

UNIVERSIDAD MICHOACANA DE
SAN NICOLÁS DE HIDALGO



División de Estudios de Posgrado de la Facultad
de Ingeniería Eléctrica

**Fusión de lecturas de un telémetro láser con un sistema de
visión estéreo trinocular para detectar obstáculos en tres
dimensiones.**

Adrián Núñez Vieyra

Tesis sometida al Programa de Maestría en Ingeniería Eléctrica para
obtener el grado de

Maestro en Ingeniería Eléctrica

Asesor:

Dr. Leonardo Romero Muñoz

Agosto 2007

Resumen

En esta tesis se presentan los resultados sobre dos problemas relacionados con los sistemas de visión estéreo: el de corrección de la distorsión y el de reconstrucción del entorno 3D utilizando las imágenes corregidas. Utilizamos dos diferentes métodos para llevar a cabo la corrección de la distorsión, el primero es un método de autocalibración, dedicado exclusivamente a la corrección de la distorsión y el segundo está basado en registro de imágenes, a través del cual podemos corregir la distorsión así como determinar información adicional de gran utilidad en la calibración de las cámaras.

El problema de reconstrucción consiste en formar un modelo tridimensional del entorno del robot, en base a dos o más imágenes de una misma escena pero tomada desde diferentes puntos de vista.

Como un proceso previo a la etapa de reconstrucción, abordamos el problema de la *correspondencia de imágenes*. La correspondencia de imágenes es el problema de encontrar elementos correspondientes en las imágenes de dos o más cámaras que observan la misma escena.

En esta tesis se propone una estrategia capaz de generar la información necesaria para la navegación, aún en entornos donde los algoritmos tradicionales de correspondencia no funcionan. Para ello se integra un telémetro láser al sistema estéreo, con el cual se genera información de distancia en la imagen, es decir, una especie de correspondencias artificiales las cuales se utilizan como semilla para propagarse a través de superficies difíciles de identificar por otras metodologías.

Índice general

1. Introducción	1
1.1. Motivación	1
1.1.1. El problema de la calibración estéreo	2
1.1.2. El problema de la reconstrucción	3
1.2. Las dificultades	4
1.3. Objetivo de la tesis	5
1.3.1. Alcance	5
1.3.2. Retos	6
1.3.3. Líneas de Investigación	6
1.3.4. Contribuciones	6
1.4. Enfoque utilizado en esta tesis	7
1.5. Organización del documento	8
2. Corrección de la distorsión con Autocalibración	9
2.1. Introducción	9
2.2. Calibración	11
2.2.1. Calibración Interna	13
2.2.2. Calibración Externa	15
2.3. Distorsión Radial	16
2.3.1. Modelado de la distorsión	16
2.4. Transformada de Hough	18
2.4.1. Propósito de la Transformada de Hough	18
2.4.2. Binarización	20
2.4.3. Representación paramétrica de la recta	21
2.4.4. Representación polar de la línea recta	24
2.4.5. Algoritmo de la Transformada de Hough	25
2.5. Determinación del ajuste de los datos	
a las rectas	27
2.6. Algoritmo para corregir la distorsión	31
2.6.1. Resultados de la implementación	32
2.7. Relleno de los huecos	36
2.7.1. Problemas al expandir una imagen	36

2.7.2. Interpolación bilineal	37
2.8. Conclusiones	40
3. Corrección de la Distorsión utilizando Técnicas de Registro de Imágenes.	43
3.1. Introducción	43
3.2. Función de error	43
3.3. Transformación Proyectiva	45
3.4. Corrección de la distorsión	47
3.5. Optimización de la función de error	47
3.5.1. Método de Gauss-Newton-Levenberg-Marquardt	48
3.5.2. Definición de los Jacobianos	49
3.6. Calibración por Puntos de Correspondencia	52
3.6.1. Método de puntos de correspondencia	53
3.7. Resultados obtenidos con registro de imágenes	54
3.8. Conclusiones	57
4. Sistema de visión estéreo	59
4.1. Introducción	59
4.1.1. Los dos problemas de la visión estéreo	60
4.2. Problema de Correspondencia	60
4.2.1. Consideraciones sobre el problema de correspondencia	62
4.2.2. Rectificación del sistema estéreo	64
4.3. La correlación	68
4.3.1. Algoritmo de Shirai	72
4.4. Problema de Reconstrucción	76
4.5. Conclusiones	81
5. Reconstrucción estéreo	83
5.1. Propagación láser	83
5.1.1. El telémetro láser	84
5.1.2. Integración del telémetro láser a la geometría estéreo	85
5.1.3. Del láser a la imagen	87
5.1.4. Propagación en el mapa de disparidad.	88
5.1.5. Propagación y Correlación	91
5.2. Reconstrucción	92
5.2.1. De la imagen al mundo real	93
5.2.2. Construcción de un mapa del entorno	93
6. Conclusiones	101
6.1. Conclusiones	101
6.2. Trabajos futuros	102

Referencias	104
A. Transformaciones espaciales	107
A.1. Traslación	107
A.2. Rotación	108
A.3. Escalamiento	110
A.4. Coordenadas homogéneas	110
A.5. Transformación compuesta	111
A.6. Transformación Afín	112
B. Optimización multivariable	115
B.1. Optimización para una variable	115
B.2. Optimización multivariable	116
B.3. Solución con el método de Newton	117
B.3.1. Valor inicial	117
B.3.2. Cálculo del Gradiente	118
B.3.3. Cálculo del Hessiano	119
B.3.4. Condición de terminación	119
C. Filtros para imágenes	121
C.1. Filtro de derivadas	121
C.2. Filtro gaussiano	121
C.3. Implementación del filtro Gaussiano y filtro de Derivadas	122
Índice de palabras	126

Índice de figuras

2.1. Funcionamiento de una lente	9
2.2. Modelo de la cámara ideal (pinhole)	10
2.3. Ejemplo de distorsión	10
2.4. Desplazamiento de un punto en la imagen	11
2.5. Proceso de captura	12
2.6. Parámetros externos	13
2.7. Marco de referencia global	15
2.8. Distorsión radial	17
2.9. Transformada de Hough (TH)	19
2.10. Diferentes soluciones de la TH	19
2.11. Patrón de calibración	20
2.12. Ejemplo de binarización	21
2.13. Representación paramétrica de la recta	22
2.14. El dominio paramétrico	23
2.15. Representación polar de una recta	24
2.16. El dominio de Hough	25
2.17. Binarización de una imagen	25
2.18. Detección de líneas con la TH	27
2.19. Dominios de la imagen y de Hough	28
2.20. Cúmulos en el dominio de Hough	29
2.21. Cálculo de la curvatura de un conjunto de puntos	29
2.22. Distancia de un punto a la recta	30
2.23. Patrón de líneas	31
2.24. Cámara Firewire	33
2.25. Binarización de una imagen	33
2.26. Calibración con la TH	34
2.27. Líneas ideales	34
2.28. Imagen expandida	35
2.29. Generación de huecos	37
2.30. Interpolación bilineal	40
2.31. Descripción de la interpolación bilineal	41
2.32. Relleno de los huecos	41

2.33. Detalles del rellenado de huecos	42
3.1. Ejemplo de proyección	45
3.2. Proyección en perspectiva	46
3.3. Patrón de calibración	54
3.4. Corrección con método de GNLM	55
3.5. Corrección con 14 parámetros	55
3.6. Geometría estéreo y el robot	56
3.7. Corrección de imágenes	57
4.1. [Correspondencia de imágenes	61
4.2. Problemas de correspondencia	62
4.3. Región de búsqueda	63
4.4. Línea y plano epipolar	64
4.5. Geometría estéreo	66
4.6. Patrón de alineación	67
4.7. Ejes de proyección paralelos	67
4.8. Sistema estéreo rectificado	68
4.9. Ejemplo de un mapa de disparidad	70
4.10. Mapa de disparidad erróneo	71
4.11. Mapa de disparidad con Shirai	75
4.12. Reconstrucción 3D	77
4.13. La distancia focal	78
4.14. Cálculo de la distancia focal	79
5.1. El telémetro láser	84
5.2. Ilustración del telémetro láser	85
5.3. Integración del telémetro láser a la geometría estéreo.	85
5.4. La geometría láser	86
5.5. El modelo en perspectiva	88
5.6. Del láser a la imagen	89
5.7. Propagación láser	91
5.8. Propagación y correlación	92
5.9. Reconstrucción del entorno	94
5.10. El robot Spirit.	95
5.11. Mapa del entorno	96
5.12. Zona de choque del robot	97
5.13. Construcción del mapa del entorno	98
5.14. El mapa del entorno	98
5.15. Detalles del mapa del entorno	99
A.1. La traslación	108

A.2. La rotación	109
A.3. Rotación sobre un punto arbitrario	112
B.1. Interpretación geométrica del Método de Newton	115
C.1. Derivada de una imagen	122
C.2. Filtro gaussiano	122
C.3. La convolución	123
C.4. Derivadas gaussianas	124
C.5. Detección de esquinas	125

Capítulo 1

Introducción

En esta tesis se abordan dos problemas relacionados con la visión estéreo: *el problema de la calibración* y *el problema de reconstrucción*. En el primer caso se trata de encontrar un conjunto de parámetros que participan en la adquisición de imágenes. Haciendo uso de estos parámetros, el segundo problema se refiere a generar información, a partir de varias imágenes, para poder llevar a cabo la reconstrucción del entorno en tres dimensiones (3D). En ambos casos, nos estamos refiriendo a que un robot con capacidades de movimiento hará uso del sistema de visión estéreo en forma autónoma, con el fin de navegar en su entorno sin colisionar.

En este capítulo se describe el objetivo, los retos y los alcances de esta tesis. Se presentan también diferentes temas relacionados con robótica móvil y visión estéreo que servirán de base para una mejor comprensión de los capítulos posteriores.

1.1. Motivación

Actualmente nos maravillamos con escenas de películas en donde aparecen robots con capacidades asombrosas y generalmente superando al propio ser humano. Aunque hasta nuestras fechas esto no deja de ser ficticio, es innegable que ya existen aplicaciones en robótica móvil que hace algunos años hubieran parecido inalcanzables. Podemos ver como un pequeño robot con dos cámaras de vídeo al hombro es capaz de moverse en el planeta Marte y enviarnos valiosa información sobre los orígenes geológicos de tan lejano mundo.

La robótica móvil es particularmente atractiva porque combina agentes o entidades físicas, sujetos a las leyes naturales, con algoritmos para llevar a cabo las tareas encomendadas. Aunque la mayoría de los robots móviles actualmente en operación son experimentales, algunos de ellos están empezando a ser utilizados en ambientes industriales o de servicios. Las aplicaciones reales apropiadas para robots móviles se caracterizan por tener uno o más de los siguientes atributos: ausencia de un operador humano en el sitio, un costo potencial muy alto para disponer de un operador humano y la necesidad

de tolerar condiciones ambientales que podrían no ser aceptables para las personas [16].

Existen diferentes recursos de los cuales un robot puede echar mano para llevar a cabo tareas de navegación. Por ejemplo podría tratarse de un *robot reactivo*, el cual navegaría en una sola dirección hasta chocar y solo entonces cambiar su rumbo. Un robot más sofisticado podría hacer uso de algún sensor apropiado para detectar los obstáculos antes de chocar con ellos; por ejemplo podría utilizar un anillo de sonares, un telémetro láser, cámaras de vídeo o una combinación de ellos. La capacidad del robot para llevar a cabo tareas de navegación, dependerá de la estrategia y de los recursos que se utilicen así como el tipo de navegación que se quiera alcanzar. No es lo mismo que el robot simplemente detecte obstáculos y evite chocar con ellos a que lleve a cabo elaboradas estrategias para eludirlos de la mejor forma.

Una de las principales limitantes, para establecer la capacidad de navegación de un robot, es el aspecto económico, por ejemplo, los sensores láser de distancia generalmente tienen un costo muy alto, aunque permiten obtener mediciones bastante precisas. Los sonares son muy económicos pero introducen bastante ruido en las mediciones de distancias, por lo que se debe considerar el desarrollo de algoritmos más robustos. Los dispositivos de captura de vídeo nos proporcionan una gran cantidad de información y una buena calidad en la misma, dependiendo de los algoritmos estéreo que se utilicen. Sin embargo es necesaria una calibración previa, ya que de lo contrario la mayoría de los algoritmos disponibles resultarían inútiles.

Otra consideración para determinar los alcances del robot es definir la o las tareas que deberá llevar a cabo. Algunas tareas en donde ha resultado útil el uso de robots móviles son las siguientes [16]: la transportación y entrega de materiales, por ejemplo en un hospital; vehículos inteligentes, ya sea en la forma de asistentes para el conductor del vehículo o vehículos sin operador humano; vigilancia e inspección en la industria, por ejemplo en el monitoreo de áreas radioactivas de plantas nucleoelectricas; en ambientes de difícil acceso, por ejemplo en minas, en el espacio o bajo el agua; en la desactivación de bombas y minas; en tareas de servicio en casas, por ejemplo robots podadores de pasto o robots limpiadores; en tareas de ayuda para personas, por ejemplo sillas de ruedas con cierta autonomía y asistentes robóticos para ancianos.

1.1.1. El problema de la calibración estéreo

¹El problema de la calibración estéreo implica, por una parte, *corregir la distorsión de las lentes de las cámaras*, y por otra, *integrar cada una de las mismas en una geometría conocida* (geometría estéreo). Aunque un sistema estéreo puede trabajar sin que se corrija la distorsión, esto lo limitaría ya que los principales algoritmos de reconstrucción están

¹La palabra estéreo es un acortamiento de Estereofónico cuyo significado es “sonido registrado simultáneamente desde dos o más puntos convenientemente distanciados para que, al reproducirlo, de una sensación de relieve espacial” [7].

basados en un modelo de cámara ideal sin distorsión llamado “pinhole”². Por lo tanto, podemos decir que la corrección de la distorsión trata de aproximar en lo posible una cámara normal, con distorsión, al modelo de la cámara ideal o de pinhole.

El proceso de calibración está relacionado con las posiciones que guarden las cámaras entre sí así como el tipo de lente que éstas utilicen, de tal forma que si alguna cámara cambia de posición con respecto a otra, o si se decide cambiar de lentes, será necesario volver a calibrar el sistema.

En nuestro proyecto también es necesario integrar al sistema estéreo al telémetro láser, esto implica conocer su posición y orientación con respecto a la disposición de las cámaras.

Por lo tanto, la calibración completa del sistema incluye la corrección de la distorsión de cada cámara así como la integración de las mismas y del telémetro láser en una geometría estéreo.

En esta tesis utilizaremos un sistema estéreo trinocular, es decir, en lugar del tradicional sistema de dos cámaras, el nuestro estará integrado por tres. Contar con tres cámaras nos permite obtener más información y generar mapas de disparidad con menos errores.

1.1.2. El problema de la reconstrucción

Después de corregida la distorsión y de establecer una geometría estéreo, se deberá poder reconstruir el entorno 3-D para fines de navegación del robot. Esto implica que a partir de dos o más imágenes tomadas de una misma escena, pero desde diferentes puntos de vista, así como de una serie de lecturas del telémetro láser, se pueda generar información que el robot pueda traducir a distancias del entorno 3-D.

Para poder aprovechar la información que nos brindan dos o más imágenes de una misma escena, es necesario resolver primero el problema de correspondencia, es decir, necesitamos establecer una estrategia que nos permita ubicar elementos de una imagen en otra. También tendremos que solucionar algunos problemas relacionados con la adquisición de imágenes, como son: los problemas de iluminación (sombras, reflejos, etc.), de oclusión (elementos que solo aparecen en una de las imágenes), de escenas que no son fijas (por ejemplo persianas moviéndose con el aire), entre otros.

Para simplificar el problema de correspondencia, se hace necesaria una calibración previa entre las cámaras llamada *rectificación*. La rectificación permite, por ejemplo, que la búsqueda de correspondencias entre dos imágenes se lleve a cabo sobre un único renglón.

Otro problema que se presenta en la mayoría de los algoritmos de correspondencia, es que suelen funcionar solamente bajo ciertas configuraciones del entorno y esta situación se hace más crítica cuando se trabaja con imágenes con superficies que presentan muy

²Pinhole es una palabra de origen inglés que se puede traducir como *agujero hecho o como si fuera hecho por un alfiler*.

pocos cambios, como por ejemplo paredes, o cuando dichas imágenes incluyen elementos que se repiten varias veces, como son las patas de sillas o mesas, barandales, etc.

Otros problemas que afectan en la reconstrucción del entorno, es la gran cantidad de factores que intervienen en la captura de las imágenes y sobre los cuales se tiene poco control, por ejemplo las condiciones de iluminación, ya sea que se trate de luz natural o artificial, las sombras que se generan y que cambian de cámara a cámara, las propiedades de reflectancia de los diferentes materiales que comúnmente podemos encontrar en una escena, las diferencias entre las cámaras a pesar de ser de características similares, los errores humanos, el hecho de que las escenas no son necesariamente estáticas, entre otros.

1.2. Las dificultades

Los problemas de calibración y reconstrucción presentan una serie de dificultades que los hacen especialmente difíciles de resolver:

- *Las cámaras.* Las cámaras digitales no han sido fabricadas para fines de reconstrucción, por tal motivo, especificaciones como la distancia focal o el tamaño de los píxeles en general no son conocidas. De hecho las cámaras son dispositivos que miden el color, la brillantés y la saturación de la luz[16], mientras que para fines de reconstrucción, lo que nos interesa de una imagen es, a qué distancia se encuentran los objetos.
- *Limitaciones perceptuales.* El rango perceptual de las cámaras digitales está relacionado con la apertura de la lente que se esté utilizando. El problema es que a mayor apertura de la lente, mayor distorsión en las imágenes y por lo tanto, esto se traduce como una mayor complejidad en los procesos de corrección.
- *Datos atípicos.* Idealmente la reconstrucción asume que en las imágenes estéreo existen las mismas condiciones de iluminación y que las escenas son estáticas, sin embargo cuando estas condiciones no están presentes, se pueden filtrar falsas correspondencias, lo que se traduce como ruido en los mapas de disparidad. Las superficies homogéneas y los detalles repetitivos en una misma escena pueden agravar aún más esta situación. El principal problema es que si queremos reducir el ruido en los mapas de disparidad, tendremos que sacrificar también bastante información que tal vez si era correcta. ¿Cómo distinguir una de la otra?
- *Errores de medición.* Otro problema de la reconstrucción, es la gran cantidad de mediciones que se tienen que hacer “a mano”, con los inevitables errores que esto conlleva. Esto se puede agravar si no conocemos algunos datos básicos como la distancia focal o el tamaño de los píxeles. La precisión de la calibración depende de la precisión de las mediciones. Si se desea una precisión en la calibración de 0.1mm, entonces las mediciones deberían tener tolerancias menores a .01mm [23].

- *Requerimientos de tiempo real.* Es deseable que los procedimientos de reconstrucción se lleven a cabo en forma rápida para que hagan de las tareas de navegación algo factible. Esto requiere de algoritmos eficientes, implementados en lenguajes de bajo nivel, además de que también puedan distinguir entre los datos de las imágenes para descartar aquellos que no sean de utilidad.
- *El telémetro láser.* Los problemas del telémetro láser están asociados a la reflectividad de las superficies. Si éstas son de color negro, es posible que la luz reflejada sea muy poca; si se trata de un espejo, es posible que la luz no regrese al sensor; si se trata de una superficie transparente, la luz láser pasa por ella sin reflejarse[16].

1.3. Objetivo de la tesis

El objetivo de esta tesis es integrar un sistema de visión estéreo y un telémetro láser que permitan la reconstrucción 3-D del entorno. Las lecturas generadas por un telémetro láser son utilizadas como semillas de propagación en un mapa de disparidad.

Como una prueba para los métodos desarrollados en esta investigación, se implementó una tarea de navegación para un robot móvil, la cual consiste en evadir obstáculos en ambientes de interiores.

1.3.1. Alcance

En esta investigación, se consideran las siguientes restricciones relativas al sistema de visión estéreo, al telémetro láser y al robot móvil utilizado para las pruebas de navegación:

Ambientes interiores estáticos sin marcas artificiales. Se asume que durante los procesos de reconstrucción y de navegación el entorno no cambia. Además se considera que el ambiente no se modifica para disponer de algún tipo de marca artificial con el propósito de ayudar en la navegación del robot.

Configuración del sistema estéreo. El sistema estéreo está integrado por tres cámaras de tipo industrial con características similares. Las imágenes con que se trabaja son de 640×480 píxeles en 256 tonos de gris. El telémetro láser puede generar hasta 361 lecturas, cubriendo 180° sobre un mismo semiplano. En la figura 5.2 se ilustra el modelo del dispositivo láser utilizado en este trabajo.

Robot Móvil. El robot cuenta con los siguientes sensores: telémetro láser, un anillo de sonares y tres cámaras de vídeo. Las cámaras de vídeo y el telémetro láser se integran para proporcionar al robot la capacidad de poder navegar en ambientes de interiores. En el caso de toparse con objetos transparentes como cristales se puede hacer uso de los sonares.

