece desde 0 hasta 0.47 grados aproximadamente, conforme aumenta la varianza del ruido gaussiano de intensidad, es decir, el valor central de los diferentes valores del error de rotación se va alejando cada vez del valor ideal de 0 de forma más o menos lineal, de acuerdo al crecimiento de la varianza del ruido, pero no tanto como en el caso del error de traslación. A su vez, en la subfigura 5.5b se aprecia que la varianza del error de rotación en Z_l es alrededor de 6 o 7 veces mayor que la respectiva media, al igual que la varianza del error de traslación. Además, se observa que la varianza crece desde 0 hasta 3.3 grados cuadrados aproximadamente, conforme aumenta la varianza del ruido, es decir, la dispersión de los diferentes valores del error de rotación respecto a la media va aumentando conforme crece la varianza del

ruido, lo que significa que se alejan cada vez más de la respectiva media con un comportamiento aproximadamente lineal, pero ese alejamiento es menor en comparación con el error de traslación. Estos resultados demuestran que el algoritmo de odometría visual estereoscópico implementado estima la rotación con una mayor precisión y exactitud que la traslación, pero al igual que con la traslación, la precisión y exactitud disminuyen entre mayor sea el ruido presente en la imagen de intensidad izquierda después del movimiento, tal que la diferencia entre la rotación estimada y la real es cada vez mayor, siendo el valor de esta de aproximadamente $0,47 \pm \sqrt{3,3}$ grados en el peor caso.

En la figura 5.6 se presentan las gráficas asociadas al error de coincidencia, específicamente, en la subfigura 5.6a se muestra la gráfica de su media y en la subfigura 5.5b la gráfica de su varianza. En la subfigura 5.6a se puede notar que la media del error de coincidencia es más grande que la media del error de traslación y la media del error de rotación, específicamente, un alrededor el doble de la media de traslación y unas 5 veces más grande que la media de rotación, ya que dicha media crece con un comportamiento más o menos no lineal desde 0 hasta 2.5 píxeles aproximadamente, conforme aumenta la varianza del ruido de intensidad, lo cual quiere decir que el valor central de los diferentes valores del error de coincidencia se distancia cada vez más del valor ideal de 0 conforme aumenta la varianza del ruido y además, dicho distanciamiento es mayor en comparación con el error de traslación y rotación. Por otra parte, en la subfigura 5.6b se observa que la varianza del error de coincidencia también es mayor que la varianza del error de traslación y rotación, específicamente, cerca de 3 veces más grande que la varianza de traslación y 6 veces más grande que la varianza de rotación, ya que dicha varianza va aumentando de forma más o menos no lineal desde 0 hasta 22 píxeles cuadrados aproximadamente, conforme aumenta la varianza del ruido, es decir, la dispersión de los diferentes valores del error de coincidencia respecto a la media va creciendo de forma más o menos no lineal, de tal forma que dichos valores se distancia cada vez más de la respectiva media, mucho más en comparación con el error de traslación y rotación. Dichos resultados demuestran que la disminución de la precisión y exactitud en la estimación de los parámetros de movimiento por parte del algoritmo de odometría visual estereoscópico, conforme aumenta el ruido presente en la imagen de intensidad izquierda después del movimiento, ocasiona que la coincidencia entre las nubes de puntos sea cada vez menos precisa y exacta, es decir, que las distancias euclidianas entre los puntos de las nubes de puntos sea cada vez mayor y se dispersen cada vez más, de tal forma que en el peor caso la distancia euclidiana entre varios puntos es de aproximadamente $2,5 \pm \sqrt{22}$ píxeles.

Ahora surge la siguiente pregunta: ¿son los resultados anteriores buenos o malos? Para responder dicha pregunta primero hay que analizar la razón de porque la precisión y exactitud de la estimación de los parámetros de mo-

vimiento baja conforme aumenta el ruido en los valores de intensidad de la segunda imagen y por lo tanto, precisión y exactitud con la que coinciden las nubes de puntos también. Lo que sucede es que conforme crece el ruido que afecta a la imagen después del movimiento, aumenta la cantidad de correspondencias incorrectas establecidas por el proceso de establecimiento de correspondencias, debido a que cada vez es más complicado establecer una correspondencia correcta, esto es porque la imagen después del movimiento difiere cada vez más de la imagen antes del movimiento y por lo tanto, el rastreo en la imagen después del movimiento, de la ventana de correlación que se establece en la imagen antes del movimiento, es cada vez más difícil, ya que la ventana de correlación en la imagen después del movimiento que verdaderamente coincide con la ventana de correlación en la imagen antes del movimiento, cada vez se ve más perturbada por el ruido. Luego, el proceso de estimación de movimiento estima la traslación y la rotación a partir de un grupo de correspondencias que cada vez tiene más correspondencias incorrectas que correctas, lo que va causando una disminución de la precisión y la exactitud. Pero luego sigue el proceso basado en RANSAC, el cual debería de ser capaz de encontrar un grupo de correspondencias de las cuales todas sean correctas, para así lograr una estimación más precisa y exacta de los parámetros de movimiento a partir de dicho grupo, pero con los resultados obtenidos se puede apreciar claramente que a dicho proceso le es cada vez más difícil excluir todas las correspondencias incorrectas o encontrar un grupo de correspondencias correctas y por lo tanto, termina estimando el movimiento con un grupo de correspondencias mixtas, el que tenga la cantidad mínima de correspondencias incorrectas, lo cual provoca una disminución paulatina de la precisión y exactitud de los parámetros de movimiento. Además, dicha disminución no solo depende de la cantidad de correspondencias incorrectas, también depende de que tan incorrectas sean dichas correspondencias, es decir, de que tan alejadas están de la correspondencia verdaderamente correcta, lo cual aumenta conforme crece el ruido. Luego, la cantidad de correspondencias incorrectas, su alejamiento de las correspondencias correctas y la diferencia entre la traslación y la rotación estimada y real, que van aumentando cada vez más, afectan directamente a la coincidencia entre las nubes de puntos, ya que la distancia euclidiana entre los puntos de las nubes de puntos depende directamente de las correspondencias incorrectas y de la traslación y rotación estimada y por lo tanto, se va perdiendo precisión y exactitud en la coincidencia de las nubes. Entonces, volviendo a la pregunta del inicio, los resultados obtenidos se pueden considerar como aceptablemente buenos pero no excelentes, ya que no están por debajo ni por encima de las expectativas, es decir, son resultados esperados, ya que por lo explicado anteriormente es de esperarse que conforme aumenta el ruido también aumente el error de traslación y de rotación, debido a que cada vez se hace más complicado establecer correspon-

dencias correctas y excluir las correspondencias incorrectas, lo cual afecta la estimación y que por lo tanto, que también aumente el error de coincidencia de forma mayor a como aumenta el error de traslación y rotación, debido a que este no solo depende de las correspondencias incorrectas, también de la diferencia entre parámetros estimados y reales. Además, cabe destacar que se pueden esperar resultados bastante buenos de parte del algoritmo, es decir, una alta precisión y exactitud en la estimación de los parámetros de movimiento y en la coincidencia de las nubes de puntos, cuando la varianza del ruido en la imagen izquierda después del movimiento es igual o menor a 8 de intensidad al cuadrado o de forma equivalente, cuando su desviación estándar es igual o menor a aproximadamente 3 de intensidad, ya que en esos casos la media del error de traslación es menor a 0.2 píxeles, la media del error de rotación es menor a 0.15 grados y la media del error de coincidencia es de 0.7 píxeles. Entonces, si la varianza del ruido se mantiene entre 0 y 8 se puede esperar un buen funcionamiento del algoritmo.

Por otro lado, la razón por la cual las varianzas de los errores son mayores a las respectivas medias, tal y como se mencionó en párrafos anteriores y como se observa en las figuras 5.4, 5.5 y 5.6, se debe principalmente al proceso de establecimiento de correspondencias temporales y al proceso basado en RANSAC. Hay que recordar que el proceso de establecimiento de correspondencias elige aleatoriamente un grupo de puntos característicos en la imagen antes del movimiento y luego rastrea los puntos característicos correspondientes en la imagen después del movimiento para establecer las correspondencias. Esa aleatoriedad ocasiona que para un mismo valor de la varianza del ruido que afecta a los valores de intensidad de la imagen izquierda después del movimiento, el grupo de correspondencias varíe demasiado de una iteración a otra, es decir, que varía mucho la cantidad de correspondencias correctas e incorrectas que conforman dicho grupo y también que tan malas son las correspondencias incorrectas. Específicamente, lo que sucede es que para un punto característico elegido aleatoriamente, la ventana de correlación correcta en la segunda imagen también es aleatoria y como el ruido que afecta a la segunda imagen es aleatorio, puede que los valores de intensidad de los píxeles de la ventana no se vean tan afectados por el ruido y por lo tanto, su rastreo es más fácil, lo que lleva a una correspondencia correcta, pero también puede que si se vean bastante afectados por el ruido y el rastreo va a ser más complicado, lo que lleva a una correspondencia incorrecta. Luego, al proceso de RANSAC van a llegar grupos de correspondencias que varían mucho de una iteración a otra para un mismo valor de la varianza del ruido y por lo tanto, en algunas ocasiones va a ser capaz de excluir todas las correspondencias incorrectas y estima el movimiento con una buena precisión y exactitud, pero en otras ocasiones no va a ser capaz de excluirlas todas, principalmente debido a que el número de veces que se repite el procedimiento de RANSAC no es suficiente para

encontrar un grupo con correspondencias donde todas son correctas y en esos casos estima el movimiento con una precisión y exactitud menor. Lo anterior ocasiona que para un mismo valor de la varianza del ruido, los diferentes valores de los errores de traslación y de rotación se dispersen mucho de la media, de tal forma que algunos valores van a estar cerca de la media pero otros van a estar bastante alejados. Adicionalmente, como la traslación y la rotación se eligen aleatoriamente para las diferentes iteraciones de un mismo valor de la varianza del ruido, hay ocasiones en las que dichos parámetros de movimiento van a estar muy cerca de los respectivos límites pero en otras ocasiones no y por lo tanto, también repercute en el comportamiento de la varianza, ya que es de esperarse que los errores sean altos cuando los parámetros movimiento están cerca del límite y al revés cuando están lejos.

Además, en las figuras 5.4, 5.5 y 5.6, también se puede notar que las gráficas de las medias y varianzas de los tres errores, están conformadas por varios altos y bajos que se van haciendo cada vez más notables conforme aumenta la varianza del ruido. Dichos altos y bajos se dan principalmente por lo comentado anteriormente, solo que en lugar de suceder de una iteración a otra, sucede entre varianzas consecutivas del ruido de intensidad. En resumen, lo que sucede es que para un σ_r^2 puede que la media y la varianza sea menor que para un $\sigma_r^2 - 0,5$ y para un $\sigma_r^2 + 0,5$, lo cual se debe a que las traslaciones y rotaciones fueron menores y la cantidad de correspondencias incorrectas establecidas y excluidas también. Es de forma análoga para un σ_r^2 donde la media y la varianza es mayor que para $\sigma_r^2 - 0,5$ y $\sigma_r^2 + 0,5$.

Cabe destacar que lo descrito en los dos párrafos anteriores se puede solucionar, es decir, se pueden reducir las varianzas de los errores respecto a las medias y también, los altos y bajos de las gráficas, esto si se lleva a cabo un ajuste más fino de los parámetros de control relacionados con el proceso de establecimiento de correspondencias y de RANSAC, por ejemplo, se podría aumentar el número de repeticiones del proceso de RANSAC, para así aumentar la probabilidad de encontrar un grupo de correspondencias correctas con las cuales estimar el movimiento de una forma más precisa y exacta o por lo menos en el peor caso que se encuentre grupo donde la cantidad de correspondencias incorrectas sea mínimo. Así mismo, se podría aumentar la cantidad de correspondencias que se establecen por el respectivo proceso, ya que de esta forma se aumenta la probabilidad de establecer correspondencias correctas y también se reduce el efecto que pueden tener pocas correspondencias incorrectas dentro del grupo de correspondencias, ya que por ejemplo, no es lo mismo estimar el movimiento con 5 correspondencias correctas y 1 incorrecta que con 18 correctas y 2 incorrectas. Adicionalmente, si la prueba se realizara para diferentes intervalos pequeños dentro de los cuales puede variar la traslación y la rotación y si se realizara el ajuste fino de los parámetros anteriores, es muy probable que las gráficas obtenidas no presentan altos y

bajos, que la varianza sea muy parecida a la media o menor.

Por último, como se mencionó en párrafos anteriores y como se aprecia en las figuras 5.4, 5.5 y 5.6, las gráficas de las diferentes medias y varianzas de los tres errores crecen con un comportamiento tanto lineal como no lineal y a diferencia de los resultados de la prueba anterior, esta vez no hay un comportamiento evidente para la media y tampoco un comportamiento evidente para la varianza, es decir, en algunos errores la media es lineal y en otros no lineal y es la misma forma con la varianza. Lo anterior se debe a que como se vio en párrafos anteriores, la estimación de los parámetros de movimiento y por lo tanto, la diferencia entre los estimados y los reales, no depende solamente del ruido de intensidad en la imagen después del movimiento, también depende de muchos otros factores como la cantidad de correspondencias que fueron establecidas incorrectamente, que tanto se alejan dichas correspondencias de la correspondencias verdaderas, si el proceso de RANSAC fue capaz de excluirlas todas o no y entonces, eso genera que el comportamiento de las gráficas conforme aumenta la varianza del ruido sea muy cambiante de una media a otra o de una varianza a otra.

6 Conclusiones y recomendaciones

6.1. Conclusiones

A partir de los resultados obtenidos para las tres pruebas experimentales realizadas a los distintos procesos que conforman al algoritmo de odometría visual estereoscópico implementado, se llega a diferentes conclusiones relacionadas con el funcionamiento, comportamiento, precisión y exactitud de estos, las cuales se mencionan y se explican a continuación.

De los resultados experimentales de la primera prueba, se concluye que cuando el valor de la rotación que experimenta el rover entre dos instantes de tiempo consecutivos, alrededor de un eje perpendicular a la superficie por la cual este se mueve, es igual o menor a 10 grados y cuando los valores de intensidad de los píxeles que conforman la imagen izquierda antes del movimiento y la imagen izquierda después del movimiento, capturadas en ambos instantes por una cámara estereoscópica atada rigidamente al rover, no se ven perturbados por ningún tipo de ruido o error, el proceso de establecimiento de correspondencias temporales basado en la correlación pseudo normalizada propuesta e implementada por Moravec y posteriormente utilizada por Matthies, establece correspondencias con una precisión y una exactitud de más del 80 %, lo cual indica que su funcionamiento es bastante bueno para las condiciones y limitantes indicadas al principio, ya que en el peor de los casos cuando el proceso considera una correspondencia establecida como correcta, está en lo cierto un 80 % o más de las veces y además, considera como correctas un 80 % o más de todas las correspondencias establecidas que verdaderamente lo son. Aquí es importante aclarar que si la rotación del rover es mayor a 10 grados, no implica que el proceso de establecimiento de correspondencias no vaya a funcionar del todo, solamente que la probabilidad de que este establezca una correspondencia de forma errónea va a ser mayor y por lo tanto, sería de esperarse que al proceso basado en RANSAC le cuesta más excluirlas y le tome más tiempo, lo cual puede llegar a perjudicar la estimación de los parámetros de movimiento y el tiempo de ejecución.

También se puede concluir que como es de esperarse, la precisión y la sensibilidad del proceso de establecimiento de correspondencias temporales disminuyen conforme aumenta la rotación que experimenta el rover entre los dos instantes de tiempo consecutivos, de tal forma que su funcionamiento se va degradando cada vez más, es decir, la probabilidad de que este establezca una correspondencia de forma errónea aumenta. Dicho comportamiento es

bastante obvio, ya que cuando hay una rotación presente, el proceso nunca va a poder rastrear correctamente en la imagen después del movimiento, la ventana de correlación que se establece en la imagen antes del movimiento. Lo que sucede es que la ventana en la imagen después del movimiento, que realmente coincide con la ventana de correlación en la imagen antes del movimiento, va a estar rotada respecto al eje horizontal de la respectiva imagen y resulta que el proceso no rastrea ventanas rotadas, es decir, no toma en cuenta esa rotación que puede llegar a experimentar la ventana de correlación de un instante a otro. Entonces, el proceso solamente rastrea aquella ventana en la imagen después del movimiento que no está rotada y que comparte la mayor similitud con la ventana en la imagen antes del movimiento y que por lo tanto, nunca va a ser la ventana verdaderamente correcta, por lo que siempre va a existir un error en el establecimiento de la correspondencia que aumenta conforme aumenta la rotación.

Otra conclusión importante es que el proceso es más propenso a cometer falsos positivos (correspondencias establecidas consideradas falsamente como correctas) que a cometer falsos negativos (correspondencias establecidas consideradas falsamente como incorrectas), debido a que en la mayoría de las situaciones, un 85 % o más de las veces, la precisión del proceso fue menor que la sensibilidad. Es importante aclarar que dicho comportamiento no es el deseado, ya que es mejor que el proceso cometa más falsos negativos que positivos, lo cual quiere decir que es mejor que el proceso cometa más errores a la hora de considerar correspondencias como incorrectas, ya que solamente estaría rechazando correspondencias verdaderamente correctas, lo cual tiene un impacto insignificante en el resto del algoritmo, en lugar de que cometa más errores cuando considera correspondencias como correctas, ya que esto aumenta la probabilidad de que se dejen pasar correspondencias erróneas a los posteriores procesos del algoritmo y que pueden tener un efecto significativo. Pero resulta que como este proceso solamente proporciona una etapa inicial para la exclusión de correspondencia atípicas y es el proceso basado en RANSAC quien verdaderamente se encarga de excluir la mayoría de estas, dicho comportamiento del proceso no debería de tener una repercusión de peso en la estimación de los parámetros de movimiento, siempre y cuando los parámetros de control relacionados con el proceso de RANSAC se elijan de forma correcta y el límite de rotación sea igual o menor a 10 grados, para garantizar que la cantidad de correspondencias incorrectas que se tengan que excluir sea relativamente bajo.