1.3.2. Retos

Los retos que se plantean para el desarrollo de esta tesis son los siguientes:

- *Eliminar la distorsión.* Se deberá eliminar la distorsión de las lentes de las cámaras (ver capítulos 2 y 3), para ello se deberán encontrar los valores de los parámetros que intervienen en dicho proceso, para cada una de las cámaras. Aunque las cámaras son de características similares, los parámetros para corregir la distorsión suelen ser diferentes. Además, en la práctica las imágenes tomadas por diferentes cámaras suelen ser muy distintas, ya que intervienen otros parámetros como la brillantez, el contraste, y algunos procesos que vienen incorporados en el hardware de las cámaras y que también deben ser considerados.
- *Integrar los sensores.* Se tienen que integrar las cámaras de vídeo y el telémetro láser, para que en conjunto puedan generar la información suficiente para la navegación aun cuando las condiciones no sean las mejores.
- *Eliminación de datos atípicos.* En algunos casos el telémetro láser no funciona adecuadamente, por ejemplo, cuando se utiliza sobre superficies muy oscuras o transparentes. Por otra parte los algoritmos de correlación tienden a generar falsas correspondencias cuando se trabaja con superficies homogéneas como pueden ser las paredes. El sistema estéreo debe ser capaz de desechar las falsas lecturas generadas por el telémetro láser así como las generadas por los algoritmos de correspondencia.
- *Reconstrucción del entorno.* A partir de la información generada por el sistema estéreo y por el láser, se debe poder crear un mapa que permita al robot llevar a cabo una navegación sin colisiones.

1.3.3. Líneas de Investigación

En esta tesis se presentan desarrollos basados en las siguientes líneas de investigación:

1. Introducir el telémetro láser a un sistema de visión estéreo, ayuda a solucionar los problemas de correspondencia en superficies homogéneas.
2. Rectificar un sistema estéreo utilizando registro de imágenes.

1.3.4. Contribuciones

Esta tesis está basada en libros y artículos los cuales a lo largo de la misma se irán citando. Sin embargo se presentan algunas estrategias y heurísticas que nos brindaron buenos resultados, por lo que podemos considerar que nuestro trabajo lleva a cabo las siguientes aportaciones:

1. Integrar el telémetro láser, a un sistema estéreo, como herramienta de construcción de los mapas de disparidad, con el objeto de tener mejores mapas en menor tiempo.
2. En el caso de utilizar lentes de mayor apertura, ha sido necesario integrar un tercer coeficiente de distorsión al algoritmo presentado por Leonardo Romero y Félix Calderón [17], el cual es más robusto ya que presenta mejores resultados que cuando son dos y no es tan sensible a los valores iniciales.
3. Se define una estrategia para estimar el valor de la distancia focal en base a imágenes estéreo (ver sección 4.4).
4. Se hacen modificaciones al algoritmo propuesto por Shirai en los criterios de búsqueda de correspondencias, lo que permite reducir el tiempo del proceso.
5. Se pueden convertir las imágenes, adquiridas por un sistema de visión con una geometría dada, a sus correspondientes imágenes de un sistema estéreo rectificado, a partir del algoritmo de calibración presentado en [17].
6. Se produjo artículo “Fusing a Laser Range Finder and a Stereo Vision System to Detect Obstacles in 3D” durante IBERAMIA 2004 [18].

1.4. Enfoque utilizado en esta tesis

En esta tesis se eligió la técnica de registro de imágenes para llevar a cabo la corrección de la distorsión, lo cual nos permitió rectificar el sistema estéreo que es esencial para la construcción de mapas de disparidad.

En las técnicas de registro se cuenta con dos imágenes, una sintética que representa el patrón a seguir y otra que es el mismo patrón sintético pero adquirido por una de las cámaras. La idea del registro es hacer empatar ambas imágenes aplicándole a una de ellas una serie de transformaciones espaciales [3], como pueden ser la translación, la rotación, el escalamiento, además de transformaciones proyectivas y de distorsión radial. Para obtener el mínimo error en el empate, se aplican mínimos cuadrados y el método de Newton.

Podemos aprovechar la técnica de registro de imágenes para rectificar el sistema estéreo. La rectificación de un sistema estéreo, se refiere básicamente a que los elementos de un par de imágenes aparezcan siempre alineados. Para lograr esto, es necesario no hacer por separado el proceso de la corrección de la distorsión, sino considerar la geometría estéreo para desplazar el mismo patrón sintético al frente de las cámaras durante la captura de las imágenes. Otra restricción es que el patrón de calibración se mueva sobre un mismo plano, paralelo al plano de las imágenes de las cámaras. Con esto se forzará a que las tres cámaras se ajusten a un mismo plano y a que se generen correspondencias colineales en las imágenes.

A partir de esta configuración podemos aplicar algoritmos de correspondencia, como por ejemplo el algoritmo de Shirai, para determinar el mapa de disparidad.

Considerando la corrección de la distorsión y la geometría estéreo del sistema, podemos integrar un telémetro láser para proporcionar información adicional, esto se logra trasladando sus lecturas del mundo real al plano de la imagen y de ahí al mapa de disparidad. La información inicial generada por el láser es propagada en la imagen en base a criterios previamente establecidos. Por ejemplo, si el láser arrojó información de distancia de una pared, en la imagen la parte correspondiente a esa pared será contagiada por dicha información, generándose así un mapa de disparidad más completo.

1.5. Organización del documento

En el capítulo 2 se describen los elementos que participan en la adquisición de imágenes utilizando cámaras digitales. Se lleva a cabo la corrección de la distorsión utilizando técnicas de autocalibración y se presentan los resultados obtenidos.

En el capítulo 3 se lleva a cabo la corrección de la distorsión utilizando técnicas de registro de imágenes y se presentan los resultados obtenidos.

En el capítulo 4 se describen las características y conceptos relacionados con la geometría estéreo, la geometría epipolar, la generación de mapas de disparidad y se introducen conceptos sobre reconstrucción.

En el capítulo 5 se integran las lecturas láser a la geometría estéreo y se construyen los mapas del entorno para la navegación del robot.

Por último, el capítulo 6 se muestra una parte experimental de esta tesis y se presentan algunas sugerencias para extender o completar este trabajo, así como las conclusiones del mismo.

Capítulo 2

Corrección de la distorsión con Autocalibración

2.1. Introducción

Dentro de los pasos previos para poder manipular las imágenes adquiridas por una cámara de vídeo, nos enfrentamos al problema de que prácticamente todas presentan, en mayor o menor grado, una distorsión o curvatura en relación a una cámara ideal.

La distorsión es producida por la lente que se utiliza en la cámara. La función principal de la lente es concentrar la luz en un dispositivo fotosensible. En la figura 2.1 se muestra como la luz al atravesar la lente es desviada idealmente hacia un mismo punto llamado punto focal. La distancia entre el punto focal y la lente se le llama distancia focal y generalmente se expresa en milímetros. Sin embargo no todos los rayos de luz logran este objetivo provocando distorsión en la imagen que se forma en el dispositivo fotosensible.

Por lo tanto, la distorsión siempre estará presente y ésta será más o menos notoria dependiendo de la calidad de la construcción de la lente, la cual no conserva el mismo espesor a través de toda su superficie, además que ésta no es plana sino esférica [11].

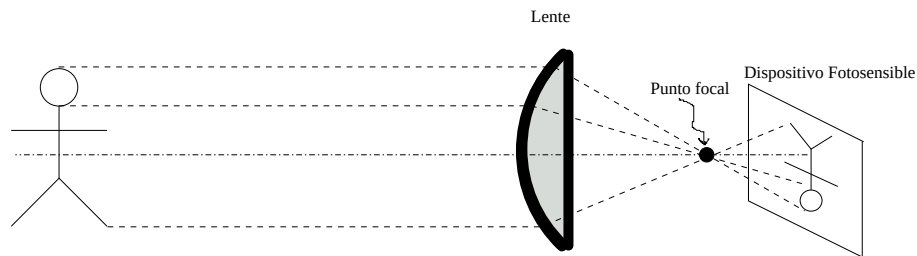


Figura 2.1: Funcionamiento de una lente

Una cámara ideal sería aquella que no introdujera distorsión, y para ello necesitaría no incluir lentes. La figura 2.2 ilustra el llamado “modelo de pinhole”: una caja perfectamente

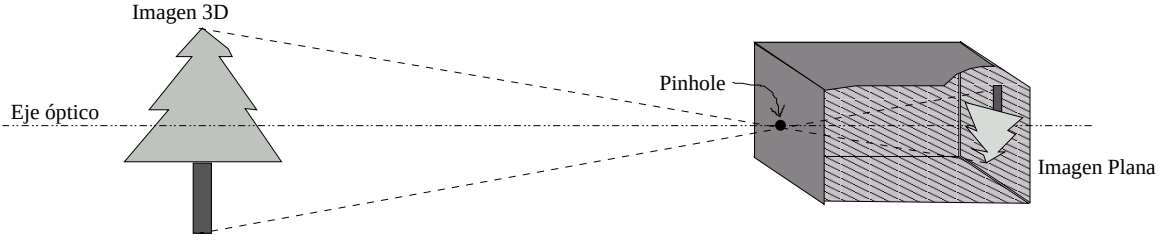


Figura 2.2: Modelo de la cámara ideal (pinhole)

cerrada con un orificio muy pequeño al frente (pinhole) por el cual penetra la luz que forma una imagen plana e invertida. La razón por la que el orificio debe ser muy pequeño (como un pinchazo de aguja) es para mantener la relación uno a uno entre cada punto de la imagen real 3D y cada punto de la imagen plana en la caja. Sin embargo esto tiene un costo: si se colocara papel fotográfico en la parte donde se forma la imagen, se requerirían varios minutos de exposición para poder formar una fotografía [14], lo que hace que este modelo sea poco práctico.

En la figura 2.3 se ejemplifica la distorsión producida por una cámara al tomar una imagen. La figura 2.3(a) muestra una imagen sintética que no tiene distorsión y que incluye varias líneas perfectamente rectas. Por otra parte, la figura 2.3(b) representa el dibujo al ser capturado por la cámara, en donde se aprecia que las líneas cercanas a los bordes ahora ya no son rectas. Por último, en la imagen 2.3(c) las líneas punteadas representan la sobreposición de la imagen distorsionada 2.3(b) con la imagen sin distorsión 2.3(a) en la cual se ilustra como un mismo punto que se encontraba a una distancia r_u del centro de distorsión C_d en la imagen sin distorsión, se desplaza hacia una nueva posición con distancia r_d en la imagen con distorsión. Por lo tanto, la distorsión se traduce como un desplazamiento de un punto con coordenadas (x, y) hacia su nueva posición con coordenadas (x', y') (figura 2.4).

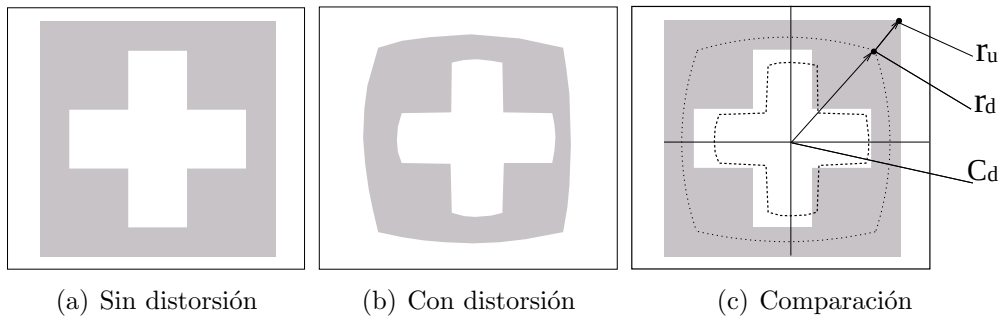


Figura 2.3: Ejemplo de distorsión

La diferencia $x' - x$ y $y' - y$ determinan la distorsión de la imagen en ese punto. El

efecto de la distorsión es distinto para cada punto de la imagen, y dependerá de que tan cerca o alejado se encuentre del centro de distorsión C_d , el cual no necesariamente coincide con el centro geométrico de la imagen y depende de la lente empleada por la cámara. Por esta razón, a este tipo de distorsión se le conoce como distorsión radial ya que la distorsión es una función de la distancia r_u . Entre más grande sea r_u mayor será también la distorsión, es decir, la diferencia entre r_u y r_d .

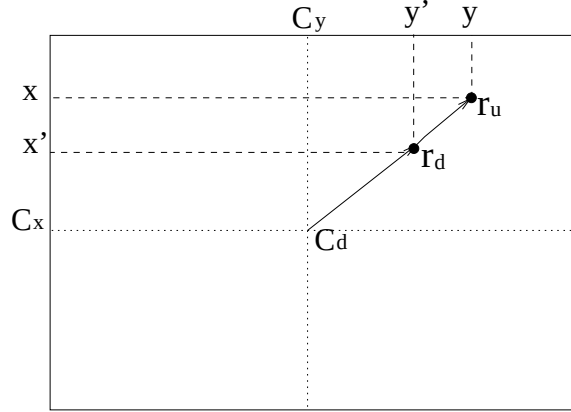


Figura 2.4: Desplazamiento de un punto en la imagen

Este capítulo está dirigido a tratar con el problema de la corrección de la distorsión radial, a la cual llamaremos simplemente distorsión.

Para corregir la distorsión utilizamos una estrategia llamada de autocalibración, en la cual solamente se necesita de una imagen adquirida por la cámara, misma que se va moldeando a través de un conjunto de parámetros de corrección. La corrección de la distorsión puede verse como un estiramiento que se le hace a la imagen distorsionada, con el objetivo de mover todos los puntos de (x', y') hacia su correspondiente (x, y) .

Aunque existen técnicas que infieren información 3-D a partir de imágenes de cámaras no calibradas, el hacerlo nos abre la posibilidad de utilizar un amplio rango de algoritmos para reconstrucción 3-D, los cuales están basados en el conocimiento de los parámetros de la cámara[23].

En la siguiente sección se describen los parámetros que intervienen en la captura de una imagen haciendo énfasis en aquellos que nos permiten corregir la distorsión.

2.2. Calibración

Calibrar una cámara es encontrar un conjunto de parámetros que intervienen en la formación de una imagen. Conocer esos parámetros nos permitirá relacionar en forma correcta un punto del mundo real (X_g, Y_g, Z_g) con un punto (x_c, y_c) en la imagen plana

obtenida por la cámara. En otras palabras, se trata de relacionar un punto del mundo real con su correspondiente pixel en la imagen.

En la figura 2.5 se ilustran algunos detalles interesantes en el proceso de captura de una imagen. El hecho de que la luz reflejada por todos los puntos de un objeto real converga en un solo punto llamada punto focal, garantiza que cada punto del objeto real será mapeado a solamente un punto en la imagen en la cámara. La imagen virtual, señalada en la imagen, es la misma proyección que la imagen en la cámara pero sin estar invertida. Por simplicidad matemática es preferible trabajar con la imagen virtual que con la imagen real en la cámara y en adelante se asumirá que la imagen capturada es la virtual.

Por su naturaleza, los parámetros que intervienen en el proceso de captura de una imagen con una cámara se agrupan en: *fotométricos*, *ópticos* y *geométricos* [23].

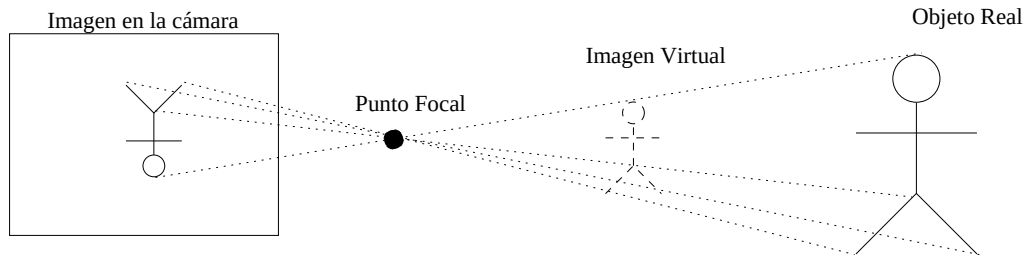


Figura 2.5: Proceso de captura

Parámetros fotométricos. los parámetros fotométricos están relacionados con las condiciones de iluminación y reflectancia del entorno. Así al obtener la imagen en la cámara, esta se verá influenciada por factores como la dirección de la iluminación o por las propiedades de reflectancia de las diferentes superficies que participan.

Parámetros ópticos. Los parámetros ópticos se refieren principalmente al tipo de lente que se utiliza para direccionar la luz. Además se incluyen otros como la apertura de la lente y las capacidades de otros dispositivos como el obturador, el sensor fotoeléctrico, etc.

Parámetros geométricos. Los parámetros geométricos se clasifican en internos (intrínsecos) y externos (extrínsecos). Dentro de los primeros están: la dimensión de los pixeles, la distancia focal y algunos otros que determinan la distorsión generada en la imagen de la cámara. Los parámetros externos están relacionados con la ubicación real, en coordenadas 3D, de la cámara con respecto a un sistema de referencia global (ver figura 2.6). El conocer con exactitud los parámetros internos y externos de la cámara determina la posibilidad de que la misma sea útil en el área de robótica móvil.

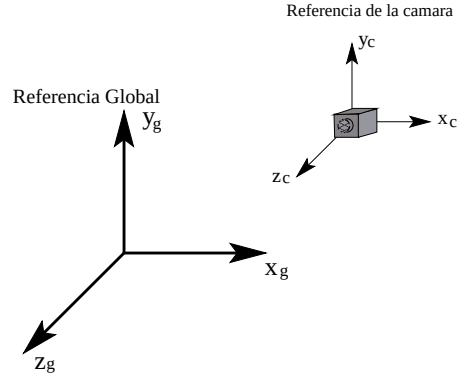


Figura 2.6: Los Parámetros externos se refieren a la ubicación de la cámara con respecto a un sistema de global

Es importante mencionar que el proceso de calibración en una cámara de vídeo implica la intervención humana, en mayor o menor grado, dependiendo del método con el que se trabaje. Por lo tanto, es casi imposible escapar de errores en los datos de entrada, en mediciones, fijación de la cámara, alineación al momento de tomar la imágenes, etcétera. La calidad de la calibración dependerá de la calidad de los datos de entrada y de los cuidados que se tengan para manipular los elementos que participan.

Por otra parte, Trucco y Verri [23] llevan a cabo el proceso de calibración sin considerar los efectos de la distorsión, considerando que usualmente dicha distorsión es muy pequeña y puede ser ignorada cuando no se requiera de mucha exactitud en todas las regiones de la imagen, o bien cuando los píxeles en la periferia puedan ser descartados [23].

Eliminar la distorsión solamente es un paso previo para dar soporte a una gran variedad de algoritmos que hacen de las cámaras un importante instrumento de navegación, desde el enfoque de robótica móvil.

A continuación describiremos el proceso de calibración, que por su tipo de parámetros que se determinan, se clasifica en calibración interna y calibración externa. La primera nos será de gran utilidad en este capítulo, ya que con ella podemos obtener los parámetros para corregir la distorsión. Con la segunda podremos llevar a cabo otro tipo de tareas que están relacionadas con los parámetros extrínsecos del sistema.

2.2.1. Calibración Interna

La calibración interna está orientada hacia las características geométricas y ópticas de las lentes, es decir sus parámetros intrínsecos. Los parámetros intrínsecos son aquellos valores que nos permiten enlazar un punto del mundo real con su correspondiente pixel en la imagen.

La idea básica de la calibración interna es manipular un conjunto de parámetros hasta

encontrar aquellos valores que permitan minimizar o eliminar la distorsión introducida por las lentes que comúnmente se utilizan en las cámaras. Algunos de los parámetros que participan en dicho proceso son:

- La distancia focal de la lente, denotada por f .
- La dimensión de los pixeles, dx, dy (generalmente expresados como un factor de escala $s_x = d_y/d_x$).
- Los coeficientes de distorsión radial, k_1 y k_2 ¹.
- Las coordenadas del centro de distorsión (c_x, c_y) .

Existen diferentes métodos para obtener la calibración de los parámetros intrínsecos de una cámara de vídeo. Destacando en primer lugar, aquellos que utilizan una rejilla de calibración con marcas identificables, y cuyas coordenadas 3D son conocidas. Estas marcas pueden ser puntos, esquinas o alguna característica que pueda ser fácilmente identificada y tratada por una computadora. Generalmente, a estas marcas, se les conoce como puntos de control. En el capítulo 6 de [23] se describe completamente un proceso de calibración que utiliza este enfoque, en donde una vez identificados los puntos en la imagen obtenida por la cámara, se estará en la posibilidad de encontrar los mejores parámetros internos y externos que ajusten la correspondencia entre los puntos de control y los puntos de la imagen. Este método sin embargo asume poca o nula distorsión de las lentes de la cámara, es decir k_1 y k_2 son valores cercanos a cero lo que en algunos casos puede conducir a errores en la calibración.

Otro tipo de calibración es aquel que no necesita conocer las coordenadas de las marcas en el mundo 3D. Este enfoque conocido como autocalibración, asume que todos los parámetros son desconocidos y se busca encontrar el mejor ajuste a través de métodos iterativos[12].