La última conclusión a la que se puede llegar con los resultados de la primera prueba es que para un mismo intervalo dentro del cual varía la rotación que puede experimentar el rover entre ambos instantes de tiempo, la precisión del proceso de establecimiento de correspondencias decrece conforme aumenta la cantidad de componentes con las que se puede describir el movimiento que

experimento el rover entre los dos instantes, lo cual es totalmente de esperarse, ya que es obvio que al proceso le cueste cada vez más establecer una correspondencia de forma correcta conforme aumenta la cantidad de componentes con las que se describe el movimiento. Lo que sucede es que la imagen después del movimiento va a ser cada vez más diferente en comparación con la imagen antes del movimiento y por consiguiente, el rastreo de la ventana de correlación va a ser cada vez más complicado, ya que la ventana de correlación en la imagen después del movimiento que verdaderamente coincide con la ventana de correlación en la imagen antes del movimiento, va a estar rotada y cada vez más lejos de su posición original antes del movimiento. Además, mientras que la precisión baja, la sensibilidad se mantiene prácticamente constante, lo cual implica que el proceso considera como correctas el mismo porcentaje de todas las correspondencias verdaderas sin importar la cantidad de componentes del movimiento, es decir, sin importar si hubo rotación, traslación vertical, traslación horizontal o cualquier combinación de estas. Entonces, la brecha entre falsos positivos y negativos se hace cada vez más grande entre más componentes del movimiento haya, ya que para que la sensibilidad se mantenga constante la cantidad de falsos negativos tiene que disminuir en la misma proporción con la que disminuye la cantidad de verdaderos positivos y para que la precisión disminuya, la cantidad de falsos positivos tiene que aumentar de acuerdo al aumento de la cantidad de verdaderos positivos.

Por otro lado, a partir de los resultados experimentales de la segunda prueba, se concluye que el proceso de estimación de movimiento basado en la descomposición en valores singulares de una matriz 3×3 , que fue adoptado e implementado por Matthies, estima los parámetros de movimiento tridimensionales que describen el movimiento que experimenta el rover entre dos instantes de tiempo consecutivos, la traslación y la rotación, con una precisión y con una exactitud casi perfecta, es decir, con un error de estimación cercano a cero y esto para prácticamente cualquier valor de la varianza del ruido gaussiano de media cero que afecta las posiciones de los puntos que conforman la nube de puntos después del movimiento, siempre y cuando las correspondencias temporales 3D entre las nubes de puntos antes y después del movimiento sean las correctas y sean conocidas. Además, el hecho de que el proceso estime los parámetros de movimiento con un error que tiende a cero, no precisamente significa que se elimine el error de posición presente en la segunda nube de puntos y por lo tanto, el error de coincidencia tampoco, más bien, implica que el proceso realiza una estimación casi perfecta de los parámetros de movimiento, es decir, la traslación y rotación estimada es prácticamente igual a la rotación y traslación real, a pesar del error de posición que afecta a la segunda nube de puntos y a pesar de que dicho error nunca va a permitir que haya una coincidencia perfecta entre las nubes de puntos para cualesquiera valores de los parámetros de movimiento. Ese comportamiento tan ideal en realidad es

totalmente esperado, debido a que el estimador de mínimos cuadrados utilizado tiene la propiedad particular de ser inseguado cuando el ruido que afecta a las posiciones de las nubes de puntos es gaussiano de media cero y misma varianza; y precisamente ese es el tipo de ruido que se les introdujo a las nubes de puntos, por lo que es de esperar que el error de estimación tiendan a ser los correctos, siempre y cuando el numero de puntos considera sea bastante grande.

Otra conclusión menos relevante, es que para un mismo valor de la varianza del ruido que afecta la posición de los puntos que conforman la segunda nube de puntos, el proceso de estimación por descomposición en valores singulares estima las componentes de la rotación tridimensional con una exactitud y una precisión mucho mayor que con la que estima las componentes de la traslación tridimensional, ya que el error con el que estima la traslación, aunque tiende a cero, es el doble del error con el que estima la rotación. Dicha situación particular es de esperarse, principalmente por dos razones. La primera razón es debido a que el proceso desacopla la matriz de rotación y el vector de traslación de la minimización del error cuadrático medio, de tal forma que este estima la matriz de rotación directamente de la minimización, y posteriormente, estima el vector de traslación considerando el estimado de la matriz de rotación, el cual contiene un error de estimación por la naturaleza del proceso. Por consiguiente, es de esperarse que el error en la estimación de la traslación sea de cierta mayor al error en la estimación de la rotación, debido a que para la estimación de la traslación se deben de considerar estimados de la rotación que contienen errores de estimación, entonces se podría decir que al error intrínseco de traslación se le adiciona ese error de rotación y termina siendo mas grande. La segunda razón tiene que ver con que el límite de la traslación (18 centímetros) es prácticamente el doble del límite de la rotación (10 grados) y es de esperarse que entre mayor sea el valor de los parámetros de movimiento, mayor va a ser el error de estimación y en este caso, como la traslación puede tomar valores más altos que la rotación, su error de estimación es más alto, esto en comparación con la rotación. Aunque esa diferencia al final es insignificante debido a que ambos errores son aproximadamente cero, solo que si se puede afirmar que el error de la rotación esta más cercano al cero absoluto que el error de traslación.

Por otra parte, de los resultados experimentales obtenidos en la tercera y última prueba, se puede concluir que el algoritmo de odometría visual estereoscópico implementado, el cual es una versión modifica de la primera parte del algoritmo de Larry Matthies, estima los parámetros de movimiento tridimensionales que describen el movimiento que experimenta el rover entre dos instantes de tiempo consecutivas, excluyendo las correspondencias temporales atípicas, con un error que aumenta y se aleja cada vez más del valor ideal de cero, esto conforme aumenta la varianza del ruido modelado como gaussiano

y de media cero, el cual afecta a los valores de intensidad de los píxeles que conforman la imagen después del movimiento. Lo anterior es totalmente esperado debido a que es obvio que entre mayor sea el efecto que tiene dicho ruido sobre la imagen después del movimiento, más difícil es para el algoritmo establecer correspondencias verdaderamente correctas y por lo tanto, excluir las correspondencias incorrectas o atípicas, lo que llega a afectar considerablemente la precisión y exactitud del algoritmo, ya que este termina estimando el movimiento con un grupo de correspondencias tanto típicas como atípicas, y su vez, provoca que las nubes de puntos nunca lleguen a coincidir de forma perfecta, más bien, que estas se alejen cada vez más, debido a que los parámetros de movimiento estimados se alejan cada vez de los parámetros reales. Específicamente, lo que sucede es que entre más grande sea el ruido, mayor es la cantidad de correspondencias que se establecen incorrectamente, debido a que el rastreo de la ventana de correlación es cada vez más complicado, ya que la ventana de correlación en la imagen después del movimiento que verdaderamente coincide con la ventana de correlación en la imagen antes del movimiento, cada vez se ve más perturbada y afectada por el ruido. Luego, al proceso de RANSAC le cuesta cada vez más excluir todas las correspondencias atípicas o encontrar un grupo de correspondencias correctas y por lo tanto, termina estimando el movimiento con un grupo de correspondencias mixtas, el que tenga la cantidad mínima de correspondencias incorrectas, lo que provoca una disminución paulatina de la precisión y exactitud de los parámetros de movimiento, que a su vez depende de que tan incorrectas sean las correspondencias erróneas, es decir, de que tan alejadas están de la correspondencia verdaderamente correcta, lo cual aumenta conforme crece el ruido.

A partir de los resultados de la tercera prueba, también se llega a la conclusión de que el algoritmo implementado es inconsistente, ya que para un mismo valor de la varianza del ruido presente en la imagen después del movimiento, puede que el error en la estimación de la traslación y la rotación sea pequeño o puede que sea grande, es decir, algunas veces estima los parámetros de movimiento con un error menor y otra veces con un error mayor y no solamente sucede para un mismo valor de la varianza, también sucede algo similar para valores consecutivos de la varianza, es decir, puede que para una varianza mayor a una varianza dada, el algoritmo estime los parámetros con un error menor, cuando lo que se espera es un error mayor y de igual forma, puede que para una varianza menor a una varianza dada, el algoritmo estime los parámetros con un error mayor, cuando se espera un error menor. Además, esa inconsistencia es cada vez mayor conforme aumenta la varianza del ruido, es decir, la diferencia entre el error más bajo y el error más alto de la estimación de los parámetros de movimiento para un mismo valor de la varianza es cada vez mayor y para varianzas consecutivas es cada vez más probable que el algoritmo estime los parámetros con un error menor cuando debería de ser mayor

y viceversa. Ese comportamiento indeseado del algoritmo es de esperarse por múltiples razones. Lo que pasa es que el algoritmo encuentra aleatoriamente un grupo de correspondencias y es esa aleatoriedad lo que ocasiona que para un mismo valor de la varianza del ruido o para varianzas consecutiva, que el grupo de correspondencias varíe demasiado, es decir, que varía mucho la cantidad de correspondencias correctas e incorrectas que conforman dicho grupo y también que tan malas son las correspondencias incorrectas. Entonces, al proceso de RANSAC del algoritmo van a llegar grupos de correspondencias que varían mucho ya sea para un mismo valor de la varianza o para varianzas consecutivas y por lo consiguiente, en algunas ocasiones va a ser capaz de excluir todas las correspondencias incorrectas y estimar el movimiento con una buena precisión y exactitud, pero en otras ocasiones no va a ser capaz de excluirlas todas, principalmente debido a que el número de veces que se repite el procedimiento de RANSAC no es suficiente para encontrar un grupo con correspondencias donde todas son correctas y en esos casos estima el movimiento con una precisión y exactitud menor.

De la mano de las dos conclusiones anteriores, cabe destacar que se pueden esperar resultados bastante buenos de parte del algoritmo implementado, es decir, una alta y consistente precisión y exactitud en la estimación de los parámetros de movimiento y por lo tanto, en la coincidencia de las nubes de puntos, cuando la varianza del ruido que afecta a los valores de intensidad de la imagen izquierda después del movimiento, es igual o menor a 8 de intensidad al cuadrado o de forma equivalente, cuando su desviación estándar es igual o menor a aproximadamente 3 de intensidad, ya que en esos casos el error de traslación, de rotación y de coincidencia son relativamente pequeños y muy cercanos a cero. Lo que sucede es que cuando el ruido es relativamente pequeño, la probabilidad de que el algoritmo establezca correspondencias correctas aumenta y por lo tanto, es más fácil para el algoritmo excluir todas o la mayor parte de las correspondencias incorrectas o atípicas, debido a que son relativamente pocas, por lo que termina estimando el movimiento con un grupo de correspondencias que son mayoritariamente típicas, lo cual se traduce en un error de estimación bastante pequeño y una coincidencia bastante buena de las nubes de puntos, ya que los parámetros estimados son muy parecidos a los reales. Además, lo anterior implica que el grupo de correspondencias con la que estima los parámetros de movimiento va a ser más consistente para un mismo valor de la varianza del ruido por lo que se espera que el error no difiera mucho y así mismo, que la probabilidad de que para varianzas consecutivas el error sea menor cuando debería ser mayor y viceversa.

6.2. Recomendaciones

Se proponen distintas recomendaciones para mejorar y optimizar el funcionamiento general del algoritmo de odometría visual estereoscópico y por lo tanto, de todos los procesos que lo conforman; para realizar un ajuste más fino de los parámetros de control del algoritmo mediante las mismas pruebas o métodos externos; para mejorar y modificar las pruebas experimentales realizadas de tal forma que los resultados muestren un comportamiento más apegado a lo que pueda suceder en la realidad con el funcionamiento del algoritmo; y para probar más a fondo el funcionamiento del algoritmo frente a múltiples situaciones que no fueron consideradas; entre otras más.

Se recomienda repetir la primera prueba pero introduciendo un ruido modelado como gaussiano, de media cero y varianza conocida, a los valores de intensidad de los píxeles de la imagen después del movimiento, para de esta forma analizar el funcionamiento que tendría el proceso de establecimiento de correspondencias temporales en una situación real donde las imágenes se ven afectadas por el ruido, principalmente, para determinar si aún se puede garantizar un funcionamiento bastante bueno del proceso cuando el límite de rotación es igual o menor a 10 grados, es decir, para determinar si la precisión y la sensibilidad siguen siendo de más del 80 % o si estas se degradan, situación en la cual se tendría que establecer una nueva rotación máxima, posiblemente de 5 grados, para garantizar un buen funcionamiento del proceso y que la probabilidad de que se establezcan correspondencias incorrectas sea baja. Así mismo, para determinar si los comportamientos que presenta el proceso de establecimiento de correspondencias temporales cuando no hay ruido presente en las imágenes, se mantienen o cambian de alguna u otra forma, como por ejemplo, que en lugar de que el proceso sea propenso a cometer más falsos positivos que negativos, que sea al contrario.

También se recomienda repetir la primera prueba pero para diferentes valores del coeficiente de similitud mínimo, el cual se utiliza dentro del proceso de establecimiento de correspondencias temporales para determinar en una fase inicial si la correspondencia establecida se puede considerar como correcta o no, es decir, para excluir correspondencias atípicas en una primera instancia. De esta forma se podría realizar un ajuste mucho más fino de dicho parámetro, eligiendo aquel valor con el que se obtengan los mejores resultados, es decir, con el que la precisión y la sensibilidad general de todos los casos fueron las mayores, en comparación con otros valores. Así mismo, se sugiere que se repita la primera prueba, pero ahora para diferentes valores del ancho y alto de la ventana de correlación, siempre manteniendo una ventana cuadrada, es decir, que el ancho y el alto tengan el mismo valor, esto principalmente para realizar un ajuste más fino de las dimensiones de la ventana de correlación, eligiendo el valor para el ancho y alto con el que se obtengan los mejores

resultados. Cabe destacar que el correcto ajuste de los dos parámetros anteriores sirve para obtener el mejor funcionamiento posible por parte del proceso de establecimiento de correspondencias temporales y por lo tanto, mejorar el funcionamiento del algoritmo general.

Así mismo, para la primera prueba, sea la original o con modificaciones o mejores, se sugiere utilizar un límite para la traslación vertical que sea diferente y menor al límite de la traslación horizontal, esto para eliminar la diferencia que existe entre los casos con solo traslación horizontal y los casos con solo traslación vertical cuando los límites son iguales, lo cual puede generar resultados incorrectos o engañosos. Entonces, por ejemplo, se podría usar 120 para el límite horizontal y 90 para el límite vertical, que se obtiene al hacer una regla de tres con las dimensiones de la imagen y el límite de 120.

En cuanto a la segunda prueba, se recomienda que esta se repita pero introduciendo un error diferente a las posiciones tridimensionales de los puntos que conforman la segunda nube de puntos, por ejemplo, el error se puede seguir modelando como gaussiano de media cero, pero en cambio, se puede utilizar una varianza diferente para cada componente de las posiciones tridimensionales, es decir, una varianza para el error de posición respecto a X_l , otra para el error respecto a Y_l y otra para el error respecto a Z_l . La idea principal de esto es determinar que tanto variará el error de estimación cuando se consideran errores de posición que sean diferentes a un error gaussiano de media cero con varianza igual para cada componente de las posiciones tridimensionales, es decir, ante variaciones de las varianzas utilizadas que ocasionan que el proceso de estimación se vuelva sesgado, que el error de estimación no tienda a cero. Esto es importante porque en la realidad el error de posición no siempre se puede modelar como se modelo en la segunda prueba.

Así mismo, para la segunda prueba, sin importar el tipo de error de posición, se aconseja dividir los intervalos dentro de los cuales pueden variar los parámetros de movimiento en intervalos más pequeños y repetir la prueba para cada intervalo. Por ejemplo, si se dividen los intervalos originales en dos, se tendría que realizar la prueba para un intervalo de 0 a 9 centímetros para la traslación y de 0 a 5 grados para la rotación y luego para un intervalo de 9 a 18 centímetros para la traslación y de 5 a 10 grados para la rotación. La idea de estos es obtener unos resultados más controlados y consistentes, ya que se estaría reduciendo la diferencia que puede existir entre el valor más bajo que puede tomar un parámetro de movimiento y el valor más alto, de tal forma que los errores en la estimación no van a variar mucho de un caso a otro para un mismo intervalo. También se sugiere que se utilice como entrada una nube de puntos donde las posiciones respecto al eje Z_l sean bastante diferentes unas de otras, esto para eliminar la diferencia que existe entre los errores relacionados con Z_l y los errores relacionados con X_l y Y_l , cuando se utiliza una nube de puntos donde las posiciones respecto Z_l son demasiado

parecidas, lo que genera resultados engañosos. Entonces, se podría obtener una nueva nube de puntos inclinando la cámara estereoscópica y/o haciendo la superficie irregular.

Además, tanto para la primera prueba como para la segunda prueba, se aconseja que se utilicen diferentes imágenes y nubes de puntos de entrada, respectivamente, de tal forma que estas sean una representación bidimensional y tridimensional de diferentes superficies dentro del campo de visión de una cámara estereoscópica en un entorno controlado como el utilizado en este trabajo. Lo anterior con el objetivo principal de determinar como sería el funcionamiento de los respectivos procesos del algoritmo implementado para distintos tipos de superficies, que tanto variaran las métricas de precisión, sensibilidad y valor F1 para la primera prueba y que tanto variara el error de estimación para la segunda prueba entre una superficie y otra. Entonces, por ejemplo, dentro del campo de visión de la cámara se podrían utilizar fachaletas con diferentes diseños, colores, acabados y materiales para recrear diferentes superficies y también, se podrían incorporar distintos objetos tridimensionales variando la posición, cantidad y tamaño de estos, de tal forma que se capture una imagen izquierda que represente la forma bidimensional de cada una de las superficies recreadas, así como una nube de puntos que represente la forma tridimensional de cada una. Luego, simplemente se repite la primera prueba para cada una de las diferentes imágenes de prueba y la segunda prueba para cada una de las nubes de puntos de prueba.