Otra alternativa ha sido extraída de [17], en donde utilizan las técnicas de registro de imágenes para corregir la distorsión, utilizando hasta 13 parámetros. De hecho en algunos artículos publicados por Tamaki ([21] y [22]), aparte de los parámetros geométricos, se introducen algunas características ópticas relacionadas con la iluminación del entorno y debido a ello se manejan hasta 19 parámetros, a través de los cuales se puede corregir la distorsión de una cámara.

Además de los parámetros internos, para poder llevar a cabo cualquier proceso de reconstrucción, es necesario conocer otro conjunto de parámetros relacionados con la geometría de las cámaras, estos son los parámetros externos o extrínsecos. Conocer estos parámetros es la tarea de la calibración externa.

¹Dependiendo del tipo de lente que estemos utilizando, podemos necesitar de más coeficientes de distorsión, por lo que el número de parámetros puede variar.

2.2.2. Calibración Externa

La calibración externa busca encontrar los parámetros externos que definen la ubicación y orientación de la cámara (X_c, Y_c, Z_c) , con respecto a un marco de referencia global conocido (X_g, Y_g, Z_g) , utilizando únicamente información proporcionada por la imagen.

Los parámetros externos se definen como un conjunto de Parámetros geométricos que identifican únicamente la transformación entre el marco de referencia de la cámara (desconocido) y el marco de referencia global (conocido)[23].

En la figura 2.7 se aprecia un punto P con coordenadas P_c y P_g con respecto a un marco de referencia de la cámara y un marco de referencia global respectivamente, en donde la relación entre ambas referencias P_c y P_g se expresa a través de una matriz de rotación R y un vector de translación T de tal forma que:

$$P_c = R(P_g - T)$$

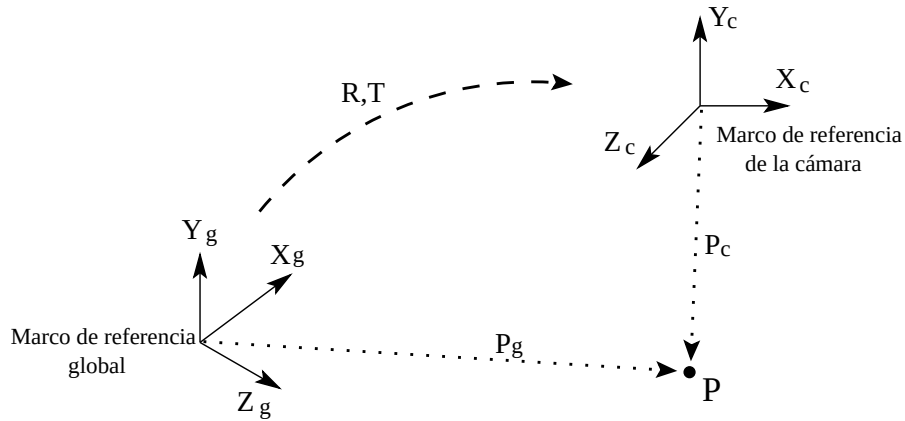


Figura 2.7: Relación entre el marco de referencia global y el marco de referencia de la cámara

El vector T describe las posiciones relativas de los orígenes de los dos marcos de referencia P_g y P_c .

La matriz de rotación R , es un arreglo de 3×3 elementos. Se trata de una matriz ortogonal, es decir que $(R^T R = R R^T = 1)$ y su función es la de hacer coincidir los ejes de P_g y P_c [23].

En la siguiente sección, estableceremos las bases necesarias para encontrar los parámetros intrínsecos que nos permitan corregir la distorsión de las imágenes. Los parámetros extrínsecos serán abordados en capítulos posteriores.

2.3. Distorsión Radial

En la sección anterior se mencionó que todas las lentes introducen distorsión en las imágenes. La distorsión más importante es la de tipo radial, como se ilustra en la figura 2.8, en donde se puede apreciar como los puntos más alejados del centro de distorsión c_d sufren una mayor compresión.

El desplazamiento en el centro de distorsión de la imagen es nulo y se va incrementando conforme más lejos se esté de él.

En esta sección detallaremos un método para llevar a cabo la corrección de la distorsión radial y que es conocido como *Método de Autocalibración*. Con este método encontraremos los parámetros intrínsecos: el centro de distorsión (c_x, c_y) , los coeficientes de distorsión k_1 y k_2 así como el factor de escala s_x . El método de Autocalibración fue extraído del artículo de Kurt Konolige y Luca Iocchi[12]. Éste está basado en que las líneas del mundo real deben conservarse rectas en la imagen. Cuando esto no sucede suele deberse a la distorsión radial introducida por las lentes. El método de Autocalibración consiste en aplicar un conjunto de transformaciones a la imagen, con el objetivo de enderezar las líneas distorsionadas. Por tal motivo, dedicaremos parte de esta sección a hablar de la *Transformada de Hough*, herramienta con la que se puede medir indirectamente la “curvatura” de una línea. Explicaremos también algunas características y procedimientos relacionados con imágenes digitales y al final mostraremos los resultados obtenidos con el método de Autocalibración. Pero antes, explicaremos otros aspectos importantes que nos permitan entender mejor el comportamiento de la distorsión radial.

2.3.1. Modelado de la distorsión

En la figura 2.8 se presentan dos imágenes que utilizaremos como punto de partida para modelar la distorsión radial, la figura (a) representa una imagen típica capturada a través de una cámara. La figura de la parte derecha representa la correspondiente imagen ideal de una cámara que no introduce distorsión. Con ambas imágenes podemos apreciar como un pixel se ha movido de p_u a p_d . La corrección de la distorsión consistirá en regresar el punto p_d a su posición original p_u . En estas imágenes c_x y c_y representan el centro de distorsión de la imagen y son dos de los parámetros que encontraremos con la calibración interna. r_u y r_d representan la distancia en pixeles del centro de distorsión c_d a los puntos p_u y p_d respectivamente.

Si se tratara de una cámara ideal, tanto p_u como p_d tendrían exactamente las mismas coordenadas. Es decir:

$$x_u = x_d$$

$$y_u = y_d$$

sin embargo esto no se da en la práctica.

Si trasladamos el origen de la imagen hacia el centro de distorsión $c_d(c_x, c_y)$ las coordenadas de p_u y p_d son $(x_u - c_x, y_u - c_y)$ y $(x_d - c_x, y_d - c_y)$ respectivamente.

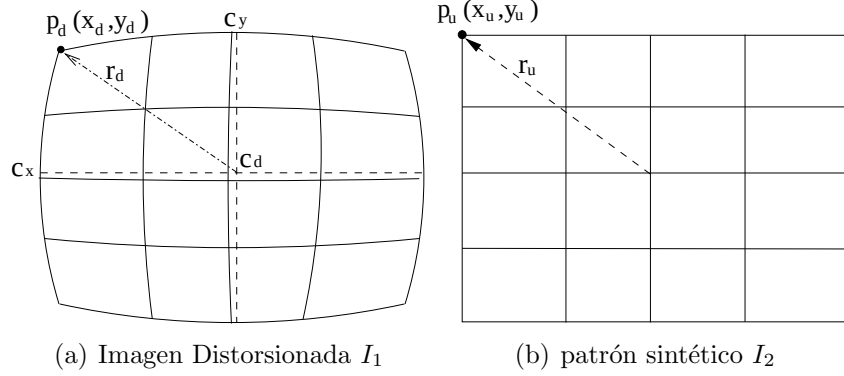


Figura 2.8: Distorsión radial

Como la distorsión tiene un comportamiento radial, necesitaremos multiplicar a r_d por un factor que permita moverlo hacia r_u . Tal factor tiene que estar en función de la distancia r_d . Llamaremos a este factor, primer coeficiente de distorsión radial y lo representaremos como k_1 .

Así, un modelo descrito en [3] que utiliza un solo coeficiente de distorsión radial, puede expresarse como:

$$\begin{aligned} x_u - c_x &= (x_d - c_x)(1 + k_1 r_d^2) \\ y_u - c_y &= (y_d - c_y)(1 + k_1 r_d^2) \end{aligned} \quad (2.1)$$

Aplicando el Teorema de Pitágoras podemos obtener r_d^2 como:

$$r_d^2 = (x_d - c_x)^2 + (y_d - c_y)^2 \quad (2.2)$$

De esta forma, si $k_1 = 0$ querrá decir que no existe distorsión, por lo que $p_u = p_d$.

Finalmente, para completar el modelo de la distorsión radial, debemos tener en cuenta que geoméricamente los pixeles no necesariamente tendrán la misma dimensión horizontal y vertical, por lo que será necesario incluir un nuevo elemento s_x que considere esta diferencia. Así, si d_x representa el tamaño en milímetros del pixel en el eje x y d_y es el tamaño del pixel en milímetros en el eje y , entonces $s_x = d_y/d_x$. En ocasiones los valores de d_y y d_x son proporcionados por el fabricante de la cámara por lo que no se necesitará calcular su valor. En caso de tratarse de pixeles cuadrados ($d_x = d_y$) el valor de s_x es unitario, por lo que no necesitará ser incluido en el modelo de distorsión.

También introduciremos el término $k_2 r_d^4$, el cual es utilizado para casos de lentes con mayor distorsión radial, por lo que llamaremos a k_2 segundo coeficiente de distorsión radial. De esta forma, un modelo de distorsión radial más completo queda como sigue:

$$\begin{aligned} x_u &= \frac{(x_d - c_x)}{s_x} (1 + k_1 r_d^2 + k_2 r_d^4) + c_x \\ y_u &= (y_d - c_y) (1 + k_1 r_d^2 + k_2 r_d^4) + c_y \end{aligned} \quad (2.3)$$

De esta forma, utilizando las ecuaciones 2.2 y 2.3 podemos mover los píxeles en la imagen distorsionada siguiendo un modelo acorde a la distorsión radial. Sin embargo, surge ahora el problema de saber cuales son los valores para k_1, k_2, c_x, c_y y s_x que nos permitan corregir la distorsión.

Una forma de resolver este problema fue presentada en un artículo de Iocchi y Konolige[12], en donde, para corregir la distorsión, solamente se necesitará un patrón de líneas rectas con cualquier pendiente, similar al de la figura 2.8(b)) y al cual denominaremos *patrón de calibración*. La idea para resolver este problema, está basada en que las rectas que se dibujen en el patrón de calibración, deben seguir siendo rectas en la imagen adquirida por la cámara. Pero como la cámara seguramente producirá imágenes distorsionadas, entonces las rectas se convertirán en curvas y como se puede apreciar en la figura 2.8(a), las líneas más alejadas del centro de distorsión de la imagen, tendrán una curvatura más pronunciada.

El problema será ahora encontrar los parámetros internos k_1, k_2, c_x, c_y y s_x , que logren enderezar las líneas distorsionadas en la imagen. Por lo tanto, nuestro problema consistirá ahora en *minimizar una función que mida la distorsión de las líneas formadas por los puntos de una imagen*.

Esto nos conduce a dos problemas por resolver, el primero implica identificar cada una de las líneas en la imagen y una vez identificadas, el segundo problema será medir que tan rectas son. Afortunadamente el primer problema ya ha sido resuelto a través de un método llamado *Transformada de Hough* (TH)[23] y el segundo problema fue resuelto en [12] utilizando algunas propiedades de la misma TH.

En seguida describiremos como se resuelven estos dos problemas en función de la TH, así como algunos conceptos relacionados con la misma.

2.4. Transformada de Hough

Iniciaremos esta sección describiendo la TH como un método utilizado para detectar líneas en una imagen. La TH está implementada para trabajar con imágenes binarizadas, es decir, imágenes cuyos píxeles solamente pueden ser, o negro o blanco. Por este motivo describiremos como llevar a cabo un proceso de binarización en una imagen. Trataremos también algunos temas necesarios para una mejor comprensión del método de la TH y finalmente describiremos una implementación de la misma.

2.4.1. Propósito de la Transformada de Hough

La Transformada de Hough (TH), es un método robusto y efectivo para encontrar líneas en base a un conjunto de puntos de una imagen.

En la figura 2.9 se muestran dos imágenes, una de ellas (a), contiene 4 puntos a partir de los cuales, la TH puede encontrar cual es la mejor línea que ajuste en ellos (b). En realidad la TH se encarga de encontrar aquella línea que pase por el mayor número de

ellos. En este ejemplo podemos ver como la línea en la figura (b) pasa por encima de 3 de los 4 puntos y ninguna otra solución pasará por un número mayor de ellos (ver figura 2.9(b)). Sin embargo con la TH no solamente podremos detectar la mejor solución sino tantas como existan o necesitemos.

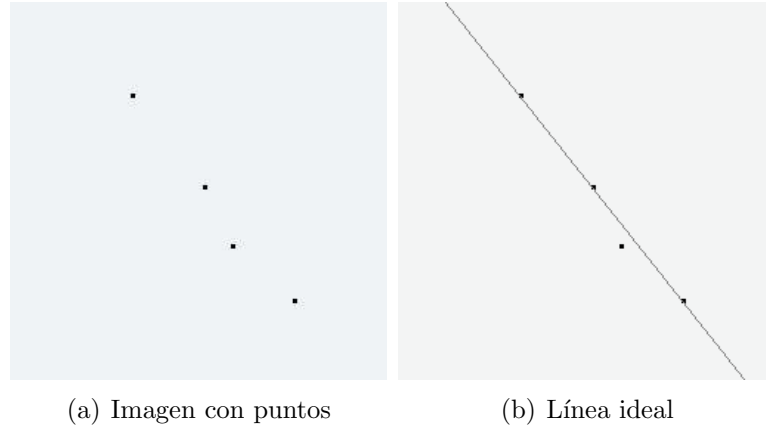


Figura 2.9: Detección de la mejor línea recta con Hough

La forma en que la TH puede encontrar líneas a partir de un conjunto de puntos, es considerando que cada punto puede generar un número infinito de líneas (todas las líneas que pasen por él). Sin embargo, cuando solamente dos puntos están presentes, existirá una sola línea que encaje en ambos (ver figura 2.10) y todas las demás líneas pasarán por un solo punto.

En la figura 2.10 podemos observar que para tres puntos que no estén alineados, la TH nos arrojaría tres mejores posibles soluciones y cualquiera de ellas sería satisfactoria. Finalmente, si la TH se enfrenta a una imagen con una gran cantidad de puntos, tal y como sucede en imágenes reales, a pesar del aparente caos que representa considerar la gran cantidad de soluciones posibles, la TH nos permitiría encontrar las mejores, siempre

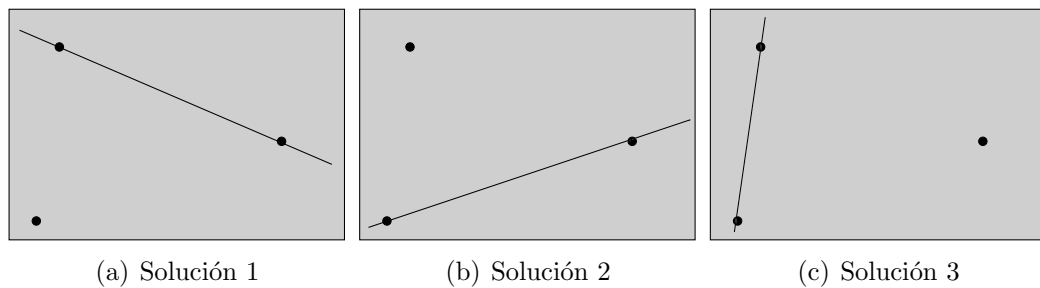


Figura 2.10: En el caso de tener 3 puntos, la TH ofrece tres posibles soluciones.

y cuando hagamos ciertas consideraciones:

- *¿Cuántas líneas?* En primer lugar, necesitamos indicarle a la TH un mecanismo para identificar cuándo se trata de una línea, es decir, en qué momento un conjunto de puntos pasan a ser considerados como una línea. Como la TH encuentra todas las líneas posibles, desde la mejor hasta la menos probable, el problema será disponer de un criterio para considerar una posible línea.
- *¿Cuáles pixeles?* Es necesario llevar a cabo un preprocesamiento de la imagen. En esta operación habrá que indicarle a la TH cuales son los pixeles que nos interesa procesar, es decir, cuales son aquellos pixeles que, por su contraste con aquellos que los rodean, puedan pertenecer a la solución.

La solución del primer problema, en nuestro caso es conocida, ya que aplicaremos la TH a una imagen cuyo número de líneas podemos determinar (ver figura 2.11). Sin embargo, y a pesar de que el patrón sintético esta formado por líneas negras sobre fondo blanco, al momento de ser capturado por la cámara, se convierte en una imagen con una gran cantidad de tonos de grises (ver fig.2.11(b)). La solución del segundo problema, consiste en determinar cuándo un pixel es lo suficientemente cercano al negro para ser considerado parte de una línea (punto válido). Un preprocesamiento de la imagen para distinguir entre pixeles que son válidos y aquellos que no los son es descrito enseguida.

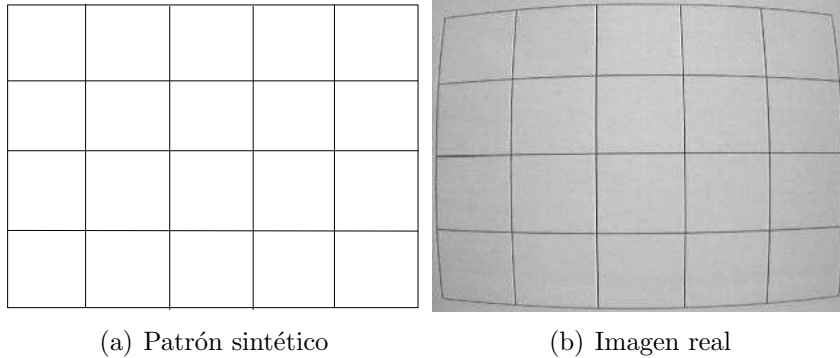


Figura 2.11: (a) El patrón sintético. (b) El mismo patrón al ser capturado por la cámara.

2.4.2. Binarización

En la práctica la TH deberá tratar con imágenes que contienen miles de pixeles, por lo que será necesario determinar cuáles nos interesa procesar, ya que no tendría sentido procesarlos todos. Por lo tanto, el siguiente paso será definir un umbral que nos permita conservar aquellos pixeles que formen parte de las líneas a detectar y eliminar los

restantes. A este proceso se le denomina binarización, ya que cada uno de los píxeles en la imagen solamente podrá tener uno de dos posibles valores, dependiendo si quedó encima o debajo del umbral.

La binarización permitirá que la TH identifique aquellos píxeles candidatos a formar una línea recta. En la figura 2.12 se muestra un ejemplo de binarización. En la figura 2.12(a) se muestra la imagen original en tonos de grises. Un determinado umbral que podemos determinar a prueba y error, nos dirá si el píxel es o no de interés. En este caso solamente nos interesamos por los píxeles más oscuros, los cuales se convierten a negro y los demás se convierten a blancos (ver figura 2.12(b)).

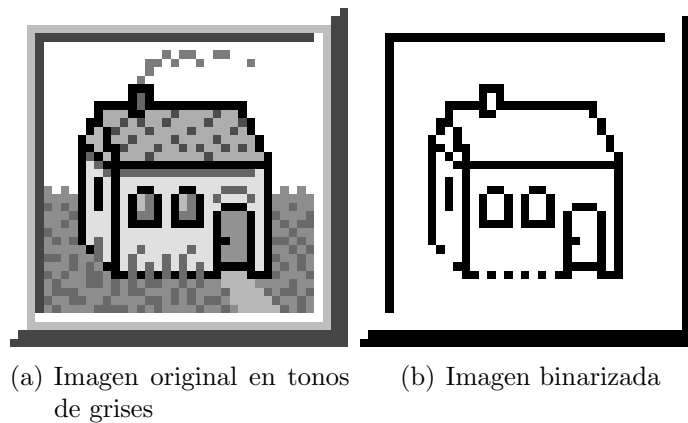


Figura 2.12: Ejemplo de binarización

La imagen binarizada distingue los píxeles de interés, los cuales serán considerados por la TH. Lo que sigue ahora, es que la TH considere todas las líneas posibles que se pueden generar con los píxeles de interés, contabilizando cuantos píxeles de interés atraviesa cada una de ellas y por último, seleccionando aquellas que atravesaron el mayor número de puntos. La forma en que la TH facilita la resolución de estos problemas, es porque aprovecha una de las diferentes formas de representar a la recta: *la representación paramétrica*. A continuación describiremos más sobre este tema.

2.4.3. Representación paramétrica de la recta

Existen diferentes formas para representar una recta r sobre un plano. Algunas de ellas son[5]:

- *Ecuación dos puntos*. La recta r puede ser representada conociendo dos puntos $p_1(x_1, y_1)$ y $p_2(x_2, y_2)$, a través de la ecuación:

$$y - y_1 = \frac{y_2 - y_1}{x_2 - x_1}(x - x_1)$$

- *Ecuación punto pendiente.* Utilizando un solo punto de la recta $p_1(x_1, y_1)$ y su pendiente m , una recta r puede ser descrita a través de la ecuación

$$y - y_1 = m(x - x_1)$$

- *Ecuación pendiente ordenada al origen.* Una recta r puede ser representada utilizando su pendiente m y la distancia que hay del punto n , donde cruza r con el eje y , al origen (ver figura(2.13)). Esta forma de describir una recta es conocida también como la representación paramétrica de la recta y está definida por la relación

$$y = m_x + n$$

en donde $\theta = \tan^{-1}(m)$.