Por otra parte, para la tercera prueba se sugiera que esta se repita pero para diferentes situaciones que son las siguientes: sin considerar el ruido de intensidad en la imagen después del movimiento ni el error de posición en la nube de puntos después del movimiento, considerando solamente el ruido de intensidad que es la prueba que se realizó en este trabajo, considerando solamente el error de posición y por último, considerando tanto el ruido de intensidad como el error de posición. El objetivo de ejecutar la tercera prueba para esas cuatro situaciones es que se pueda realizar una comparación de los resultados, que son las medias y varianzas de los errores de estimación en la traslación y en la rotación y del error de coincidencia de las nubes de puntos, y así determinar si el funcionamiento del algoritmo de odometría visual estereoscópico implementado se ve más afectado por el ruido de intensidad o por el error de posición, en comparación con la situación donde no se consideró ninguno de los dos, también se puede determinar que tanto degrada el funcionamiento del algoritmo cuando se considera a la vez el ruido de intensidad y el error de posición, que es la situación posible más cercana a la realidad, así mismo, se puede determinar para que varianzas del ruido y el error de posición el algoritmo se podría garantizar un buen funcionamiento y entre otros resultados importantes que se podrían inferir.

También se recomienda utilizar la tercera prueba para realizar un ajuste

más fino de ciertos parámetros de control como la cantidad de repeticiones del proceso basado en RANSAC, que sirve para encontrar un grupo de correspondencias en las que todas sean típicas y así estimar el movimiento de una forma más precisa y exacta, y también el tamaño del grupo de correspondencias que se deben de establecer en cada repetición y que se utilizan para estimar los parámetros de movimiento, para de esta forma garantizar el mejor funcionamiento posible que puede llegar a tener el algoritmo de odometría visual estereoscópico implementado, es decir, para obtener los mejores resultados que este puede llegar a dar bajo ciertas condiciones. Entonces, simplemente se repite la tercera prueba pero para diferentes valores de cada parámetro y se elige el valor de cada parámetro con el que se obtengan los mejores resultados, es decir, los menores errores de traslación y rotación y la menor inconsistencia o dispersión.

Así mismo, al igual con que la segunda prueba, se sugiere para la tercera prueba que se dividan los intervalos dentro de los cuales pueden variar los parámetros de movimiento en intervalos más pequeños y repetir la prueba para cada intervalo, para así obtener resultados que estén más controlados y que sean más consistentes.

Por otro lado, se aconseja repetir la tercera prueba pero utilizando nubes de puntos tridimensionales que describan la forma tridimensional de la superficie que se muestra de forma bidimensional en las imágenes antes y después del movimiento, y que el valor de la componente respecto al eje perpendicular al plano de la cámara varíe de un punto a otro de la nube, en lugar de las nubes utilizadas en este trabajo para la tercera prueba, que son nubes de puntos tridimensionales que se asume que se encuentran sobre el plano de la cámara izquierda antes y después del movimiento, que describen la proyección de la superficie en la imagen y que la componente perpendicular es constante para cualquier punto. Para ello, se debe de encontrar algún método o procedimiento que permita obtener los parámetros de movimiento tridimensionales que describen el movimiento de la nube de puntos de un instante a otro, a partir de los parámetros de movimiento bidimensionales con los que se mueve la respectiva imagen de un instante a otro. De esa forma, si se conocen los parámetros de movimiento bidimensionales con los que se transforma la imagen antes del movimiento para obtener la imagen después del movimiento de forma sintética, se podrían obtener los parámetros de movimiento tridimensionales para transformar la nube de puntos tridimensional antes del movimiento y así obtener una nube de puntos tridimensional después del movimiento de forma sintética, que verdaderamente representa la forma tridimensional de la superficie. De esta forma no se tendría que asumir que las nubes de puntos tridimensionales se encuentran sobre el plano de la cámara en ambos instantes y por consiguiente, se podrían repetir las distintas variaciones de la tercera prueba, pero utilizando nubes de puntos tridimensionales antes y después del

movimiento que verdaderamente describen la forma tridimensional de la superficie, para finalmente determinar si el funcionamiento del algoritmo sigue siendo el mismo o si mejora o empeora.

Por último, una vez que se hayan llevado a cabo todas las recomendaciones anteriores, es decir, que se tenga total conocimiento de en qué situaciones el algoritmo y los procesos funcionan bastante bien y en cuales situaciones no, que se tenga una noción bastante clara de que se puede esperar del algoritmo en una situación real para distintos casos y además, que se haya realizado un ajuste bastante fino de los parámetros de control, para aprovechar lo máximo que puede llegar a dar el algoritmo y los procesos que lo componen, se aconseja plantear y llevar a cabo una cuarta prueba en la cual se determine el desempeño, precisión y exactitud del algoritmo implementado, funcionando continuamente dentro de un entorno controlado de pruebas, es decir, dentro de un espacio cerrado del mundo real que fue configurado y adaptado para tal fin. Entonces, por ejemplo, se podría llevar a cabo una prueba donde el algoritmo tiene que ir estimando cada cierta cantidad de tiempo el movimiento que experimenta una plataforma robótica que se mueve en línea recta y a velocidad constante, a partir de las imágenes capturadas por una cámara estereoscópica sujeta rígidamente a un costado de la plataforma y que describen la forma bidimensional de alguna superficie compuesta por fachaletas cerámicas dentro del campo de visión de la cámara. Luego, se puede comparar cada una de las estimaciones realizadas por el algoritmo con los valores reales que se pueden obtener con gran precisión a partir de la sincronización de una estación total a un prisma atado rígidamente a la plataforma. Además, dentro del entorno se pueden variar distintos factores como el tipo de fachaleta utilizada, el tipo de iluminación, la inclinación de la cámara y su altura.

Bibliografía

- Ali, K. y otros (2005). Attitude and position estimation on the mars exploration rovers. páginas 10–27. IEEE Int. Conf. on Systems, Man, and Cybern.
- Aqel, M., Marhaban, M., Saripan, M., y Ismail, N. (2016). Review of visual odometry: types, approaches, challenges, and applications. *Intelligent Industrial Systems*, 5(1):1–26.
- Arun, K., Huang, T., y Blostein, S. (1987). Least-squares fitting of two 3-d point sets. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-9(5):698–700.
- Arun, K., Huang, T., y Blostein, S. (2007). Stereo odometry - a review of approaches. Reporte técnico, Institute of Automation, Department of Electrical Engineering and Information Technology, Chemnitz University of Technology.
- Arvidson, R. y otros (2011). Opportunity mars rover mission: Overview and selected results from purgatory ripple to traverses to endeavour crater. *J. of Geophysical Res.*, 116(E00F15):1–33.
- Baker, K. (2005). *Singular Value Decomposition Tutorial*. The Ohio State University.
- Biesiadecki, J. y otros (2005). Mars exploration rover surface operations: Driving opportunity at meridiani planum. páginas 1823–1830, Hawaii, USA. IEEE Int. Conf. on Systems, Man, and Cybern.
- Biesiadecki, L., Leger, P., y Maimone, M. (2007). Tradeoffs between directed and autonomous driving on the mars exploration rovers. *The Int. J. of Robotics Res.*, 26(1):91–104.
- Borovkov, A. (2009). *Probability Theory*. Springer, New York, Estados Unidos, quinta edición.
- Bradski, G. y Kaehler, A. (2008). *Learning OpenCV*. O'Reilly Media, Inc., California, Estados Unidos, primera edición.
- Clearpath (2020). Husky Unmanned Ground Vehicle. <https://clearpathrobotics.com/husky-unmanned-ground-vehicle-robot/>. Accedido el 23 de Septiembre de 2020.

- Comport, A., Malis, E., y Rives, P. (2007). Accurate quadrifocal tracking for robust 3d visual odometry. páginas 40–45, Rome, Italy. IEEE Int. Conf. on Robotics and Automation.
- Corke, P., Strelow, D., y Singh, S. (2004). Omnidirectional visual odometry for a planetary rover. páginas 4007–4012, Sendai, Japan. IEEE Int. Conf. on Intelligent Robots and Systems.
- Craig, J. (1999). *Introduction to Robotics: Mechanics and Control*. Pearson Education, New Jersey, Estados Unidos, tercera edición.
- Eisenman, A., Liebe, C., y Perez, R. (2002). Sun sensing on the mars exploration rovers. páginas 20–27, Montana, USA. IEEE Aerospace Conf.
- Engel, J., Koltun, V., y Cremers, D. (2018). Direct sparse odometry. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 4(3):611–625.
- Fernández, E., Sánchez, L., Mahtani, A., y Martínez, A. (2015). *Learning ROS for Robotic Programming*. PACKT Publishing, Birmingham, Inglaterra, segunda edición.
- Fischler, M. y Bolles, R. (1981). Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395.
- Fraubdirfer, F. y Scaramuzza, D. (2012). Visual odometry: Part ii - matching, robustness, and applications. *IEEE Robot. Autom. Mag.*, 19(2):78–90.
- Gao, Y. (2016). *Contemporary Planetary Robotics: An Approach Toward Autonomous Systems*. Wiley-VCH, Weinheim, Alemania, primera edición.
- Gluckman, J. y Nayar, S. (1998). Ego-motion and omnidirectional cameras. páginas 999–1005, Bombay, India. IEEE 6th Int. Conf. on Comput. Vision.
- Grotzinger, J. y otros (2012). Mars science laboratory mission and science investigation. *Space Sci. Rev.*, 170(1):5–56.
- Géron, A. (2017). *Hands-On Machine Learning with Scikit-Learn And TensorFlow*. O’Reilly Media, Inc., California, Estados Unidos, primera edición.
- Hartley, R. y Zisserman, A. (2004). *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2 edición.
- Helmick, D., Cheng, Y., Clouse, D., Bajracharya, M., Matthies, L., y Roumeliotis, S. (2005). Slip compensation for a mars rover. página 2806–2813, Edmonton, Alberta, Canada. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems.

- Heverly, M. y otros (2013). Traverse performance characterization for the mars science laboratory rover. *J. of Field Robotics*, 30(6):835–846.
- Horn, B., Hilden, H., y Negahdaripour, S. (1987). Close-form solution of absolute orientation using orthonormal matrices. *Journal of the Optical Society of America*, 5(6):1426–1446.
- Howard, A. (2008). Real-time stereo visual odometry for autonomous ground vehicles. página 3946–3952, Nice, France. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems.
- Howard, A., Morfopoulos, J., Morrison, J., Kuwata, Y., Villalpando, C., Matthies, M., y McHenry, M. (2013). Enabling continuous planetary rover navigation through fpga stereo and visual odometry. páginas 1–9, Big Sky, MT, USA. IEEE Aerospace Conference.
- Hubbard, D. (2007). Vision-based control and flight optimization of a rotorcraft uav. Master’s thesis, Brigham Young University.
- Johnson, A., Goldberg, S., Cheng, Y., y Matthies, L. (2008). Robust and efficient stereo feature tracking for visual odometry. páginas 39–46, Pasadena, California, USA. IEEE Int. Conf. on Robotics and Automation.
- Klancar, G., Zdesar, A., Blazic, S., y Skrjanc, I. (2017). *Wheeled Mobile Robotics: From Fundamentals Towards Autonomous Systems*. Butterworth-Heinemann, Massachusetts, Estados Unidos, primera edición.
- Koubaa, A. (2016). *Robot Operating System (ROS): The Complete Reference*. Springer, Cham, Suiza, primera edición.
- Lacroix, S., Mallet, A., Chatila, R., y Gallo, L. (1999). Rover self localization in planetary-like environments. páginas 433–440, Noordwijk, The Netherlands. Int. Symp. on Artificial Intell., Robotics and Automation in Space.
- Leger, P. y otros (2005). Mars exploration rover surface operations: Driving spirit at gusev crater. páginas 1815–1822, Hawaii, USA. IEEE Int. Conf. on Systems, Man, and Cybern.
- Li, R. y otros (2006). Spirit rover localization and topographic mapping at the landing site of gusev crater, mars. *J. of Geophysical Res.*, 111(E02S06):1–13.
- Li, R. y otros (2008). Characterization of traverse slippage experienced by spirit rover on husband hill at gusev crater. *J. of Geophysical Research: Planets*, 113(E12S35):1–16.

- Lindemann, R. y Voorhees, C. (2005). Mars exploration rover mobility assembly design, test and performance. páginas 450–455, Hawaii, USA. IEEE Int. Conf. on Systems, Man, and Cybern.
- Liu, S., Li, L., Tang, J., Wu, S., y Gaudiot, J. (2018). *Creating Autonomous Vehicle Systems*. Morgan and Claypool Publishers, California, Estados Unidos, primera edición.
- Maimone, M. (2013). Curiouser and curiouser: Surface robotic technology driving mars rover curiosity's exploration of gale crater. páginas 1–2, Karlsruhe, Germany. Planetary Rovers Workshop.
- Maimone, M., Biesiadecki, J., Tunstel, E., Cheng, Y., y Leger, C. (2006). *Surface navigation and mobility intelligence on the Mars Exploration Rovers*, Capítulo 3, páginas 45–69. Intell. for Space Robotics, TSI Press Series.
- Maimone, M., Cheng, Y., y Matthies, L. (2007). Two years of visual odometry on the mars exploration rovers. *J. of Field Robotics*, 24(3):169–186.
- Martinez, G. (2014). Intensity-difference based monocular visual odometry for planetary rovers. En *New Development in Robot Vision*, volumen 23 of *Cognitive Systems Monographs, Berlin, Heidelberg: Springer Verlag*, Capítulo 10, páginas 181–198.
- Martínez, G. (2013). Monocular visual odometry from frame to frame intensity differences for planetary exploration mobile robots. páginas 54–59, Tampa Bay, Florida, USA. IEEE Workshop on Robot Vision (IEEE WoRV).
- Martínez, G. (2017). Field tests on flat ground of an intensity-difference based monocular visual odometry algorithm for planetary rovers. Nagoya, Japan. 15th IAPR International Conference on Machine Vision Applications (IAPR MVA).
- Martínez, G. (2018a). Experimental results of testing a direct monocular visual odometry algorithm outdoors on flat terrain under severe global illumination changes for planetary exploration rovers. *Computación y Sistemas, an International Journal of Computing Science and Applications*, 22(4):1581–1593.
- Martínez, G. (2018b). Improving the robustness of a direct visual odometry algorithm for planetary rovers. Mexico, City, Mexico. IEEE International Conference on Electrical Engineering, Computing Science and Automatic Control (IEEE CCE).

- Martínez, G. (2019a). Extending the measurement error model of a direct visual odometry algorithm to improve its accuracy for planetary rover navigation. Queretaro, Mexico. IEEE International Conference on Applied Science and Advanced Technology (IEEE iCASAT).
- Martínez, G. (2019b). *Folleto del Curso Visión por Computador IE0449 de la Escuela de Ingeniería Eléctrica de la Universidad de Costa Rica*. Laboratorio de Investigación en Procesamiento Digital de Imágenes y Visión por Computador (IPCV-LAB), San José, Costa Rica.
- Massot, M. (2013). camera1394stereo. <http://wiki.ros.org/camera1394stereo>. Accedido el 11 de Agosto de 2020.
- Matthies, L. (1989). *Dynamic stereo vision*. PhD thesis, Carnegie Mellon University, Pittsburgh, PA, USA.
- Matthies, L. y Shafer, S. (1987). Error modeling in stereo navigation. *IEEE J. of Robotics and Automation*, 3(3):239–248.
- Mishkin, A., Limonadi, D., Laubach, S., y Bass, D. (2006). Working the martian night shift - the mer surface operations process. *IEEE Robot. Autom. Mag.*, 13(2):46–53.
- Moravec, H. (1980). *Obstacle Avoidance and Navigation in the Real World by a Seeing Robot Rover*. PhD thesis, Stanford University, Stanford, California, USA.
- Nister, D., Naroditsky, O., y Bergen, J. (2006). Visual odometry for ground vehicle applications. *J. of Field Robotics*, 23(1):3–20.
- Olguín, J. (2019). *3D Motion of Rigid Bodies: A Foundation for Robot Dynamics Analysis*. Springer, Cham, Suiza, primera edición.
- Olson, C., Matthies, L., Schoppers, M., y Maimone, M. (2003). Rover navigation using stereo ego-motion. *Robotics and Autonomous Systems*, 43(4):215–229.
- Olson, E. (2003). *A Primer on Odometry and Motor Control*. Massachusetts Institute of Technology, Massachusetts, Estados Unidos.
- OpenCV (2020a). Affine Transformations. https://docs.opencv.org/3.4/d4/d61/tutorial_warp_affine.html. Accedido el 27 de Octubre de 2020.
- OpenCV (2020b). Geometric Image Transformations. https://docs.opencv.org/master/da/d54/group__imgproc__transform.html#ga0203d9ee5fcd28d40dbc4a1ea4451983. Accedido el 27 de Octubre de 2020.