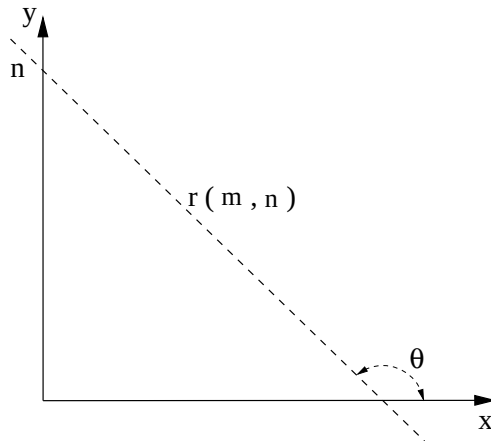


Figura 2.13: Representación paramétrica de una línea recta

En otras palabras, podemos describir cualquier recta utilizando únicamente dos parámetros, m y n .

Si el dominio de la imagen está definido por los ejes coordenados (x, y) , ahora podemos definir el *dominio paramétrico* (m, n) , en donde cada línea en el dominio de la imagen, puede ser representada por un punto en el dominio paramétrico.

Visto desde otro enfoque, si trazáramos algunas de las líneas que pasan por un punto p_i en el dominio de la imagen (figura 2.14(a)), y luego llevamos todas esas líneas al dominio paramétrico, observaríamos que todas las posibles líneas a las que pertenece p_i , formarían una sola línea r_p en la representación paramétrica(ver figura 2.14(b)).

De la misma forma, si tuviéramos dos puntos $(p_{i1}$ y $p_{i2})$ en el dominio de la imagen, ahora se formarían dos líneas en el dominio paramétrico (r_{p1} y r_{p2}). La intersección de r_{p1} y r_{p2} arrojaría un solo valor (m, n) (ver figura 2.14(c)), el cual describiría la mejor recta que ajusta en los puntos del dominio de la imagen.

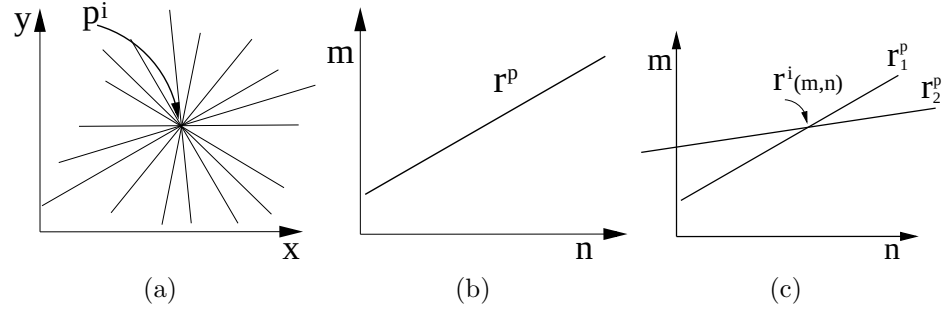


Figura 2.14: (a) Un punto puede pertenecer a diferentes líneas en el dominio de la imagen, (b) Las diferentes líneas en el dominio de la imagen, forman una sola línea en el dominio paramétrico. (c) La intersección de dos líneas en el dominio paramétrico, definen la mejor línea para el dominio de la imagen.

Ahora, si en lugar de 2 tenemos N puntos en el dominio de la imagen, todos aquellos que sean colineales, generarán líneas en el dominio paramétrico que se intersectarán en un mismo punto, por lo que bastará con contabilizar para cual punto (m, n) se tiene el mayor número de cruces y poder así determinar cual es la mejor recta en el dominio de la imagen.

Sin embargo, surgen ahora dos problemas que debemos considerar:

1. *Número de líneas.* Es necesario determinar cuántas líneas generará un punto en el dominio de la imagen. Dicho de otra forma, cuantos puntos se generarán en el dominio paramétrico por cada punto en el dominio de la imagen. Como ya mencionamos antes, todo parte de que un punto en el dominio de la imagen puede ser atravesado por un número infinito de líneas, con diferentes valores para m y n . Esto implica que en el dominio paramétrico m y n puedan tomar una cantidad infinita de valores.
2. *Pendientes infinitas.* Cuando una línea está cerca de ser paralela al eje y , los valores de m tienden a ser muy grandes o muy pequeños. De hecho, el rango de valores posibles para la pendiente es de $[-\infty, \infty]$. Lo que implicaría que las líneas verticales en una imagen difícilmente serían detectadas, cuando de hecho, este tipo de líneas son de las más comunes.

El primer problema se soluciona estableciendo un criterio de discretización. Por ejemplo, podríamos considerar que un punto en el dominio de la imagen es atravesado por un número finito de líneas, por decir cada medio grado, lo que arrojaría que m podría tener hasta 360 valores diferentes.

Para el segundo problema, como $m_r = \frac{y-n_r}{x}$, entonces m puede tomar una infinidad de valores en un rango de $[-\infty, \infty]$. De esta forma, y pensando en una implementación

en un programa de computadora ², las líneas verticales o cercanas a serlo no podrían ser consideradas.

Este problema se agudiza si tomamos en cuenta que en la práctica, frecuentemente las líneas a detectar pueden ser paralelas al eje y . Afortunadamente, una solución a este problema es presentada por Trucco [23], en donde se propone sustituir la representación paramétrica de la recta por su representación polar.

2.4.4. Representación polar de la línea recta

La representación polar de una recta está definida por

$$\rho = x \cos(\theta) + y \sin(\theta) \quad (2.4)$$

Donde ρ representa la distancia más corta de la recta al origen (ver figura 2.15) y θ es la orientación de la línea en un rango de $[0, \pi]$. De esta forma, el problema de la representación paramétrica ha sido resuelto, ya que no trabajamos en función de la pendiente m , la cual puede tomar valores en un rango de $[-\infty, \infty]$, sino en función de θ y ρ .

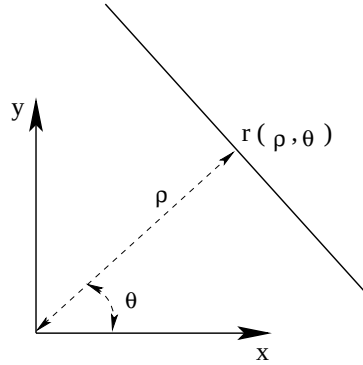


Figura 2.15: Representación polar de una línea recta

Ahora, utilizando la representación polar de la recta, podemos definir el *dominio de Hough* en base a ρ y a θ . A diferencia de la representación paramétrica, un punto en el dominio de la imagen, generará una senoide en el dominio de Hough y no una línea recta. Nuevamente, la intersección de varias senoides indicarán el valor (ρ, θ) que define la mejor línea recta en el dominio de la imagen. La figura 2.9(a) de la página 19 genera la nueva figura 2.16, en la cual se muestran cuatro senoides. El punto donde se intersectan el mayor número de ellas, corresponde a la mejor línea (ρ, θ) que ajusta a los puntos en el dominio de la imagen (ver figura 2.9(b)).

²En [23] se implementa el dominio paramétrico como una matriz tipo entera de $M \times N$, en donde cada celda representa un posible valor (m, n) . Necesariamente N no podrá incluir toda la gama de valores de m y esto representa una limitante.

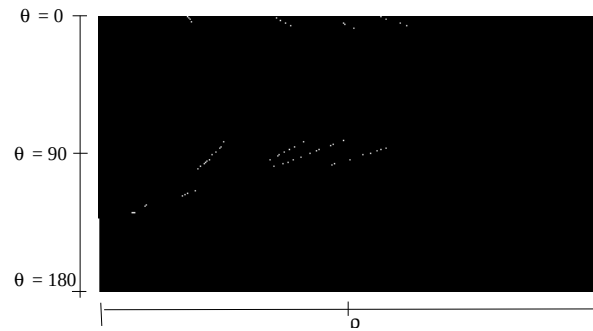


Figura 2.16: El dominio de Hough. Se puede apreciar como las senoides se cruzan en diferentes puntos, lo que sugiere la existencia de varias soluciones.

A continuación se describe un algoritmo para detectar líneas usando la TH y en el cual se aplican algunos conceptos previamente vistos, como binarización, discretización, dominio de Hough y la representación polar de una recta. El algoritmo que se presenta fue tomado de [23].

2.4.5. Algoritmo de la Transformada de Hough

Para ilustrar el funcionamiento de la Transformada de Hough, utilizamos la imagen de la figura 2.17(a). Se trata de una imagen de largo $M = 200$ y ancho $N = 134$. A partir de esta imagen, la cual identificaremos como I , el algoritmo de la TH deberá encontrar las mejores líneas. El procedimiento completo para implementar la TH es el siguiente:

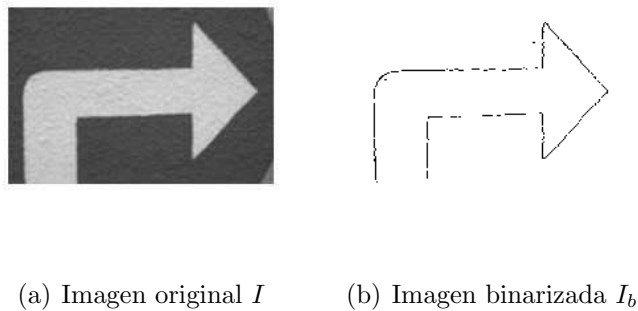


Figura 2.17: Proceso de binarización con un umbral entre 160 y 180 en el rango de tonos de grises

1. El primer paso es binarizar la imagen I para obtener I_b . Existen diferentes programas para manipular imágenes de mapa de bits que nos permitirán fácilmente llevar a cabo este proceso. Para este ejemplo se utilizó el programa Gimp. El resultado de la binarización se aprecia en la figura 2.17(b). Los pixeles en negro son los que nos interesa procesar y los pixeles en blanco serán descartados. En esta ocasión el umbral permitió pixeles con un tono de gris mayor de 160 y menor de 180.³
2. Definir un vector θ de tamaño T , donde T es la resolución de θ cuyos valores irán de 0 hasta π . Así el vector θ tendrá valores $\theta[0] = 1(\pi/T)$, $\theta[1] = 2(\pi/T)$, $\theta[2] = 3(\pi/T)$, ..., $\theta[T-1] = (T)(\pi/T)$.
3. Igualmente definir un vector ρ de tamaño R , donde R es la resolución de ρ cuyos valores irán de 0 hasta $\sqrt{M^2 + N^2}$.
4. Ahora se define una matriz A de T renglones por R columnas y se inicializa en cero. En otras palabras, la versión discretizada del dominio de Hough queda representada por la matriz A .
5. Para cada pixel $I_b[i, j]$ negro y para cada valor de $h = 1...T$:
 - a) Hacer $\rho = j\cos(\theta[h]) + isen(\theta[h])$.
 - b) Encontrar el índice k del elemento de θ que mejor empate con ρ .
 - c) Incrementar en uno el valor de $A[h, k]$.
6. Encontrar los máximos locales $A[h_p, k_p]$ tal que $A[h_p, k_p] > \xi$, donde ξ es un umbral previamente definido.

El resultado final, son un conjunto de pares $(\theta[h_p], \rho[k_p])$ que describen las líneas detectadas en I .

En la figura 2.18(a) se grafica la matriz A . Las partes oscuras representan elementos de A que tuvieron pocos incrementos. Las partes más claras son aquellos elementos de A que más crecieron, por tal motivo podemos apreciar aglomeraciones o cúmulos, en los cuales se encuentren los máximos locales $A[h_p, k_p]$ que estamos buscando. La tarea que tenemos ahora es encontrar los máximos locales de cada cúmulo en el dominio de Hough, para obtener así las líneas que mejor ajusten a los puntos en el dominio de la imagen.

La estrategia que seguimos para detectar los máximos locales es la siguiente:

- Primero buscamos el elemento $A[h_p, k_p]$ de mayor valor en toda la matriz.
- Almacenamos el par de valores $(\theta[h_p], \rho[k_p])$.

³En el caso de imágenes en color el tono de gris se puede obtener como el promedio de sus tres componentes Rojo, Verde y Azul.

- Para evitar detectar más de una vez una misma línea, se pone en cero el elemento $A[h_p, k_p]$ así como sus vecinos⁴.
- Se repite el proceso desde el principio hasta encontrar el número de líneas deseado.

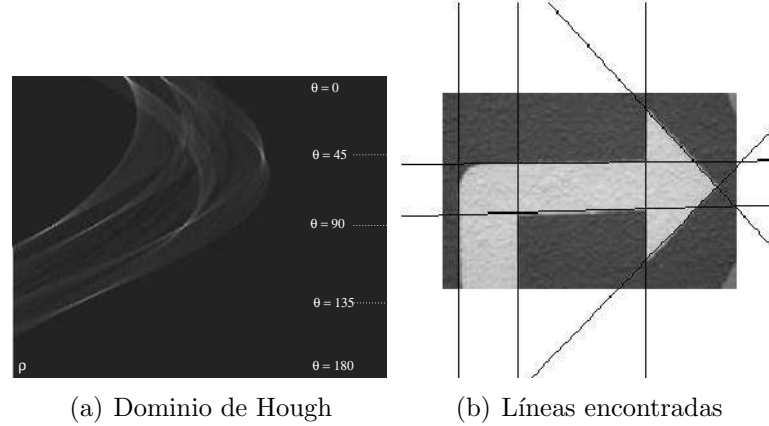


Figura 2.18: (a) El dominio de Hough, obtenido al graficar la matriz A . Se pueden apreciar los cúmulos como los cruces más claros. En cada uno de ellos se encuentran los máximos locales que corresponden a las líneas que mejor ajustan en la imagen I , pero en su forma polar (θ, ρ) . (b) Las líneas que mejor ajustan en el dominio de la imagen.

Finalmente, la figura 2.18(b) grafica las líneas encontradas, sobrepuestas con la imagen original.

En la siguiente sección se utiliza la TH como una herramienta para determinar la distorsión en una imagen. Esto se hace definiendo una función que determine el ajuste de los píxeles de una imagen, a las rectas que se suponen pertenecen.

2.5. Determinación del ajuste de los datos a las rectas

Comenzaremos indicando algunas características adicionales de la TH que nos ayudarán a explicar el objetivo de esta sección. En la figura 2.19 se puede observar como cada línea en el dominio de la imagen genera un grupo de curvas en el dominio de Hough. Dichas curvas se intersectan buscando *ajustarse* hacía un mismo punto el cual representa la mejor línea recta en el dominio de la imagen. Como las líneas en el dominio de la imagen no son rectas, esto se refleja en el dominio de Hough al no intersectarse en un

⁴Se deberán definir cuantos vecinos es conveniente eliminar, para este caso se eliminaron cúmulos cuadrados de 25 píxeles con centro en $A[h_p, k_p]$.

mismo punto, sino en un cúmulo de ellos. En la figura 2.20 se distingue como la línea que en apariencia luce más recta produce un cúmulo más concentrado(2.20(a)) a diferencia de 2.20(b) y 2.20(c) cuya apariencia muestra como las curvas en el dominio de Hough están lejos de intersectarse en un mismo punto.

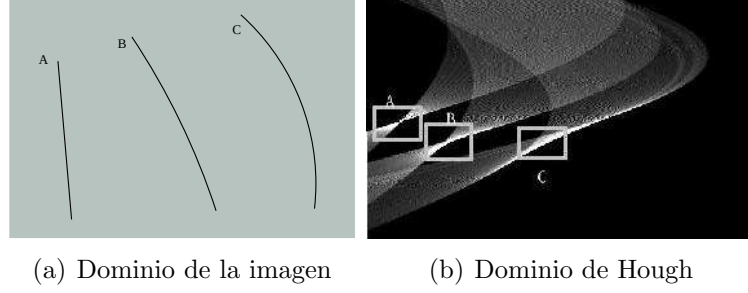


Figura 2.19: Representación de tres líneas con diferente curvatura en el dominio de la imagen(a) y en el dominio de Hough(b)

La dispersión de los puntos dentro del cúmulo es un indicador de como una curva en el dominio de la imagen está cercana a ser una recta. Una forma de medir la curvatura de un conjunto de puntos es calculando la distancia de cada uno de ellos hacia la línea recta que mejor ajuste en ellos. En la figura 2.21 se muestra lo anterior: en donde se miden las distancias $d_1, d_2, d_3, \dots, d_n$ de los n puntos de la imagen en relación a la línea de mejor ajuste r .

Por lo tanto, la curvatura χ_r de un conjunto de puntos en una imagen, se puede medir como:

$$\chi_r = \sum_i dist(P_i, r)^2 \quad (2.5)$$

En donde P_i son cada uno de los puntos a los que ajusta la línea r y $dist(P_i, r)$ es la distancia más corta del punto P_i a la línea r .

En el caso de tratarse de varias posibles líneas, la curvatura total será la suma de la curvatura para cada línea:

$$\chi = \sum_j \chi_{rj} = \sum_j \sum_i dist(P_{i,j}, r_j)^2 \quad (2.6)$$

en donde χ_{rj} es cada una de las posibles líneas, $P_{i,j}$ son los puntos P_i que pertenecen a cada una de las posibles líneas r_j y χ es la curvatura total. Esta función requiere conocer la pertenencia de un punto P hacia una línea r .

Ahora observando la figura 2.22 podemos apreciar como en el dominio de la imagen la distancia de un punto P a una línea r se puede calcular como:

$$dist(P, r) = |\rho_r - \rho_p(\theta_r)| \quad (2.7)$$

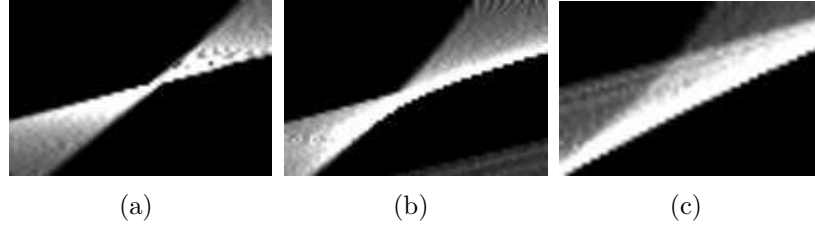


Figura 2.20: Acercamiento a los tres cúmulos de la figura 2.19(b)

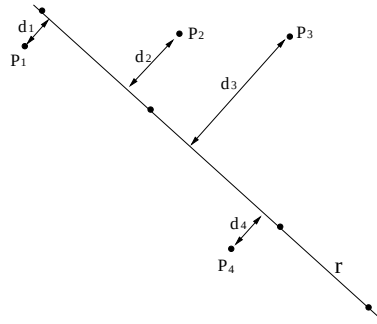


Figura 2.21: Cálculo de la curvatura de un conjunto de puntos

es decir, la distancia de un punto P a una línea r en el dominio de la imagen, se puede calcular en función de los parámetros (ρ, θ) del dominio de Hough. En la misma imagen se muestra una nueva línea z , la cual es la única línea que pasa por P y que comparte el mismo parámetro θ con la línea r , es decir, $\theta_r = \theta_z$. El término $\rho_P(\theta_r)$ representa precisamente una línea paralela a r pero con un ρ diferente.

La ecuación 2.7 expresada en función de z se puede enunciar como:

$$\text{dist}(P, r) = |\rho_r - \rho_z| \quad (2.8)$$

Considerando la ecuación polar de la recta (ec. 2.4), podemos sustituirla en 2.7 para obtener:

$$\text{dist}(P, r) = |\rho_r - (P_x \cos(\theta_r) + P_y \sin(\theta_r))| \quad (2.9)$$

la cual nos permite calcular la distancia más corta de un punto P a la línea r , asumiendo que $P \in r$.

Ahora, sustituyendo la ecuación 2.8 en 2.5 obtenemos:

$$\chi_r = \sum_{\rho} (\rho_r - \rho) \quad (2.10)$$

en donde ya participan todos los puntos que, en el dominio de la imagen, pertenecen

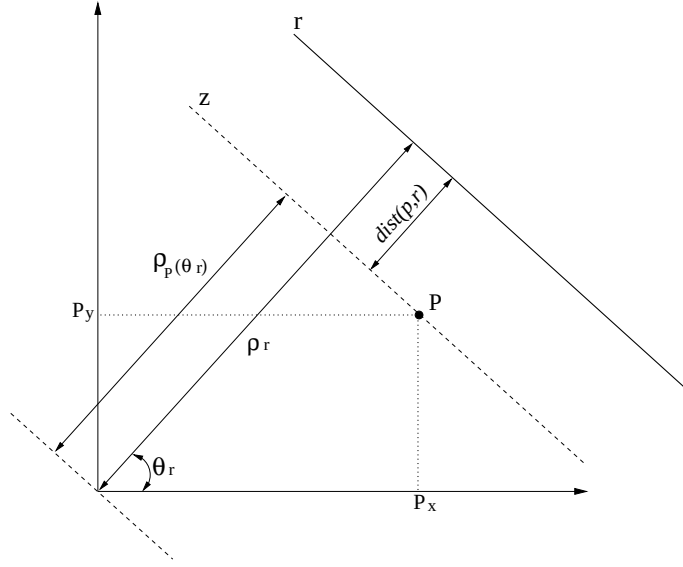


Figura 2.22: Distancia de un punto a la recta

a una misma línea r . Ahora por claridad, en las siguientes ecuaciones, identificaremos a ρ_z simplemente como ρ .

Como ya explicamos anteriormente, los valores de ρ se han discretizado y ya han sido considerados en el algoritmo de la TH. En dicho algoritmo, los valores para ρ y θ de todos los puntos válidos de la imagen, ya han sido calculados y contabilizados. De esta forma, para cada máximo local en el dominio de Hough, corresponde un grupo de líneas en el dominio de la imagen, con diferentes valores de ρ pero con el mismo valor de θ (ver figura 2.22).

De esta forma, y considerando que en la TH sabemos cuantas líneas corresponden a las coordenadas (ρ, θ_r) , podemos expresar la ecuación 2.10 en base a la TH de la siguiente forma:

$$\chi_r \simeq \sum_{\rho} (\rho_r - \rho)^2 TH(\rho, \theta_r) \quad (2.11)$$

en donde los valores de ρ han sido discretizados⁵ y se asume que $\rho \in r$.

Finalmente, considerando las ecuaciones 2.6 y 2.11, podemos determinar la medida de la distorsión $\hat{\chi}$ para el caso de varias líneas como:

$$\hat{\chi} = \sum_j \sum_{\rho \in \Delta_{\rho r_j}} (\rho_{r_j} - \rho)^2 TH(\rho, \theta_{r_j}) \quad (2.12)$$

⁵Precisamente la discretización de la TH es la que define a la ecuación 2.11 como una aproximación

En donde $\Delta_{\rho r_j}$ define un apropiado intervalo alrededor de cada máximo local ρ_r y j especifica la línea sobre la cual se está trabajando.

El desarrollo de esta sección está basado en [12], agregando figuras y explicaciones más detalladas para entender mejor el proceso.

A continuación, explicamos como se llevó a cabo la corrección de la distorsión utilizando la TH. Se muestran los resultados obtenidos durante las diferentes etapas de este proceso.

2.6. Algoritmo para corregir la distorsión

En esta sección se lleva a cabo la tarea de corregir la distorsión. Para realizarla únicamente necesitaremos un patrón de líneas, el cual será adquirido por la cámara. Se calculará la distorsión para la imagen generada, utilizando la ecuación 2.12 y se le aplicarán diferentes valores para los parámetros intrínsecos buscando minimizar la distorsión.

El proceso para corregir la distorsión es el siguiente:

1. Generar un patrón de líneas rectas. La figura 2.23(a) ilustra el patrón que utilizamos. Una recomendación para que el algoritmo de Hough trabaje más eficientemente es que las líneas no estén demasiado cerca una de la otra. Debido a que la distorsión varía con respecto al centro de distorsión de la imagen, que no necesariamente es el centro geométrico, el desplazamiento de los píxeles será mayor en las orillas de la misma que en la parte central. Por ello es recomendable colocar líneas cerca de los límites de la imagen para darnos una mejor idea de como éstas serán enderezadas.

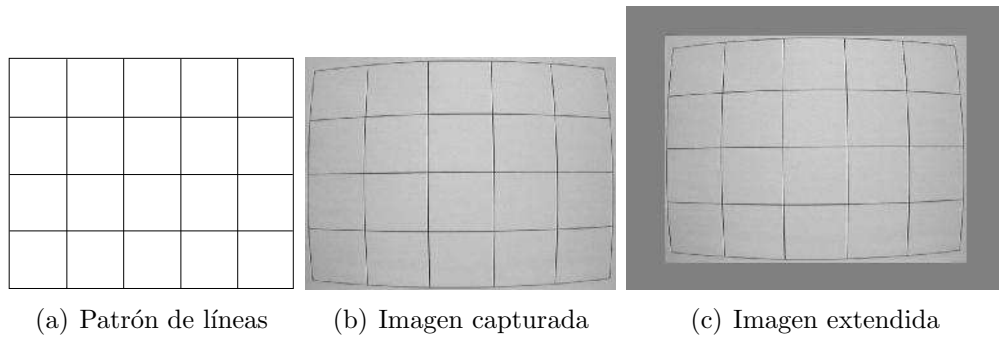


Figura 2.23: Procesos previos para corregir la distorsión.

2. Capturar con la cámara el patrón de líneas. La figura 2.23(b) ilustra la imagen adquirida y será sobre la cual se trabajará.

3. Definir un umbral adecuado y binarizar la imagen (ver sección 2.4). La figura 2.25 muestra este proceso aplicado a la imagen de la figura 2.23(b).
4. Aplicar el algoritmo de Hough sobre la imagen capturada. Es necesario indicar el número de líneas que deseamos detectar. Con esto aumentamos las probabilidades de que el algoritmo funcione correctamente. Es decir, que encuentre las líneas deseadas.
5. Ahora se trabajará sobre el conjunto de cúmulos en el dominio de Hough (ver figura 2.26(a)). Por lo tanto, el siguiente paso será medir la distorsión de la imagen utilizando la ecuación 2.12 de la página 30.
6. Utilizando las ecuaciones 2.3 de la página 17 generar una nueva imagen, esto implica jugar con los parámetros internos de la cámara: moviendo el centro de la imagen, modificando el factor de escala, los coeficientes de distorsión y el tamaño de los píxeles.

Aplicando iterativamente estos tres últimos pasos, podremos encontrar aquella imagen que nos arroje la menor distorsión. El objetivo del algoritmo es minimizar la función de distorsión dada por la ecuación 2.12.
7. Debido a que la imagen corregida se expande (ver figura 2.28), será necesario implementar una estrategia para rellenar los píxeles vacíos generados durante la expansión. En la sección 2.7 se describirá dicha estrategia.

En pocas palabras, el trabajo del algoritmo de corrección de la distorsión, será mover bajo ciertas restricciones los parámetros intrínsecos, con el objetivo de enderezar las líneas de la imagen. Como realmente lo que hacemos es estirar la imagen distorsionada, necesariamente la imagen se expandirá y como la distorsión es radial, provocará que los píxeles que están más alejados del centro de la imagen se muevan más que aquellos que están cerca del centro de distorsión (c_x, c_y) .

A continuación se ilustra paso a paso el funcionamiento del algoritmo aquí presentado.

2.6.1. Resultados de la implementación

Ahora presentaremos los resultados de cada una de las etapas del proceso de autocalibración utilizado.

La figura 2.23(b) es la imagen capturada por la cámara de vídeo. En este trabajo se utilizó una cámara de uso industrial tipo FireWire, modelo i400 de la marca Unibrain, con una resolución de 640x480 y una lente de 4mm. Utiliza un dispositivo fotosensible de la marca SONY con píxeles cuadrados de $5.6 \mu\text{m}$ (ver figura 2.24)[1].

Como se puede apreciar, la imagen de la figura 2.23(b) no está binarizada, aunque al ser procesada solamente se tomarán en cuenta los puntos cuyo tono de gris esté dentro



Figura 2.24: Cámara de vídeo i400 de la marca Unibrain, con interfaz IEEE 1394 (Firewire)

de cierto umbral. Sin embargo el proceso de binarización generaría la imagen de la figura 2.25. La distorsión o curvatura se enfatiza más en las orillas de la imagen y debería notarse que las líneas cercanas al centro son prácticamente líneas rectas.

A la imagen de la figura 2.23(b), fue necesario extenderla. El nuevo tamaño de la imagen original (figura 2.23(c)) fue de 800x600, siendo que el anterior era de 640x480. La razón por la que se tiene que extender es que al procesarla, ésta se expandirá también en todas direcciones. Al expandirse, seguramente algunos de los puntos saldrán de la imagen original, por lo que ya no se les tomaría en cuenta durante el proceso de calibración. Esto conduce a otro error mas grave, ya que al haber menos puntos, el error acumulado sería mucho menor, por lo que la calibración será completamente equivocada. De hecho, los parámetros crecerán sin control y el error siempre disminuirá. Al extender la imagen se garantizará que los puntos siempre serán tomados en cuenta, aunque obviamente el proceso tardará un poco más.

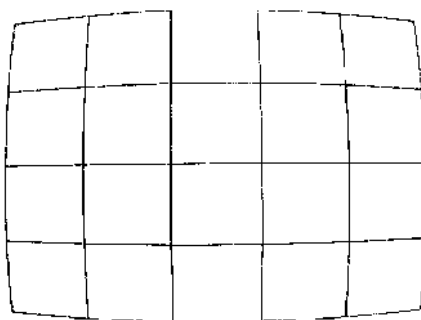


Figura 2.25: Binarización de la imagen capturada utilizando un umbral de 152 a 255

La figura 2.26(a) nos muestra al conjunto de líneas en el dominio de Hough. Sobre la vertical se tiene a θ la cual está expresada en radianes cubriendo un rango desde 0 hasta π con una resolución 0.01 radianes, lo que dará 314 renglones. Sobre la horizontal descansa

ρ desde 0 hasta $\sqrt{M^2 + N^2}$, donde M y N son el número de filas y columnas de la imagen original. Se pueden distinguir perfectamente los cúmulos que descansan aproximadamente sobre $\theta = 0$ (parte superior de la imagen) y $\theta = \pi/2$ (altura media). Los valores de ρ son variables, pero se puede inferir que los máximos locales de cada cúmulo en el dominio de Hough forman un patrón cuadrulado en el dominio de la imagen.

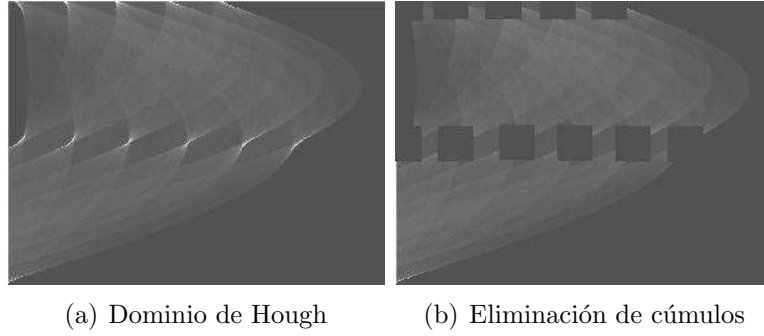


Figura 2.26: Transformada de Hough, durante el proceso de calibración.

La figura 2.26(b), es el mismo dominio de Hough, pero los cúmulos han sido eliminados. Esto ilustra el tamaño aproximado para dichos cúmulos. El cúmulo determina directamente cuales puntos serán considerados para medir la distorsión.

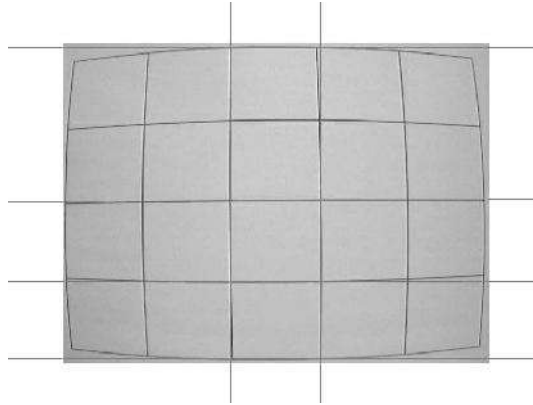


Figura 2.27: Líneas ideales

La figura 2.27 sobpone las líneas correspondientes a los máximos locales detectados por la transformada de Hough, pero en el dominio de la imagen. Conviene recordar que la implementación de la Transformada de Hough necesita de la imagen de entrada y del número de líneas a detectar. De no incluir el segundo parámetro, Hough seguramente encontrará muchas más líneas.

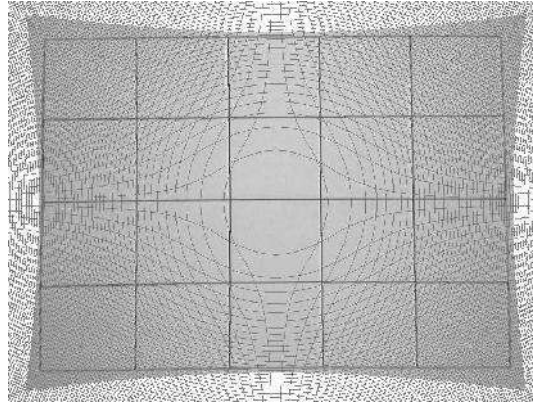


Figura 2.28: Imagen expandida

Después de aplicar el algoritmo de corrección de la distorsión se pudo obtener la figura 2.28. Las curvas nuevas que aparecen sobre la imagen, son los huecos producto de la expansión de la imagen distorsionada. El origen de los huecos se da precisamente por que la imagen crece. Además, durante el proceso de calibración, se trabaja con valores reales para la imagen de tal forma que, al graficar los resultados, se produce una pérdida de precisión debido al truncamiento originado por la discretización. Más adelante, en la sección 2.7 se abordará este tema y se establecerán algunas estrategias para eliminar los huecos.

La imagen creció lo suficiente para permitir que las líneas se enderezaran. Aquí se puede apreciar claramente el efecto del algoritmo de corrección, el cual estira la imagen desde las esquinas hacia afuera.

Siguiendo este proceso de calibración se obtuvieron los siguientes parámetros intrínsecos:

$$k_1 = 1,1 \times 10^{-6}$$

$$k_2 = -1,04 \times 10^{-13}$$

$$c_x = N/2 + 1,581$$

$$c_y = M/2 - 0,548$$

En la siguiente sección, se explicarán dos estrategias para remover los huecos de la imagen corregida. Uno de ellos está basado en la información de los vecinos de la imagen ya corregida. El otro método se basa en considerar la información de los vecinos, pero en base a la imagen original (distorsionada). Los resultados de esta última corrección se presentan al final, haciendo una comparación a detalle para apreciar las diferencias entre ambos.

2.7. Relleno de los huecos

En esta sección, describiremos dos estrategias para rellenar los huecos originados por la corrección de la distorsión. La primera de ellas está basada en la información de los vecinos de los propios huecos, ya que se utiliza la imagen que ha sido expandida. La segunda se basa en una técnica conocida como *interpolación bilineal* en donde, a diferencia de la primera, ahora se utilizará la información de la imagen distorsionada. Sin embargo, no existe un elemento en la imagen distorsionada que corresponda a cualquiera de los huecos. Por lo tanto, ahora nos enfrentamos al problema contrario de la corrección de la distorsión, es decir, tenemos que encontrar la posición a nivel subpixel, en la imagen distorsionada, que en base a los parámetros intrínsecos tendría cada hueco de la imagen corregida.

2.7.1. Problemas al expandir una imagen

Siempre que una imagen crece en sus dimensiones originales, necesariamente surge el problema de como asignar un valor a aquellos pixeles que no pudieron ser mapeados desde la imagen original.

El propósito de esta sección es mostrar alternativas para rellenar estos huecos con información aproximada que afecte lo menos posible la calidad de la imagen. Por lo tanto, el problema es como determinar que tono de gris se le deberá asignar a cada hueco.

En la sección 2.3.1 de la página 16 se plantearon las ecuaciones que nos permiten corregir la distorsión de una imagen. Dependiendo del valor que se le asigne a los parámetros de las ecuaciones 2.3 la imagen se estira o se contrae, existiendo necesariamente pérdida de información. En la figura 2.29 mostramos tal efecto: a una imagen de 100 x 100 pixeles se le aplican las ecuaciones 2.3 con parámetros de $k_1 = 0,000038$, $k_2 = 0$, $s_x = 1$ y con el centro de distorsión en el centro geométrico de la imagen. La imagen 2.29(a) se expandió a una nueva imagen 2.29(b) de 116x116 pixeles. En la expansión, 3456 pixeles se quedaron sin información. Estos pixeles son los huecos que forman una especie de malla en la figura 2.29(b).

Una de las estrategias para el rellenar los huecos, es asignarle el tono de gris del *vecino más próximo*. Esta técnica es muy simple y el único problema es determinar cual de los vecinos considerar, tomando en cuenta que posiblemente alguno de ellos sea también un hueco. Se pueden considerar dos o más vecinos, lo que puede producir mejores resultados, ya sea aplicando un proceso de interpolación o un promedio de los mismos para obtener el nuevo tono que será asignado al hueco.

Una mejor solución es usar una interpolación bilineal [6], donde en lugar de rellenar en base a los vecinos de la imagen corregida, se mapea el hueco hacia la imagen original utilizando como base las mismas ecuaciones que lo produjeron y se utiliza la información de los vecinos en la imagen original. Esta técnica trabaja a nivel subpixel produciendo

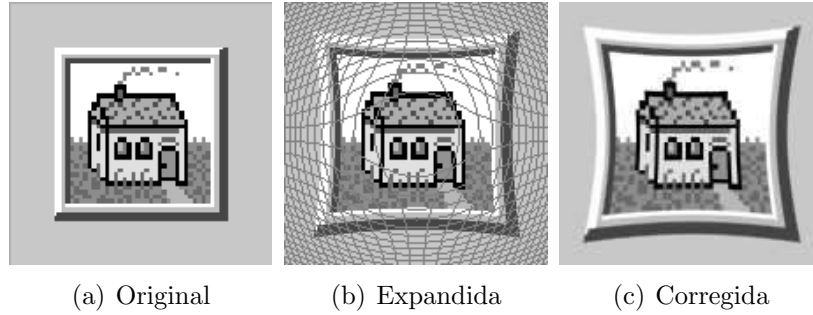


Figura 2.29: Generación de huecos al estirar una imagen.

menor pérdida de definición al expandir las imágenes. La figura 2.29(c) ilustra lo anterior. Esta técnica de interpolación bilineal será el tema a tratar en la siguiente sección.

2.7.2. Interpolación bilineal

Una interpolación bilineal, aplicada a una imagen, consiste en asignarle un valor de tono de gris a un pixel en base al tono de gris de sus cuatro vecinos.

Para rellenar los huecos de las imágenes corregidas, en este trabajo utilizamos un método de interpolación bilineal en donde, en lugar de utilizar la información de los vecinos en la imagen corregida, utilizamos información de los vecinos en la imagen original.

El problema de querer utilizar la interpolación bilineal en base a la imagen original, consistirá en mapear la imagen expandida a la imagen distorsionada. Como en general los pixeles de la imagen corregida no tienen una correspondencia con los pixeles de la imagen original distorsionada, será necesario ubicarlos a nivel subpixel en la imagen distorsionada y a partir de ésta utilizar a sus cuatro vecinos para obtener el tono de gris que asignaremos al pixel en la imagen corregida.

A continuación explicaremos más detalladamente este proceso de cálculo de las coordenadas reales (no enteras) (x_d, y_d) correspondientes a un pixel en coordenadas (x_u, y_u) de la imagen corregida.