- Pajares, G. y de la Escalera, A. (2016). *Conceptos y Métodos en Visión por Computador*. Grupo de Visión del Comité Español de Autonomía (CEA), Barcelona, España, primera edición.
- PointGrey (2009). Bumblebee 2 Datasheet. https://www.upc.edu/sct/ca/documents_equipment/d_186_id-488.pdf. Accedido el 23 de Septiembre de 2020.
- Press, H., Teukolsky, S., y Flannery, B. (2007). *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University Press, New York, Estados Unidos, tercera edición.
- Rabaud, V. (2016). stereo image proc. http://wiki.ros.org/stereo_image_proc. Accedido el 11 de Agosto de 2020.
- Raguram, R., Pollefeys, M., y Frahm, J. (2008). A comparative analysis of ransac techniques leading to adaptive time random sample consensus. páginas 500–513, Marseille, France. Computer Vision – ECCV.
- Scaramuzza, D. (2011). Performance evaluation of 1-point-ransac visual odometry. *J. of Field Robotics*, 28(5):792–811.
- Scaramuzza, F. y Fraundorfer, F. (2011). Visual odometry: Part i: The first 30 years and fundamentals. *IEEE Robot. Autom. Mag.*, 18(4):80–92.
- Shi, J. y Tomasi, C. (1994). Good features to track. Reporte técnico, Cornell University.
- Squyres, S. y otros (2004). The opportunity rover’s athena science investigation at meridiani planum, mars. *Sci.*, 306(5702):1698–1703.
- Strasdat, H., Montiel, J., y Davison, A. (2010a). Real time monocular slam: Why filter? página 2657–2664, Anchorage, Alaska, USA. IEEE Int. Conf. on Robotics and Automation.
- Strasdat, H., Montiel, J., y Davison, A. (2010b). Scale drift-aware large scale monocular slam. Zaragoza, Spain. Robotics: Sci. and Systems.
- Stuerzl, W., Burschka, D., y Suppa, M. (2010). Monocular ego-motion estimation with a compact omnidirectional camera. páginas 822–828, Taipei, Taiwan. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems.
- Szeliski, R. (2011). *Computer Vision: Algorithms and Applications*. Springer, Cham, Suiza, primera edición.
- Torres, D. (2020). camera calibration. http://wiki.ros.org/camera_calibration. Accedido el 11 de Agosto de 2020.

- Townsend, J. y otros (2014). Mars exploration rovers 2004– 2013: Evolving operational tactics driven by aging robotic systems. páginas 1–22, Pasadena, California, USA. AIAA SpaceOps Conf.
- Tran, L. (2017). *Data Fusion with 9 Degrees of Freedom Inertial Measurement Unit to Determine Object's Orientation*. PhD thesis, California Polytechnic State University, San Luis Obispo, CA, USA.
- Triggs, B., McLauchlan, P., Hartley, R., y Fitzgibbon, A. (1999). Bundle adjustment — a modern synthesis. página 298–372, Berlin, Germany. International Workshop on Vision Algorithms.
- Tsai, L. (1999). *Robot Analysis: The Mechanics of Serial and Parallel Manipulators*. John Wiley and Sons, Maryland, Estados Unidos, primera edición.
- Umeyama, S. (1991). Least-squares estimation of transformation parameters between two point patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(4):376–380.
- Vasavada, A. y otros (2014). Overview of the mars science laboratory mission: Bradbury landing to yellowknife bay and beyond. *J. of Geophysical Res.*, 119(6):1134–1161.
- Vassallo, R., Santos-Victor, J., y Schneebeli, H. (2002). A general approach for egomotion estimation with omnidirectional images. páginas 97–103, Copenhagen, Denmark. IEEE 3rd Workshop on Omnidirectional Vision.
- Yousif, K., Bab-Hadiashar, A., y Hoseinnezhad, R. (2015). An overview to visual odometry and visual slam: Applications to mobile robotics. *Intelligent Industrial Systems*, 1(4):289–311.
- Zhang, Z., Faugeras, O., y Ayache, N. (1998). Analysis of a sequence of stereo scenes containing multiple moving objects using rigidity constraints. páginas 177–186, Tampa, Florida, USA. IEEE Int. Conf. on Comput. Vision.

A Funcionamiento de ROS

El Sistema Operativo Robótico (ROS, por sus siglas en inglés) es un marco o plataforma flexible de código abierto, es decir, es un entorno de trabajo basada en la colaboración abierta y que se utiliza para el desarrollo de software en robótica (Koubaa, 2016).

Esta plataforma de desarrollo para robots tiene la capacidad de comunicarse tanto con los robots mas nuevos y novedosos, así como con aquellos que están en el mercado desde hace varios años. Además, permite el uso de distintos lenguajes de programación como Python, C++ y Lisp, y puede ser ejecutado en maquinas con sistemas operativos tipo Unix, como Ubuntu y Mac OS X (Koubaa, 2016).

El objetivo primordial de ROS es apoyar a nivel mundial el desarrollo de código reutilizable dentro de la investigación robótica, es decir su objetivo principal es el de compartir y colaborar. También existen otros objetivos relevantes dentro de ROS, los cuales se mencionan a continuación (Koubaa, 2016):

- Llegar a ser tan sencillo como sea posible de tal manera que todo el código que sea escrito en ROS se pueda utilizar también en otras plataformas de desarrollo para robots.
- Brindar diferentes bibliotecas de uso abierto, las cuales consisten en la recopilación de una serie de programas con interfaces limpias y funcionales que faciliten una rápida modificación.
- Aceptar diferentes lenguajes de programación de tal forma que se pueda implementar en cualquier lenguaje moderno como Python.

En cuanto a su funcionamiento y arquitectura, ROS cuenta con dos niveles principales de conceptos: el nivel de sistema de archivos y el nivel de computación gráfica.

A.1. Sistema de archivos

En la figura A.1, se muestra un diagrama del sistema de archivos de ROS, que son una serie de conceptos para explicar como esta conformado internamente, la estructura de sus carpetas y los archivos básicos necesarios para su funcionamiento. ROS es como un sistema operativo, ya que este se divide

en carpetas y estas tienen algunos archivos que describen sus funcionalidades (Fernández et al., 2015):

- Paquetes: son la unidad principal de organización en ROS. Tienen la estructura y el contenido mínimo para crear un programa dentro de ROS. Pueden tener procesos ejecutables (nodos), bibliotecas dependientes, archivos de configuración, conjuntos de datos, entre otros.
- Manifiestos: proporcionan la información básica sobre un paquete, información de licencia y dependencias, indicadores del compilador, entre otros.
- Pilas: es una colección de varios paquetes que tienen una misma funcionalidad.
- Manifiestos de pila: proporcionan información sobre una pila como la licencia y sus dependencias con otras pilas.
- Mensajes: es la información que un proceso envía a otros procesos.
- Servicios: definen las estructuras de datos de solicitud y de respuesta de los servicios usados en ROS.

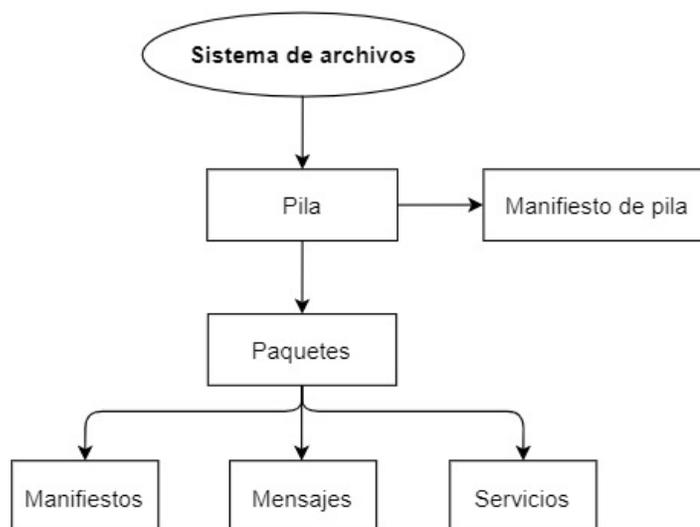


Figura A.1: Esquema del nivel de sistema de archivos de ROS.

En la figura A.2, se muestra un esquema de la jerarquización de ROS cuando se tienen varias pilas.

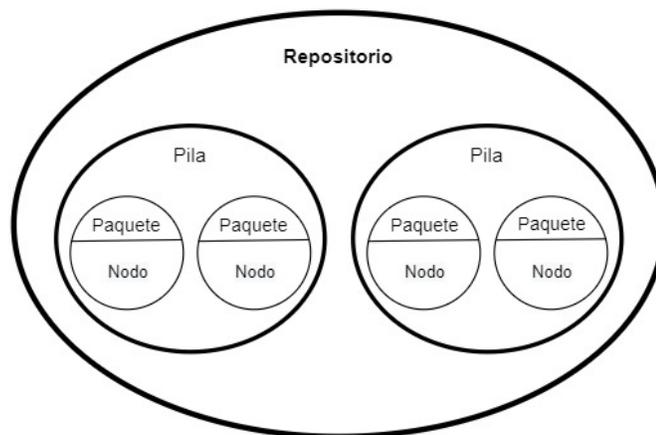


Figura A.2: Esquema de la jerarquización de ROS.

A.2. Computación a nivel gráfico

En este nivel se lleva a cabo la comunicación entre procesos y sistemas. ROS crea una red donde todos los procesos están conectados entre sí, de tal forma que cualquier nodo del sistema puede acceder a esta red e interactuar con otros nodos, transmitir datos a la red y ver la información que otros están enviando (Fernández et al., 2015).