Comenzaremos recordando las ecuaciones que modelan la distorsión:

$$x_u = (x_d - c_x)/s_x(1 + k_1 r_d^2 + k_2 r_d^4) + c_x \quad (2.13)$$

$$y_u = (y_d - c_y)(1 + k_1 r_d^2 + k_2 r_d^4) + c_y \quad (2.14)$$

donde

$$r_d^2 = (x_d - c_x)^2 + (y_d - c_y)^2 \quad (2.15)$$

En [6] se considera que con un solo componente k_1 es suficiente para casos en que existe poca distorsión, por lo que se puede excluir k_2 . Por lo tanto podemos escribir las ecuaciones 2.13 y 2.14 como:

$$x_u = x_d + (x_d - c_x)k_1r_d^2 \quad (2.16)$$

$$y_u = y_d + (y_d - c_y)k_1r_d^2 \quad (2.17)$$

Referenciando la figura 2.4 de la página 11, podemos establecer las coordenadas distorsionadas r_d en función de las coordenadas sin distorsión r_u con la siguiente ecuación:

$$r_u = r_d(1 + k_1r_d^2) \quad (2.18)$$

Así mismo, de 2.18 obtenemos:

$$r_d^3k_1 + r_d - r_u = 0$$

Por lo tanto:

$$r_d^3 + \frac{r_d}{k_1} - \frac{r_u}{k_1} = 0 \quad (2.19)$$

El cual es un polinomio de tercer grado de la forma $r_d^3 + cr_d + d = 0$, con $c = \frac{1}{k_1}$ y $d = -\frac{r_u}{k_1}$ la cual se puede resolver en forma directa a través del método de Cardano [6]. Con el método de Cardano podemos encontrar una de las tres soluciones reales dependiendo del signo del siguiente término:

$$\Delta = Q^3 + R^2$$

donde $Q = \frac{c}{3}$ y $R = -\frac{d}{2}$

Si $\Delta > 0$ la solución está dada por:

$$r_d = \sqrt[3]{R + \sqrt{\Delta}} - \frac{Q}{\sqrt[3]{R + \sqrt{\Delta}}} \quad (2.20)$$

y si $\Delta < 0$ la solución está dada por:

$$r_d = -S\cos(T) + S\sqrt{3}\text{Sen}(T) \quad (2.21)$$

donde $S = \sqrt[3]{\sqrt{R^2 - \Delta}}$ y $T = \frac{1}{3}\arctan(\frac{\sqrt{-\Delta}}{R})$

De las ecuaciones 2.16 y 2.17 tenemos:

$$k_1r_d^2 = \frac{x_u - x_d}{x_d - c_x} \quad (2.22)$$

$$k_1r_d^2 = \frac{y_u - y_d}{y_d - c_y} \quad (2.23)$$

y de la ecuación 2.18 podemos despejar:

$$k_1 r_d^2 = \frac{r_u}{r_d} - 1 \quad (2.24)$$

Igualando las ecuaciones 2.22 y 2.23 con 2.24 obtenemos:

$$\frac{x_u - x_d}{x_d - c_x} = \frac{r_u}{r_d} - 1 \quad (2.25)$$

$$\frac{y_u - y_d}{y_d - c_y} = \frac{r_u}{r_d} - 1 \quad (2.26)$$

Desarrollando paso a paso la ecuación 2.25 obtenemos:

$$\begin{aligned} x_u - x_d &= (x_d - c_x) \left[\frac{r_u}{r_d} - 1 \right] \\ x_u - x_d &= \frac{x_d r_u}{r_d} - \frac{c_x r_u}{r_d} + c_x - x_d \\ x_u - x_d &= \frac{x_d r_u}{r_d} - \frac{c_x r_u}{r_d} + \frac{c_x r_d}{r_d} - \frac{x_d r_d}{r_d} \\ x_u r_d - x_d r_d - x_d r_u + c_x r_u - c_x r_d + x_d r_d &= 0 \\ x_u r_d - x_d r_u + c_x r_u - c_x r_d &= 0 \\ x_d r_u &= c_x r_u + r_d x_u - c_x r_d \\ x_d &= \frac{c_x r_u}{r_u} + \frac{x_u r_d}{r_u} - \frac{c_x r_d}{r_u} \\ x_d &= c_x + \frac{r_d}{r_u} (x_u - c_x) \end{aligned}$$

Si hacemos un desarrollo similar para la ecuación 2.26, finalmente podemos obtener las coordenadas distorsionadas como:

$$x_d = c_x + (x_u - c_x) \frac{r_d}{r_u} \quad (2.27)$$

$$y_d = c_y + (y_u - c_y) \frac{r_d}{r_u} \quad (2.28)$$

Con estas ecuaciones estamos en posibilidad de mapear cada punto de la imagen sin distorsión (x_u, y_u) con la imagen original distorsionada (x_d, y_d) y por lo tanto aplicar la interpolación bilineal.

Por ejemplo, supongamos que $x_d = 2,16$ y $y_d = 3,31$. En la figura 2.30 se aprecian como los vecinos a considerar son $P_1(2, 3)$, $P_2(2, 4)$, $P_3(3, 3)$ y $P_4(3, 4)$.

Finalmente la interpolación se llevará a cabo en tres pasos (ver figura 2.31): primero se interpolará la información de los vecinos P_1 y P_3 con la coordenada x_d , obteniendo así el valor T_1 . De la misma forma se hará con los vecinos P_2 y P_4 , obteniendo T_2 . Llamaremos a esta etapa primera interpolación y se caracteriza por que se construye en base a tonos de gris de píxeles reales. Posteriormente se interpolará entre T_1 y T_2 , pero ahora con la coordenada y_d , para obtener el tono de gris T . Esta es la segunda interpolación y está hecha en base a dos valores interpolados. El valor de T es el tono de gris que corresponderá al hueco con coordenadas (x_u, y_u) , del cual obtuvimos las coordenadas (x_d, y_d) .

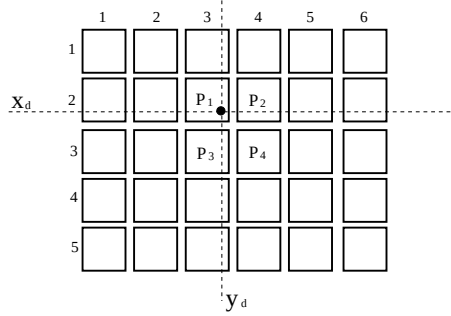


Figura 2.30: Interpolación bilineal, los vecinos en la imagen con distorsión son P_1, P_2, P_3 y P_4 mientras x_d y y_d son las coordenadas del punto que se va a interpolar.

Finalmente se muestran imágenes reales (ver figura 2.32), en donde aplicamos la interpolación bilineal y un método que aprovecha la información de los vecinos en la imagen corregida. Las diferencias se pueden apreciar al realizar un acercamiento a ambas imágenes (ver figura 2.33), observándose que la interpolación bilineal produce muchos mejores resultados tomando como referencia a la figura 2.33(c)) que muestra un acercamiento de la imagen original distorsionada. En la técnica de relleno de huecos usando información de vecinos se aprecia los efectos de la discretización, las “líneas” son menos suaves con un efecto de “escalera”.

2.8. Conclusiones

Dependiendo de las características de fabricación de los lentes éstos siempre introducirán, en menor o mayor grado, distorsión en las imágenes. El método de autocalibración nos ha permitido corregir dicha distorsión al poder identificar los parámetros intrínsecos de nuestras cámaras. En este capítulo hemos utilizado la Transformada de Hough para encontrar los valores de estos parámetros que nos permitan enderezar las líneas distorsionadas en una imagen. Sin embargo es necesario binarizar las imágenes para que este método pueda funcionar, lo que agrega costo al tiempo de procesamiento. De igual forma, al obtener las imágenes corregidas, se hace necesario rellenar los huecos producto de la corrección, en tal caso hemos encontrado en la interpolación bilineal una herramienta suficiente para este trabajo.

Con esto finalizamos el capítulo de corrección de la distorsión, utilizando el método de Autocalibración. A continuación dedicaremos un capítulo más a este tema, pero ahora utilizando un método basado en técnicas de registro de imágenes.

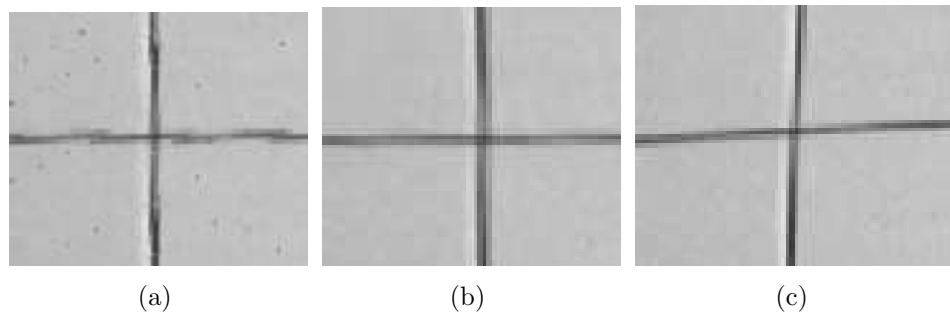


Figura 2.33: Detalles del relleno de huecos utilizando (a) Información de los vecinos de la imagen corregida, (b) Interpolación bilineal y (c) Imagen original distorsionada (ver figura 2.23(b)) de la página 31.

Capítulo 3

Corrección de la Distorsión utilizando Técnicas de Registro de Imágenes.

3.1. Introducción

El registro de imágenes es el proceso de empatar dos imágenes, que describen un mismo escenario, aplicándole a una de ellas un conjunto de transformaciones espaciales. La imagen original es aquella que servirá de referencia a la imagen a la cual se le aplicarán las transformaciones espaciales que en nuestro caso llamaremos imagen de trabajo. La imagen de trabajo con respecto a la imagen original puede encontrarse trasladada, girada, escalada, o incluso puede tener transformaciones más complejas como las producidas por la distorsión radial o por transformaciones proyectivas.

En este capítulo utilizamos la técnica de registro de imágenes para encontrar los parámetros de calibración de una cámara que nos permitan corregir su distorsión. Al igual que en el capítulo 2, se elaborará un patrón sintético el cual será capturado por la cámara para obtener la imagen de trabajo. La imagen original será el propio patrón sintético. Los procedimientos utilizados en esta sección están basados en [21] y [17].

En la siguiente sección estableceremos una función de error para medir las diferencias entre dos imágenes. Más adelante se utilizará el Método de Newton para resolver el problema de minimizar la función de error. Minimizar la función de error nos permitirá aproximarnos a solucionar el problema de registro de imágenes.

3.2. Función de error

Supongamos que tenemos dos imágenes, I_1 e I_2 , a las cuales llamaremos imagen de trabajo e imagen original respectivamente, representan una misma escena pero con algunas diferencias geométricas que pudieran ser originadas, por ejemplo, por una transformación

espacial (ver apéndice A).

Podemos plantear el empate de las dos imágenes en base a una función de error $e(r)$, la cual se encargará de medir las diferencias, pixel a pixel, entre ambas imágenes. Decimos que las imágenes estarán perfectamente empatadas cuando $e(r) = 0$.

Para empatar ambas imágenes, suponiendo que esto se pueda llevar a cabo, necesitaremos trabajar con un conjunto de parámetros que nos permitan ajustar la posición de los píxeles de una de ellas hasta obtener el mejor empate. Es decir, aplicaremos un conjunto de transformaciones espaciales a una de las imágenes buscando minimizar una función que mida las diferencias entre ambas.

Podemos establecer una medida de error como sigue:

$$e(r) = (I_1(\hat{r}) - I_2(r))^2 \quad (3.1)$$

En donde r representa la posición (x, y) para un pixel en particular y $e(r)$ es la diferencia al cuadrado entre los píxeles (x, y) de una de las imágenes y $(\hat{x}, \hat{y})$ de la otra. La presencia del término $I_1(\hat{r})$ indica que al punto $r = (x, y)$ de la imagen I_1 se le ha aplicado una transformación espacial.

Por ejemplo, si fuera suficiente empatar dos imágenes con una transformación afín (ver apéndice A.6), podríamos definir la transformación $\hat{r}$ como sigue:

$$\hat{r} = \begin{bmatrix} x & y & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & x & y & 1 \end{bmatrix} \begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \\ \theta_3 \\ \theta_4 \\ \theta_5 \end{bmatrix} \quad (3.2)$$

En donde los parámetros $(\theta_1, \theta_2, \theta_3, \theta_4$ y $\theta_5)$ describen la traslación, la rotación y el escalamiento, a través de los cuales se busca empatar las dos imágenes.

De esta forma el error total $E(\theta)$, donde $\theta = \theta_1, \theta_2, \dots, \theta_5$ y considerando que es suficiente una transformación afín para empatar las imágenes, puede expresarse de la siguiente forma:

$$E(\theta) = \sum_{r \in L} (I_1(\hat{r}(\theta)) - I_2(r))^2 \quad (3.3)$$

En donde L establece los puntos válidos dentro de las dimensiones de la imagen. Con la ecuación 3.3, ahora el problema es encontrar aquellos valores para θ que minimicen la función de error $E(\theta)$,

Sin embargo, cuando se trata de imágenes adquiridas por cámaras, es natural pensar en que otro tipo de transformaciones serán necesarias, como por ejemplo la transformación proyectiva. Aún más, todas las cámaras introducen distorsión la cual deberá también ser considerada.

En las siguientes dos secciones consideraremos la transformación proyectiva y la corrección de la distorsión, por lo cual será necesario agregar siete parámetros más al modelo de la transformación afín. Necesitaremos dos parámetros más para la transformación proyectiva y cinco para la corrección de la distorsión. En total estaremos trabajando con trece parámetros y en algunos casos, cuando la distorsión sea más pronunciada, con catorce.

3.3. Transformación Proyectiva

En esta sección se agregarán dos parámetros más a la ecuación 3.3, con los cuales estaremos considerando la deformación geométrica que se presenta en las imágenes, al ser capturadas por una cámara, producto de una transformación proyectiva.

Una transformación proyectiva transforma un plano P en otro plano P' equivalente (ver figura 3.1) y en el cual se conservan algunas de las características del primero, aunque otras se pierden (ver demostración en [20]). Una de las características que se conservan es la de colinealidad, es decir, las líneas que son rectas en P seguirán siendo rectas en P' . Sin embargo las líneas que eran paralelas no necesariamente lo seguirán siendo en la proyección P' .

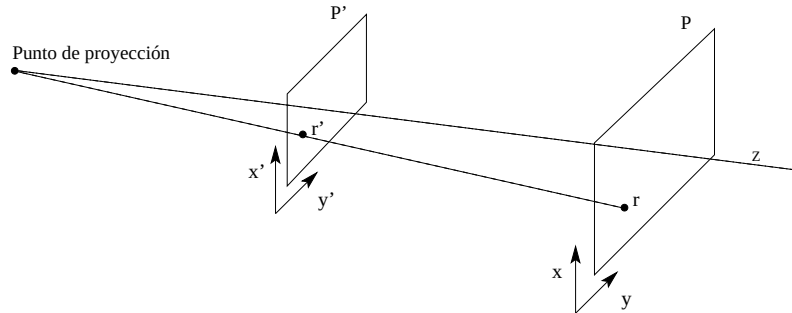


Figura 3.1: Ejemplo de proyección

En la figura 3.1 se muestran dos planos P y P' que comparten el mismo eje de proyección Z y en donde idealmente los dos planos son perpendiculares a dicho eje. De la misma forma se muestra como r es proyectado en r' , utilizando como referencia un punto llamado *punto de proyección*. Este tipo de proyección es conocido como *proyección en perspectiva*, en el cual, las líneas paralelas en una escena que no son paralelas al plano de la imagen, se proyectan en líneas convergentes. Este tipo de proyección hace parecer a las imágenes más reales ya que esta es la manera en que nuestros ojos y la lente de una cámara forman las imágenes [9].

En la figura 3.2 se ilustra como en la proyección en perspectiva, se transforman las posiciones de los objetos en el plano de la imagen a lo largo de líneas que convergen en

un punto que se denomina punto de referencia o centro de proyección [9]. En nuestro caso el centro de proyección es el punto focal y la imagen que se forma es resultado de la intersección entre las líneas de proyección y el plano de la imagen.

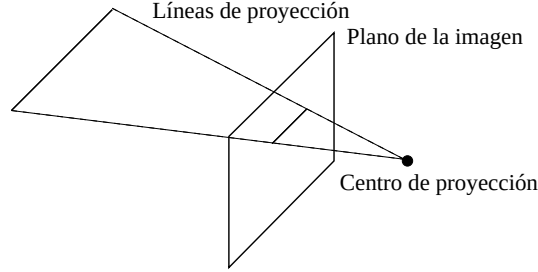


Figura 3.2: Proyección en perspectiva

Por lo tanto, considerando que nuestro objetivo es empatar dos imágenes, en donde una de ellas es una proyección de la otra, y considerando que al momento de la adquisición de la imagen difícilmente el plano de la imagen coincidió con el plano de la imagen frente a la cámara, será necesario encontrar los parámetros que nos permitan corregir este problema.

Para el caso de una transformación proyectiva se hace necesario extender los procedimientos de una transformación afín, ya que ahora estarán en juego ocho parámetros y no seis [3][20]. La ecuación 3.4 describe la transformación proyectiva expresada en coordenadas homogéneas (ver apéndice A).

$$\begin{bmatrix} x' \\ y' \\ h \end{bmatrix} = \begin{bmatrix} \theta_0 & \theta_1 & \theta_2 \\ \theta_3 & \theta_4 & \theta_5 \\ \theta_6 & \theta_7 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (3.4)$$

En el apéndice A se describe la importancia de trabajar con coordenadas homogéneas, por lo que será necesario dividir todo el sistema entre un mismo factor $\theta_6x + \theta_7y + 1$ para que h siga siendo la unidad. De esta forma podemos establecer las siguientes ecuaciones:

$$\begin{aligned} \hat{x} &= \frac{\theta_0x + \theta_1y + \theta_2}{\theta_6x + \theta_7y + 1} \\ \hat{y} &= \frac{\theta_3x + \theta_4y + \theta_5}{\theta_6x + \theta_7y + 1} \end{aligned} \quad (3.5)$$

Ahora, de la misma forma como hemos considerado la transformación proyectiva, también tomaremos en cuenta la distorsión radial introducida por la lente de las cámaras. Por lo tanto, en la siguiente sección quedará completo el modelo para resolver el problema de registro, en donde podremos llevar a cabo uno de los objetivos de este capítulo y de esta tesis, la corrección de la distorsión.

3.4. Corrección de la distorsión

En esta sección agregaremos 5 parámetros más a la ecuación 3.5, con lo cual consideraremos la corrección de la distorsión como parte del problema de registro.

En la sección 2.3.1 de la página 16, ya obtuvimos un modelo para corregir la distorsión generada por la lente de una cámara. Básicamente el procedimiento fue manipular 5 variables para ir moldeando un patrón de líneas previamente adquirido. De esta forma pudimos obtener la ecuación 2.3 de la página 17, en la cual podemos apreciar los 5 parámetros en juego: c_x, c_y, k_1, k_2 y s_x . A diferencia de la corrección de la distorsión usando autocalibración, en donde la idea es que las curvas se hagan rectas, ahora generaremos un modelo que nos permita llevar una imagen distorsionada I_1 , a un patrón ideal sintético I_2 (ver figura 2.8 de la página 17). Este proceso nos permitirá obtener los 13 parámetros necesarios para este fin: 6 para el modelo afín, 2 para la transformación proyectiva y 5 más para la corrección de la distorsión.

Reagrupando las ecuaciones 2.3 (de la página 17) y 3.5 utilizadas para la corrección de la distorsión y la transformación proyectiva respectivamente, podemos establecer las siguiente nomenclatura para facilitar las expresiones subsecuentes:

$$r' = D(r) = \left\{ \begin{array}{l} x' = \frac{(x-c_x)}{s_x}(1 + k_1 r_d^2 + k_2 r_d^4) + c_x \\ y' = (y - c_y)(1 + k_1 r_d^2 + k_2 r_d^4) + c_y \end{array} \right\} \quad (3.6)$$

$$r'' = P(r') = \left\{ \begin{array}{l} x'' = \frac{\theta_0 x' + \theta_1 y' + \theta_2}{\theta_6 x' + \theta_7 y' + 1} \\ y'' = \frac{\theta_3 x' + \theta_4 y' + \theta_5}{\theta_6 x' + \theta_7 y' + 1} \end{array} \right\} \quad (3.7)$$

En donde $D(r)$ es una transformación para corregir la distorsión con 5 parámetros y está en función directamente de x y y , mientras que $P(r')$ es una transformación proyectiva de 8 parámetros y está en función de r' , es decir de $D(r)$.

Por lo tanto la función de error a minimizar es:

$$E(\theta) = \sum_{i=1}^{i=N} [I_1(P(D(r_i))) - I_2(r_i)]^2 \quad (3.8)$$

En donde intervienen tanto la transformación proyectiva como la función para corregir la distorsión.

Ahora el siguiente paso será llevar a cabo la optimización de la función de error, es decir, encontrar los parámetros $\theta_1, \theta_2, \theta_3, \dots, \theta_{13}$ que minimizen $E(\theta)$.

3.5. Optimización de la función de error

En esta sección se describe un proceso completo de optimización llamado método de Gauss-Newton-Levenberg-Marquardt. El proceso de optimización ha sido extraído del

artículo de Romero y Calderón [17] y se ha ampliado la descripción de cada una de las etapas que intervienen en él.

3.5.1. Método de Gauss-Newton-Levenberg-Marquardt

El método de Gauss-Newton-Levenberg-Marquardt (GNLM), es una variante del método de Newton (ver el apéndice B.3).

En el método de GNLM al momento de calcular un nuevo valor de θ se utilizan los incrementos de la misma, $\delta\theta$, para resolver la siguiente ecuación:

$$A\delta\theta = B \quad (3.9)$$

donde $A = [J^T J + \lambda I]$ y $B = -J^T e$

De tal forma que si la imagen está formada por p pixeles y q es el número de parámetros, en este caso $q = 13$, entonces A es una matriz de tamaño $q \times q$.

J es una matriz de $p \times q$ y respresenta el Jacobiano de e (ver apéndice B.3).

I es la matriz identidad y e es un vector que contiene todas las diferencias entre los pixeles de ambas imágenes y tiene una dimensión de $p \times 1$.

Por último λ es un parámetro que va variando en cada iteración.

El proceso completo de calibración utilizando el método de GNLM es como sigue:

1. Dar un valor inicial para θ^0
2. Procesar el error total, $E_t(I_1(\theta), I_2)$
3. Fijar un valor pequeño para λ , por ejemplo $\lambda = 0.001$
4. Aplicar interpolación bilineal a $I_1(\theta)$ para mejorar su calidad.
5. Resolver el sistema de ecuaciones 3.9 y calcular $E_t(I_1(\theta + \delta\theta), I_2)$
6. Si $E_t(I_1(\theta + \delta\theta), I_2) > E_t(I_1(\theta), I_2)$, incrementar λ por un factor de 10 y regresar al paso anterior. Si λ se hace muy grande querrá decir que no existe forma para mejorar la solución θ .
7. Si $E_t(I_1(\theta + \delta\theta), I_2) < E_t(I_1(\theta), I_2)$, decrementar λ por un factor de 10, reemplazar θ por $\theta + \lambda\theta$ y regresar al paso 2.

Cuando $\lambda = 0$, el método de GNLM es el método de Newton.

Básicamente los elementos que hay que calcular son el conjunto de Jacobianos $J_i(\theta)$. A continuación se definirán las ecuaciones necesarias para llevar a cabo el método de GNLM.

3.5.2. Definición de los Jacobianos

Considerando que $J_i(\theta)$ es el Jacobiano para uno de los P pixeles entonces el elemento $J_i(\theta)$ puede ser expresado de la siguiente forma:

$$J_i(\theta) = \frac{\partial e_i(\theta)}{\partial \theta} \quad (3.10)$$

$$(3.11)$$

En donde:

$$\theta = \left\{ \begin{array}{ll} \theta_0, \dots, \theta_7 & = \text{(TransformaciónProyectiva)} \\ \theta_8, \dots, \theta_{12} & = (k_1, k_2, c_x, c_y, s_x) \end{array} \right\}$$

y

$$e_i(\theta) = I_1(r_i'') - I_2(r_i)$$

Ahora calcularemos $J_i(\theta_0)$

$$J_i(\theta_0) = \frac{\partial e_i(\theta)}{\partial \theta_0} = \frac{\partial I_1(r'')}{\partial x''} \frac{\partial x''}{\partial \theta_0} + \frac{\partial I_1(r'')}{\partial y''} \frac{\partial y''}{\partial \theta_0} \quad (3.12)$$

Como $\frac{\partial y''}{\partial \theta_0} = 0$ entonces nos queda:

$$J_i(\theta_0) = \frac{\partial I_1(r'')}{\partial x''} \frac{\partial x''}{\partial \theta_0} \quad (3.13)$$

Sustituyendo 3.7 para x'' en 3.13:

$$J_i(\theta_0) = \frac{\partial I_1(r'')}{\partial x''} \frac{\partial \frac{\theta_0 x' + \theta_1 y' + \theta_2}{\theta_6 x' + \theta_7 y' + 1}}{\partial \theta_0} \quad (3.14)$$

Derivando 3.14 con respecto a θ_0 finalmente obtenemos:

$$J_i(\theta_0) = \frac{\partial I_1(r'')}{\partial x''} \frac{x'}{\theta_6 x' + \theta_7 y' + 1} \quad (3.15)$$

Ahora si consideramos $G' = \theta_6 x' + \theta_7 y' + 1$ y repetimos el procedimiento para $\theta_1, \dots, \theta_7$ finalmente obtendremos las siguientes ecuaciones:

$$\begin{aligned}
J_i(\theta_0) &= \frac{\partial I_1(r'')}{\partial x''} \frac{x'}{G'} \\
J_i(\theta_1) &= \frac{\partial I_1(r'')}{\partial x''} \frac{y'}{G'} \\
J_i(\theta_2) &= \frac{\partial I_1(r'')}{\partial x''} \frac{1}{G'} \\
J_i(\theta_3) &= \frac{\partial I_1(r'')}{\partial y''} \frac{x'}{G'} \\
J_i(\theta_4) &= \frac{\partial I_1(r'')}{\partial y''} \frac{y'}{G'} \\
J_i(\theta_5) &= \frac{\partial I_1(r'')}{\partial y''} \frac{1}{G'} \\
J_i(\theta_6) &= \frac{\partial I_1(r'')}{\partial x''} \frac{-x'x''}{G'} + \frac{\partial I_1(r'')}{\partial y''} \frac{-x'y''}{G'} \\
J_i(\theta_7) &= \frac{\partial I_1(r'')}{\partial x''} \frac{-y'x''}{G'} + \frac{\partial I_1(r'')}{\partial y''} \frac{-y'y''}{G'}
\end{aligned} \tag{3.16}$$

Ahora haremos lo mismo para $\theta_8, \dots, \theta_{12}$ recordando que:

$\theta_8 = k_1$ (Primer coeficiente de distorsión)
 $\theta_9 = k_2$ (Segundo coeficiente de distorsión)
 $\theta_{10} = c_x$ (centro de distorsión en x)
 $\theta_{11} = c_y$ (centro de distorsión en y)
 $\theta_{12} = s_x$ (Factor de escala)

Para θ_0 :

$$J(\theta_8) = \frac{\partial I_1(r'')}{\partial x''} \frac{\partial x''}{\partial k_1} + \frac{\partial I_1(r'')}{\partial y''} \frac{\partial y''}{\partial k_1} \tag{3.17}$$

Ahora para $\frac{\partial x''}{\partial k_1}$ y $\frac{\partial y''}{\partial k_1}$ volvemos a derivar:

$$\begin{aligned}
\frac{\partial x''}{\partial k_1} &= \frac{\partial x''}{\partial x'} \frac{\partial x'}{\partial k_1} + \frac{\partial x''}{\partial y'} \frac{\partial y'}{\partial k_1} \\
\frac{\partial y''}{\partial k_1} &= \frac{\partial y''}{\partial x'} \frac{\partial x'}{\partial k_1} + \frac{\partial y''}{\partial y'} \frac{\partial y'}{\partial k_1}
\end{aligned} \tag{3.18}$$

Podemos representar a la ecuación 3.18 en forma matricial como:

$$\begin{bmatrix} \frac{\partial x''}{\partial k_1} \\ \frac{\partial y''}{\partial k_1} \end{bmatrix} = \begin{bmatrix} \frac{\partial x''}{\partial x'} & \frac{\partial x''}{\partial y'} \\ \frac{\partial y''}{\partial x'} & \frac{\partial y''}{\partial y'} \end{bmatrix} \begin{bmatrix} \frac{\partial x'}{\partial k_1} \\ \frac{\partial y'}{\partial k_1} \end{bmatrix}$$

Entonces de 3.17 y 3.18 podemos deducir que:

$$J_i(\theta_8 = k_1) = \begin{bmatrix} \frac{\partial x''}{\partial k_1} & \frac{\partial y''}{\partial k_1} \end{bmatrix} \begin{bmatrix} \frac{\partial I_1(r'')}{\partial x''} \\ \frac{\partial I_1(r'')}{\partial y''} \end{bmatrix} = \begin{bmatrix} \frac{\partial r''}{\partial k_1} \end{bmatrix}^T \nabla I_1(r'')$$

y ya que $\frac{\partial r''}{\partial k_1} = \frac{\partial r''}{\partial r'} \frac{\partial r'}{\partial k_1}$ entonces:

$$J_i(\theta_8 = k_1) = \begin{bmatrix} \frac{\partial r'}{\partial k_1} \end{bmatrix}^T \begin{bmatrix} \frac{\partial r''}{\partial r'} \end{bmatrix}^T \nabla I_1(r'') \quad (3.19)$$

Por lo tanto considerando $\theta_8, \dots, \theta_{12}$ la ecuación 3.19 expresada como producto matricial queda de la siguiente forma:

$$J_i(\theta_8, \dots, \theta_{12}) = \begin{bmatrix} \frac{\partial x'}{\partial k_1} & \frac{\partial y'}{\partial k_1} \\ \frac{\partial x'}{\partial k_2} & \frac{\partial y'}{\partial k_2} \\ \frac{\partial x'}{\partial c_x} & \frac{\partial y'}{\partial c_x} \\ \frac{\partial x'}{\partial c_y} & \frac{\partial y'}{\partial c_y} \\ \frac{\partial x'}{\partial s_x} & \frac{\partial y'}{\partial s_x} \end{bmatrix} \begin{bmatrix} \frac{\partial x''}{\partial x'} & \frac{\partial y''}{\partial x'} \\ \frac{\partial x''}{\partial y'} & \frac{\partial y''}{\partial y'} \end{bmatrix} \begin{bmatrix} \frac{\partial I_1(r'')}{\partial x''} \\ \frac{\partial I_1(r'')}{\partial y''} \end{bmatrix} \quad (3.20)$$

Ahora, resolviendo $\begin{bmatrix} \frac{\partial r''}{\partial r'} \end{bmatrix}^T$ de 3.20 obtenemos:

$$\begin{aligned} \frac{\partial x''}{\partial x'} &= \frac{G'\theta_0 + \theta_6(\theta_0 x' + \theta_1 y' + \theta_2)}{(G')^2} \\ \frac{\partial x''}{\partial y'} &= \frac{G'\theta_1 + \theta_7(\theta_0 x' + \theta_1 y' + \theta_2)}{(G')^2} \\ \frac{\partial y''}{\partial x'} &= \frac{G'\theta_3 + \theta_6(\theta_3 x' + \theta_4 y' + \theta_5)}{(G')^2} \\ \frac{\partial y''}{\partial y'} &= \frac{G'\theta_4 + \theta_7(\theta_3 x' + \theta_4 y' + \theta_5)}{(G')^2} \end{aligned} \quad (3.21)$$

Y por último resolvemos $\begin{bmatrix} \frac{\partial r'}{\partial (\theta_8, \dots, \theta_{12})} \end{bmatrix}^T$ para obtener:

$$\begin{aligned}
\frac{\partial x'}{\partial k_1} &= \frac{r^2(x-c_x)}{s_x} \\
\frac{\partial y'}{\partial k_1} &= r^2(y-c_y) \\
\frac{\partial x'}{\partial k_2} &= \frac{r^4(x-c_x)}{s_x} \\
\frac{\partial y'}{\partial k_3} &= r^4(y-c_y) \\
\frac{\partial x'}{\partial c_x} &= 1 - \left(\frac{1}{s_x}\right)(1 + k_1 r^2 + k_2 r^4) - \frac{2(k_1+2k_2 r^2)(x-c_x)^2}{s_x^3} \\
\frac{\partial y'}{\partial c_x} &= -\frac{2(k_1+2k_2 r^2)(x-c_x)(y-c_y)}{s_x^2} \\
\frac{\partial x'}{\partial c_y} &= -\frac{2(k_1+2k_2 r^2)(x-c_x)(y-c_y)}{s_x} \\
\frac{\partial y'}{\partial c_y} &= 1 - (1 + k_1 r^2 + k_2 r^4) - 2(y-c_y)^2(k_1 + 2k_2 r^2) \\
\frac{\partial x'}{\partial s_x} &= \frac{-(x-c_x)(1+k_1 r^2+k_2 r^4)}{s_x^2} - \frac{2(k_1+2k_2 r^2)(x-c_x)^3}{s_x^4} \\
\frac{\partial y'}{\partial s_x} &= -\frac{2(k_1+2k_2 r^2)(y-c_y)(x-c_x)}{s_x^3}
\end{aligned} \tag{3.22}$$

Ahora el método de GNLM puede ser aplicado para minimizar la función de error 3.8, y resolver $\theta = (\theta_0, \dots, \theta_{12})$.

Adicionalmente, de la misma forma que lo hacen en [17], aplicaremos una estrategia de optimización por puntos de correspondencia, la cual es utilizada para mejorar los resultados obtenidos con el método de GNLM.

El método de puntos de correspondencia está basado en la idea de que, después de haber aplicado el GNLM, las imágenes estarán casi empatadas. De tal forma que si se logran encontrar puntos característicos, a nivel subpixel, en las imágenes previamente empatadas, los pares correspondientes de puntos pueden ser utilizados para un segundo proceso del GNLM. En la siguiente sección se amplían los detalles sobre este método.

3.6. Calibración por Puntos de Correspondencia

Este método es utilizado en [17] intentando mejorar los resultados obtenidos con el método de GNLM. La idea básica es encontrar características comunes y fáciles de identificar tanto en la imagen patrón I_2 (y que ahora referenciamos como I_r), como en la imagen proyectada, y ya corregida con el método de GNLM, $I_1(\theta)$ (que ahora llamaremos I_p). En [17] se utilizan las esquinas como características comunes, ya que éstas generalmente son fáciles de encontrar. Por lo tanto, el primer paso es encontrar

dichas esquinas en I_r y después buscar las mismas en I_p . En el apéndice C se explica una estrategia para detectar esquinas en base al gradiente de la imagen.

3.6.1. Método de puntos de correspondencia

Si n es el número de esquinas encontradas en I_r con coordenadas (x_{ri}, y_{ri}) y (x_i, y_i) sus correspondencias en I_p , entonces utilizando las ecuaciones 3.6 y 3.7 de la página 47, a partir de (x_i, y_i) podemos calcular las coordenadas (x_{pi}, y_{pi}) de las mismas características pero en la imagen de la cámara I_d (I_1). Se utiliza la siguiente notación para dicho cálculo:

$$\begin{aligned} x_{pi} &= f_x^{pd}(\theta, x = x_i, y = y_i) \\ y_{pi} &= f_y^{pd}(\theta, x = x_i, y = y_i) \\ (x_{pi}, y_{pi}) &= f^{pd}(\theta, x = x_i, y = y_i) \end{aligned}$$

Por lo tanto obtendremos un conjunto de pares de puntos

$$P = \{< (x_{r1}, y_{r1}), (x_{p1}, y_{p1}) >, \dots, < (x_{rn}, y_{rn}), (x_{pn}, y_{pn}) >\} \quad (3.23)$$

El objetivo es encontrar el conjunto de parámetros θ de tal forma que la suma D_t , del cuadrado de las distancias entre los puntos $f^{pd}(\theta, x_{ri}, y_{ri})$ y (x_{pi}, y_{pi}) sea un mínimo:

$$\theta = \operatorname{argmin} D_t(\theta, P) = \operatorname{argmin} \sum_{i=1}^{i=n} (f_x^{pd}(\theta, x_{ri}, y_{ri}) - x_{pi})^2 + (f_y^{pd}(\theta, x_{ri}, y_{ri}) - y_{pi})^2 \quad (3.24)$$

Finalmente se lleva a cabo nuevamente una optimización no lineal utilizando el método de GNLM para minimizar D_t , pero en esta ocasión, los elementos de la matriz A y B de la ecuación 3.9 están dados por:

$$\begin{aligned} a_{i,j} &= \sum_{i=1}^{i=n} \left(\frac{\partial x_p}{\partial \theta_i} \frac{\partial x_p}{\partial \theta_j} + \frac{\partial y_p}{\partial \theta_i} \frac{\partial y_p}{\partial \theta_j} \right) \\ b_i &= - \sum_{i=1}^{i=n} \frac{\partial x_p}{\partial \theta_i} d_{xi} + \frac{\partial y_p}{\partial \theta_i} d_{yi} \\ d_{xi} &= f_x^{pd}(\theta, x_{ri}, y_{ri} - x_{pi}) \\ d_{yi} &= f_y^{pd}(\theta, x_{ri}, y_{ri} - y_{pi}) \end{aligned} \quad (3.25)$$

A continuación presentaremos los resultados obtenidos utilizando el método de GNLM y la optimización por puntos de correspondencia para resolver el problema de registro de imágenes.

3.7. Resultados obtenidos con registro de imágenes

En esta sección presentamos los resultados obtenidos al utilizar el registro de imágenes para resolver el problema de la corrección de la distorsión.

En los experimentos se utilizaron lentes de 0.21cm de distancia focal así como lentes de 0.4cm. La diferencia entre uno y otro se puede ver en la figura 3.3, en donde la primera imagen corresponde al patrón original, la imagen del centro corresponde al lente de 0.4cm, el cual presenta menor distorsión radial que la imagen de la derecha, la cual fue adquirida con la lente de 0.21cm.

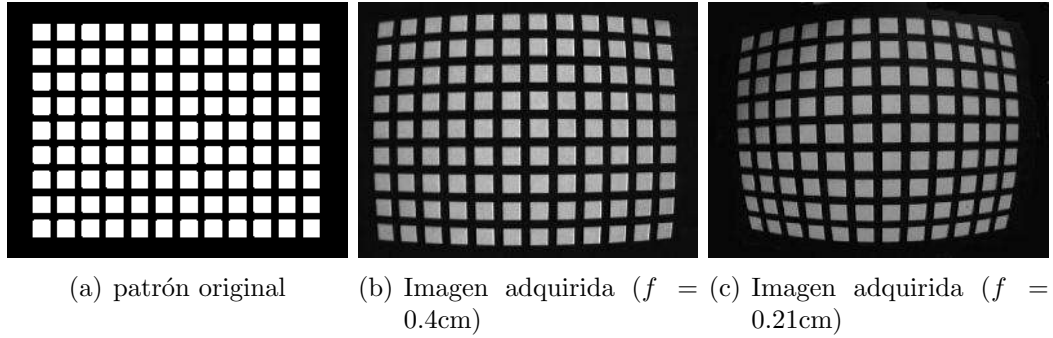


Figura 3.3: Imágenes utilizadas durante la aplicación del método de registro

El comportamiento del método de GNLM utilizando 13 parámetros fue diferente con la lente de 0.21cm que con la de 0.4cm, y con los mismos valores iniciales θ^0 . La figura 3.4 muestra que el método de GNLM obtiene una calibración muy favorable para el lente de 0.4cm, mientras que con la lente de 0.21 no tiene el mismo éxito.

Los valores iniciales θ^0 fueron los siguientes:

$$\begin{aligned}
 \theta_0 &= 1, & \theta_1 &= 0, & \theta_2 &= 0, \\
 \theta_3 &= 0, & \theta_4 &= 1, & \theta_5 &= 0, \\
 \theta_6 &= 0, & \theta_7 &= 0, & \theta_8 &= 0, \\
 \theta_9 &= 0, & \theta_{10} &= 240, & \theta_{11} &= 320, \\
 \theta_{12} &= 1
 \end{aligned}$$

En donde $\theta_8 = k_1$, $\theta_9 = k_2$, $\theta_{10} = c_x$, $\theta_{11} = c_y$ y $\theta_{12} = s_x$.

Una forma de lograr que el método de GNLM obtenga un buen resultado con lentes de más apertura (por ejemplo el de 0.21cm), es ajustando θ^0 a un valor θ cercano al buscado.

Otra alternativa, y con la cual obtuvimos buenos resultados, es agregar un tercer coeficiente de distorsión. De esta forma podemos generar un modelo menos sensible a los valores iniciales θ_0 .

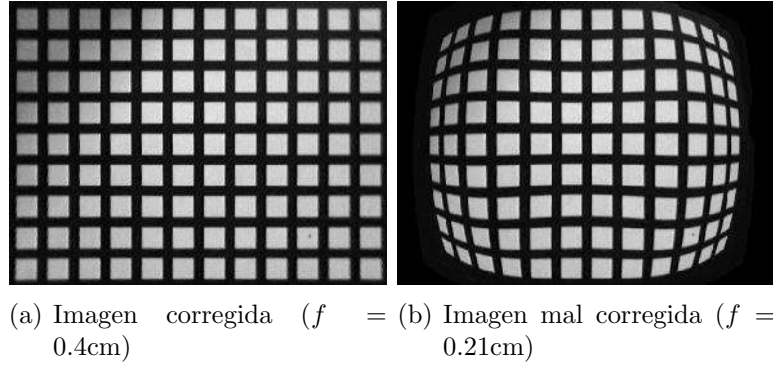


Figura 3.4: Imágenes corregidas con el método de GNLM, (a) La corrección es satisfactoria para la lente de 0.4cm, mientras que en (b) la corrección deja mucho de desear para la lente de 0.21 cm, sobre todo en las zonas de máxima distorsión que son cerca de las esquinas de la imagen.

Al utilizar un tercer coeficiente de distorsión k_3 , la ecuación 3.6 se altera de la siguiente forma:

$$r' = D(r) = \left\{ \begin{array}{l} x' = \frac{(x-c_x)}{s_x} (1 + k_1 r_d^2 + k_2 r_d^4 + k_3 r_d^6) + c_x \\ y' = (y - c_y) (1 + k_1 r_d^2 + k_2 r_d^4 + k_3 r_d^6) + c_y \end{array} \right\} \quad (3.26)$$

Ahora se hace necesario recalcular las ecuaciones 3.22, ya que estamos agregando un parámetro más, por lo que ahora el vector θ esta integrado de 14 elementos. El valor inicial para $\theta_{13} = k_3$, al aplicar el método de GNLM, es también cero.

En la figura 3.5 se muestran las imágenes procesadas para la calibración de una de las cámaras que utilizamos durante este proyecto.

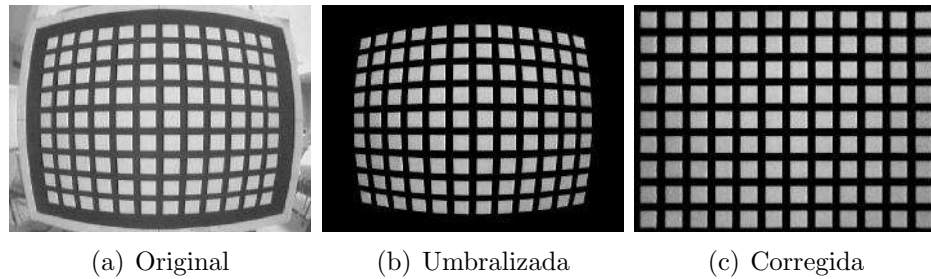


Figura 3.5: Imágenes corregidas con el método de GNLM utilizando 14 parámetros y lente de 0.21cm, se muestra la toma original, la imagen umbralizada y por último la imagen corregida.