Los conceptos básicos dentro de este nivel son los siguientes:

- **Nodos:** son procesos donde se llevan a cabo cálculos. Si se quiere que un proceso pueda interactuar con otros nodos, se debe de crear un nodo con dicho proceso para poder conectarlo a la red. Usualmente, un sistema está compuesto por muchos nodos para controlar diferentes funciones. Es mejor tener varios nodos que cumplen con una única funcionalidad, a tener un nodo grande que hace todo a la vez.
- **Maestro:** proporciona los registros de nombres y de búsqueda para todos los procesos o nodos. Si el maestro no se encuentra en el sistema, no se puede comunicar con nodos, mensajes y servicios.
- **Mensajes:** los nodos se comunican mediante mensajes. Un mensaje es una estructura de datos que envían información a otros nodos.
- **Tópicos:** cada mensaje debe de tener un nombre para ser enrutado a través de un sistema de transporte de publicación/subscripción semántica dentro de la red de ROS. Cuando un nodo envía datos o información, en realidad está publicando un tópico. Además, cualquier nodo puede

recibir datos de otros nodos simplemente suscribiéndose al respectivo tópico. Uno nodo puede suscribirse a un tópico y no es necesario que exista el nodo que publica en ese tópico. Es vital que el nombre de un tópico sea único.

- Servicios: brindan la posibilidad de interactuar con los nodos. Cuando se necesita una solicitud o una respuesta de un nodo, no se puede hacer mediante tópicos.

En la figura A.3, se muestra como dos nodos deben de registrarse a través del *maestro* para que estos puedan enviar o recibir datos a través de un tópico.

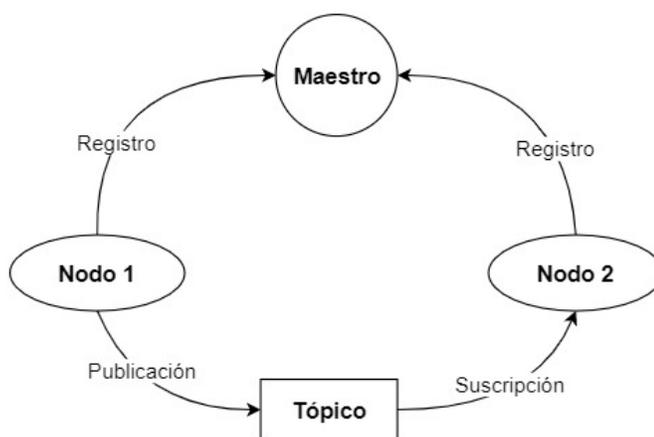


Figura A.3: Comunicación entre nodos a través de un tópico.

A manera de ejemplo, considérese la sección 2.2.5 en la cual se menciona que se usaron tres nodos de ROS para poder obtener una nube de puntos a partir de las imágenes capturadas por una cámara estereoscópica. En la figura A.4 se muestra de una manera más específica y detallada la comunicación entre estos nodos. El nodo llamado *camera1394stereo* toma las imágenes crudas de intensidad de la cámara estereoscópica y las publica como un mensaje en el tópico llamado */image_raw*. Luego, se corre el nodo *camera_calibration*, el cual se suscribe al tópico */image_raw* para obtener las imágenes crudas de intensidad, a partir de las cuales calcula los parámetros de calibración de las cámaras (izquierda y derecha) y finalmente, los guarda en archivos tipo yaml en una carpeta interna de ROS. Por último, se corre el nodo *stereo_image_proc* que se suscribe al tópico */image_raw* para obtener las imágenes crudas de intensidad y luego, utiliza los parámetros de calibración de los archivos tipo yaml para calibrar y rectificar dichas imágenes, las cuales las publica a través del tópico */image_rect*. Además, calcula una nube de puntos a partir de las

imágenes rectificadas y calibradas y la publica a través del tópic llamado */points2*.

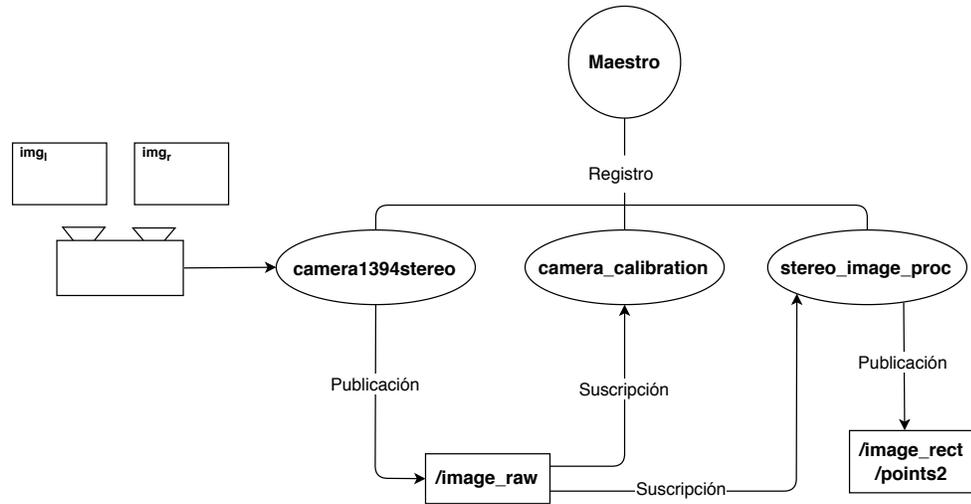


Figura A.4: Ejemplo del uso de tres nodos de ROS para obtener una nube de puntos a partir de una cámara estereoscópica.

B Algoritmo de RANSAC

El algoritmo de consenso de muestra aleatoria (RANSAC, del inglés *iRandom Sample Consensus*) fue publicado por primera vez por Fischler y Bolles (1981) en la revista mensual de la Asociación de Maquinaria Informática (ACM, del inglés *Association for Computing Machinery*). Es una herramienta popular que se utiliza en muchos problemas de estimación robustos en visión por computador, esto es principalmente a causa de su gran capacidad para tolerar una tremenda fracción de valores atípicos.

El algoritmo de RANSAC es un técnica bastante simple pero a su vez muy poderosa, que se aplica usualmente a la tarea de estimar los parámetros de un modelo cualquiera, utilizando datos que están corrompidos debido a una cierta cantidad de valores atípicos que puede ser significativa o no. Lo que hace el algoritmo es estimar lo que se conoce como una relación global que se ajuste a los datos, a la vez que clasifica los datos en típicos (compatibles con la relación) y atípicos (no consistentes con la relación) (Raguram et al., 2008). Dicho de otra forma, es un estimador robusto general, que se basa en un método iterativo que permite estimar los parámetros de un modelo matemático a partir de un conjunto de datos que contienen valores atípicos, cuando no se le debe de dar importancia a dichos valores en las estimaciones, ya que pueden generar múltiples errores e incongruencias; es por esto que también se considera como un método de detección y exclusión de valores atípicos.

Entonces, es un enfoque de estimación de parámetros generales que fue diseñado para hacer frente a una gran proporción de valores atípicos dentro del conjunto de datos de entrada. 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 por computador, RANSAC se desarrolló y perfeccionó desde dentro de dicha comunidad. 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, 1981), 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.

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 (Hartley y Zisserman, 2004), 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 de S .
3. Si el tamaño de S_i (el número de valores típicos) 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.2. Parámetros

Los tres parámetros principales del algoritmo de RANSAC (t , T y N) se analizan a continuación (Hartley y Zisserman, 2004).

B.2.1 Umbral de distancia t para determinar la compatibilidad dato/modelo

El umbral t se elige de tal forma que la probabilidad de que un dato sea un valor atípico sea α . Para poder calcular el valor de t se requiere conocer la distribución de probabilidad de la distancia de un valor típico del modelo. En la práctica, el umbral t se elige de forma totalmente empírica.

B.2.2 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 y $v = 1 - u$ representa la probabilidad de

observar un valor atípico. Se requieren N iteraciones del número mínimo de puntos denotados como m , por lo tanto $1 - p$ esta dado por la expresión (B.1).

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

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

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

Entonces, el número de muestras está vinculado a la proporción más que al número de valores atípicos. Esto significa que el número de muestras necesarias puede ser menor que el número de valores atípicos. Por lo tanto, el costo computacional del muestreo puede ser aceptable incluso cuando el número de valores atípicos es grande.

Además, el número de muestras aumenta con el tamaño del subconjunto mínimo (para un dado v y p dado). Se podría pensar que sería ventajoso usar más que el subconjunto mínimo, para obtener una mejor estimación. Sin embargo, esta posible ventaja se ve superada por el fuerte aumento en el costo computacional incurrido por el aumento en el número de muestras.

B.2.3 Número mínimo T de puntos en el subconjunto de decisión

El consenso aceptable establecido T , se elige de tal forma que el algoritmo se termine si el tamaño del conjunto de consenso es similar al número de valores típicos 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).

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