La siguiente tabla muestra los resultados del método de GNLM utilizando tres cámaras de características similares, con una lente de 0.21cm.

Resultados del GNLM			
θ	Cámara Superior	Cámara Izquierda	Cámara Derecha
θ_0	9.69685754E-01	9.20325618E-01	9.86838243E-01
θ_1	1.84623182E-02	-1.15999145E-02	-1.95223106E-02
θ_2	1.74961381E+00	1.62436686E+01	6.34774171E+00
θ_3	-1.44484675E-02	-2.71394002E-02	2.38153866E-02
θ_4	1.00252124E+00	9.20921225E-01	9.39346687E-01
θ_5	-6.81573185E-01	1.68634466E+01	9.60934249E+00
θ_6	-4.80994769E-05	-8.79139353E-05	9.60934249E+00
θ_7	8.31396089E-05	-4.89520922E-05	-7.60156670E-05
k_1	-2.03221665E-06	-2.09068624E-06	-2.16375911E-06
k_2	5.47811303E-12	5.63059731E-12	6.69052537E-12
k_3	-9.71716976E-18	-8.50644517E-18	-1.32139897E-17
c_x	2.35545925E+02	2.18624223E+02	2.57112123E+02
c_y	3.44318824E+02	3.11888104E+02	3.06148744E+02
s_x	1.00705380E+00	1.00558047E+00	1.00744102E+00

Por su distribución en el robot, las cámaras son identificadas como: cámara superior, cámara izquierda y cámara derecha (ver figura:3.6).

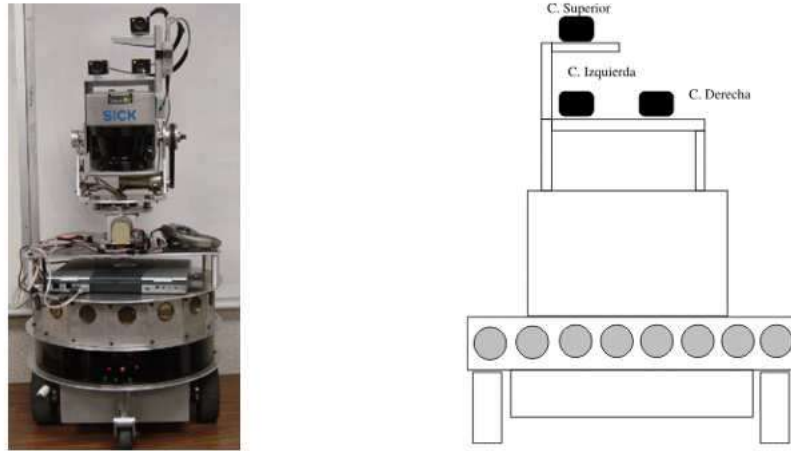


Figura 3.6: Imagen del Robot real y esquema de la vista trasera del robot. En ambas imágenes se puede apreciar la Distribución de las 3 cámaras utilizadas en el robot.

Ahora en la figura 3.7 mostramos un ejemplo de corrección de imágenes utilizando el método de GNLM con los 14 parámetros, la imagen corresponde al Laboratorio de Instrumentación y Control en el edificio de Posgrado de Ingeniería Eléctrica (UMSNH).

Finalmente hacemos una comparación entre el método de calibración original presentado en [17] que utiliza 13 parámetros y la extensión a 14 parámetros realizada en esta

investigación. El método de calibración calcula el error total producto de la diferencia pixel a pixel entre las imágenes comparadas. Los errores utilizando 13 parámetros fueron de 18931 y 17113 para los lentes de 0.4 y 0.21 respectivamente. De la misma forma, utilizando 14 parámetros se obtuvieron errores de 18815 y 17021 para los lentes de 0.4 y 0.21 respectivamente.

Sin embargo, una medida de error más significativa se lleva a cabo obteniendo la máxima diferencia entre los pares de puntos $f^{pd}(\theta, x_{ri}, y_{ri})$ y (x_{pi}, y_{pi}) (ver sección 3.6.1) después de haber aplicado el método de puntos de correspondencia. En este caso, el método de calibración con 13 parámetros obtuvo mediciones de 1.313 y 1.735 píxeles para los lentes de .4 y .21 respectivamente. Al usar 14 parámetros estas distancias se redujeron a 1.028 y 1.2944 píxeles utilizando los mismos lentes.

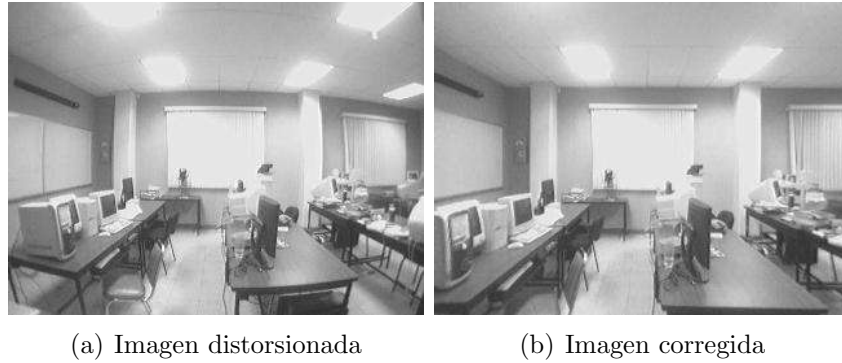


Figura 3.7: Corrección de imágenes utilizando los parámetros θ .

Cabe mencionar que durante estas pruebas, el método de 13 parámetros utiliza valores iniciales significativamente más cerca de la solución. El método de 14 parámetros sigue utilizando los mismos valores iniciales ya descritos con anterioridad.

3.8. Conclusiones

Hemos corregido la distorsión radial introducida por la lente de una cámara utilizando la técnica de registro de imágenes. Al igual que el método de Autocalibración las imágenes son corregidas con éxito. Sin embargo podemos identificar algunas diferencias que hacen más conveniente, para esta investigación, utilizar la técnica de registro de imágenes:

1. En la Autocalibración es necesario binarizar las imágenes.
2. En la Autocalibración se hace necesario redimensionar los cúmulos dependiendo del patrón de calibración utilizado o bien implementar aplicaciones robustas que identifiquen las líneas correctas.

3. En la Autocalibración es importante conocer el número de líneas que se quiere encontrar.

Por otra parte, en la técnica de registro de imágenes observamos que utilizar 14 parámetros permite reducir la dependencia hacia los valores iniciales de θ además de reducir el error.

Capítulo 4

Sistema de visión estéreo

En este capítulo estableceremos las bases del funcionamiento de la visión estéreo, asumiendo que las cámaras utilizadas han sido corregidas de la distorsión. Se describirán los problemas relacionados y se establecerán las estrategias para solucionarlos. Este capítulo sienta las bases para el capítulo de reconstrucción que se cubrirá más adelante.

4.1. Introducción

“La visión estéreo se refiere a la habilidad para inferir información sobre la estructura 3-D y la distancia de una escena a partir de dos o más imágenes tomadas desde diferentes puntos de vista” [23]. Y esta es precisamente la razón por la que los seres vivos tenemos dos o más ojos, ya que esto permite a nuestro cerebro procesar dos o más imágenes de la misma escena y generar así un sentido de la profundidad. Incluso algunos animales que tienen sus ojos totalmente opuestos (las gallinas por ejemplo), hacen continuos movimientos de su cabeza para obtener dicho recurso [8].

Una forma de entender el concepto y los problemas de la visión estéreo es a través del ejercicio del dedo pulgar: simplemente colocamos nuestro dedo pulgar a la altura de nuestros ojos y a una cierta distancia, entonces abrimos y cerramos alternadamente cada ojo. Podemos observar como nuestro dedo parece moverse horizontalmente hacia un lado y hacia otro, dependiendo de con cual ojo lo estemos observando. Incluso, si somos un poco más observadores, nos daremos cuenta de que los objetos detrás de nuestro dedo también se desplazan, algunos más que otros (pero seguramente menos que nuestro dedo). Este desplazamiento horizontal de los objetos es precisamente lo que utiliza nuestro cerebro para calcular la profundidad a la que se encuentran las cosas. Y con este simple ejercicio, es posible detectar que cuanto más cerca se encuentren los objetos más parecerá que se desplazan y que algunos de ellos están tan lejos que apenas se pueden percibir diferencias.

Por esta misma razón, el uso de dos cámaras o mejor dicho, el poder contar con dos o más imágenes de una misma escena, nos permitirá obtener la información de distancia con la cual podamos resolver una gran cantidad de problemas, por ejemplo, los relacionados

con la navegación en robótica móvil, reconocimiento aéreo, fotogrametría, cartografía, etc.[8].

En este capítulo se describe la problemática relacionada con los sistemas de visión estéreo y se describen algunas estrategias para resolverla. Sin embargo, para ello será necesario adentrarnos en el terreno de la geometría estéreo, en el análisis de algoritmos de correspondencia y en como minimizar los errores a la hora de hacer la reconstrucción 3-D.

4.1.1. Los dos problemas de la visión estéreo

Desde el punto de vista computacional, un sistema estéreo se enfrenta a dos problemas a resolver, uno de ellos es la correspondencia entre las imágenes y el otro es la posibilidad de reconstruir las estructuras 3-D a partir de imágenes estéreo.

La *correspondencia* entre imágenes se refiere a localizar, en ambas, la posición exacta de un mismo objeto. Una vez solucionado el problema de correspondencia, y utilizando las diferencias de las posiciones de los objetos (*disparidad*), en ambas imágenes, estamos en posibilidad de calcular la ubicación de ellos en el mundo real 3-D(*reconstrucción*). Esas diferencias entre imágenes de una misma escena, son conocidas como *mapa de disparidad*, en el cual se plasman todas aquellas correspondencias que sea posible detectar a través de algún procesamiento adecuado.

En [23] se definen estos problemas de la siguiente forma:

El problema de correspondencia: ¿Qué parte de las imágenes son proyecciones del mismo elemento de la escena?

El problema de la Reconstrucción: Dado un número de correspondencias entre las imágenes y algunos datos sobre la geometría del sistema estéreo ¿Qué tanto podemos averiguar sobre la estructura y ubicación 3-D de los objetos observados?

La *geometría estéreo* se refiere a conocer aquellas características geométricas relacionados con las cámaras, como son su distancia focal, la posición de una cámara con respecto a la otra y la orientación de las mismas.

Este capítulo está dedicado principalmente al problema de la correspondencia entre imágenes, describiendo sus características y una solución del mismo. Se hace una introducción al tema de la reconstrucción pero este problema será abordado en el siguiente capítulo.

4.2. Problema de Correspondencia

El problema de correspondencia, parte de que tenemos dos o más imágenes de una misma escena, en la cual tenemos que encontrar aquellos pixeles que correspondan al mismo elemento del mundo 3-D. En la figura 4.1 se muestran dos imágenes de una misma escena, tomadas desde diferentes puntos de vista. En la imagen izquierda se destacan con círculos algunos elementos que a simple vista tienen correspondencia con la imagen

derecha. Se observa que son precisamente las esquinas las características que más evidencian la correspondencia entre las imágenes. Con cuadros observamos elementos de la imagen izquierda que resulta difícil ubicar, aún a simple vista, en la imagen derecha, en donde indudablemente también aparecen. Aquí el problema es que el número de lugares donde pueden empatar es muy grande. Por último, encerrados en triángulos, destacamos elementos de la imagen izquierda que simplemente no tienen correspondencia en la imagen derecha. Esto último puede deberse a que la cámara izquierda literalmente no ve lo mismo que la cámara derecha. Aún más, la carencia de correspondencia puede ser originada por errores en la captura de la imagen, por la proyección de sombras o por diferencias en la iluminación.

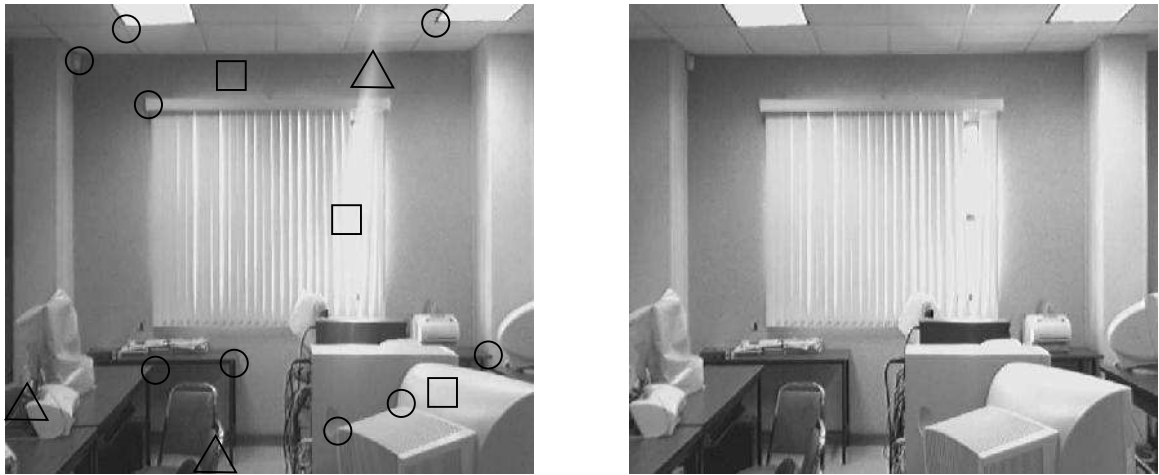


Figura 4.1: Características de correspondencia entre dos imágenes

En la figura 4.2(a) se ilustran algunos de los problemas más difíciles de resolver en la correspondencia de imágenes. La parte mas oscura es el área donde es probable encontrar correspondencias. Todos aquellos elementos que se encuentren en la parte más clara no proporcionarán ningún tipo de información útil al sistema estéreo. Con esto se ilustra la importancia que tienen tanto la apertura de las lentes como la distancia a la que se encuentran las cámaras entre sí (geometría estéreo).

La oclusión es otro elemento que complica la búsqueda de correspondencias entre las imágenes (ver figura 4.2(b)), lo que una cámara puede ver, es probable que la otra cámara no lo perciba.

Ahora comenzaremos a estructurar nuestra estrategia para solucionar los problemas de correspondencia. Para ello será necesario establecer un sistema estéreo cuya geometría nos facilite la resolución de los problemas ya citados.

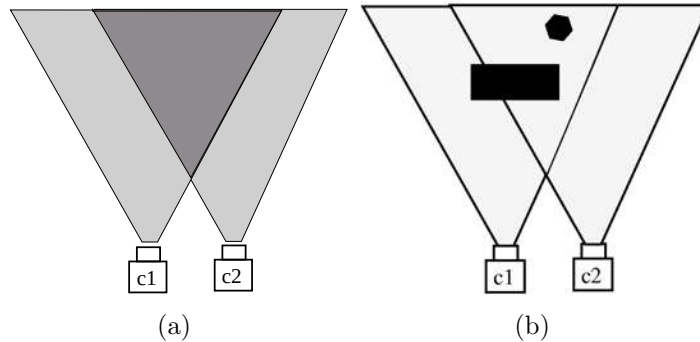


Figura 4.2: La geometría del sistema estéreo influye directamente en el problema de correspondencia. Colocar las cámaras demasiado lejos una de la otra (a), puede provocar grandes zonas de no correspondencia. Al tener puntos de vista diferente, las cámaras no necesariamente ven los mismos elementos (b), en este caso la cámara de la derecha *c2* puede ver el elemento hexagonal detrás del rectángulo, mientras que *c1* no lo alcanza a percibir. Este problema se conoce como oclusión y es uno de los grandes retos a los que se enfrenta la visión estéreo.

4.2.1. Consideraciones sobre el problema de correspondencia

Ahora que ya conocemos algunos de los problemas más fuertes a los que se enfrenta la visión estéreo, en lo que se refiere a correspondencia de imágenes, es necesario establecer que en primer lugar, no todos los elementos de una imagen tendrán correspondencia en la otra, es decir, no necesariamente empatarán con algún elemento en la otra imagen, e incluso, aún habiendo un empate, no necesariamente se trata de una correspondencia correcta. De hecho son pocos los elementos de las imágenes que son buenos candidatos para hacer una correcta correspondencia. Cuando un elemento de una imagen empata con algún elemento de otra imagen, se dice que éstos han hecho *match*¹. Puede haber varios matches para un mismo elemento pero solamente habrá uno correcto: su correspondencia. Nuestra primer tarea a resolver será buscar el mayor número de correspondencias entre las imágenes.

Por lo tanto, es necesario establecer un criterio apropiado para determinar cuando un match ha ocurrido. Llamaremos a este criterio como de similaridad.

Partiendo de la idea de que las imágenes digitales son un arreglo de elementos discretos (píxeles), que para nuestro caso pueden tomar valores de 0(negro) a 255(blanco), el problema al que nos enfrentamos es una búsqueda de patrones de números que no necesariamente serán idénticos, o que incluso pueden no existir debido a los problemas de oclusión.

Por otra parte, las imágenes que utilizamos para nuestro trabajo contienen 640 píxeles

¹Palabra que en inglés se puede interpretar como igualar, emparejar o corresponder.

de ancho por 480 pixeles de altura, esto quiere decir que de entrada tenemos más de 300 mil posibles candidatos que pueden tener correspondencia con los elementos de la otra imagen (exactamente son 307,200 elementos). Para cada pixel de una de las imágenes, es necesario buscar en la otra imagen algún elemento que, en base a un criterio previamente definido, logre hacer match.

Suponiendo que nuestro sistema estéreo está integrado por dos cámaras, alineadas horizontalmente una al lado de la otra, y haciendo una analogía con nuestros ojos las llamaremos cámara derecha y cámara izquierda. En el problema de correspondencia, podemos establecer que un pixel de la imagen tomada por la cámara derecha $p_d(x_d, y_d)$, puede tener o no correspondencia. En caso afirmativo, ésta estará en la posición $p_i(x_i, y_i)$ de la imagen tomada por la cámara izquierda, en donde $y_i \geq y_d$ (ver figura 4.3). Si las cámaras se encuentra a alturas similares y con ejes de proyección paralelos, seguramente x_d tendrá un valor cercano a x_i . A pesar de estas consideraciones, una búsqueda razonable implicaría cubrir al menos una área de unos 80 por 10 pixeles, es decir 800 elementos. En la figura 4.3 se ilustra este ejemplo, donde la característica buscada (esquina inferior izquierda del botiquín), puede estar un poco hacia arriba o hacia abajo pero siempre a la derecha de la posición inicial y_d .

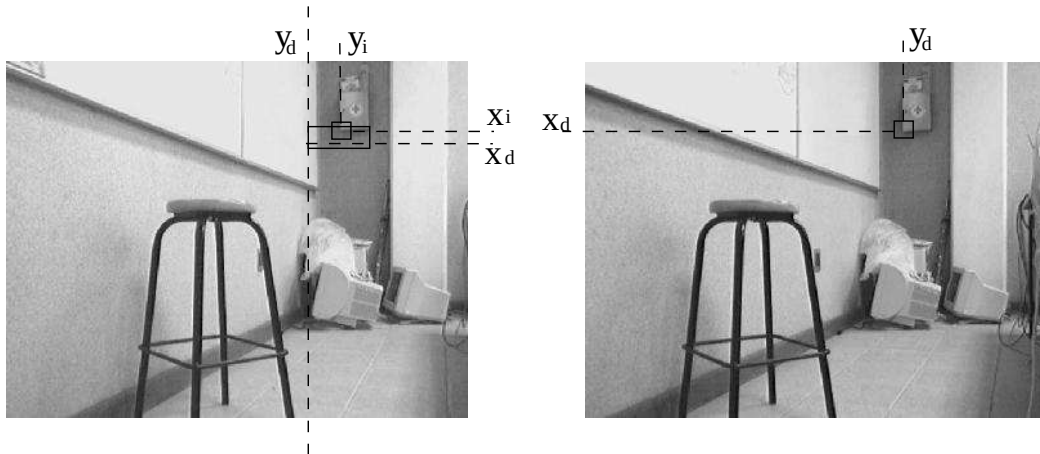


Figura 4.3: Una búsqueda horizontal de 80 pixeles apenas parece razonable para determinar un match, sin embargo el banco al frente de las imágenes nos advierte que los objetos cercanos pueden requerir de una área de búsqueda más extensa.

Al no estar perfectamente alineadas las cámaras es necesario considerar, en la búsqueda de correspondencias, varios pixeles hacia arriba y hacia abajo de x_d , en lugar de hacer la búsqueda en una sola dirección (sobre el mismo reglón x_d).

Nuestro problema es ahora lograr que las imágenes estéreo queden perfectamente alineadas, para reducir el tiempo de búsqueda de correspondencias. Dicho de otra forma, necesitamos alinear las imágenes para que $x_d = x_i$. En la siguiente sección hablaremos

sobre este problema y cómo solucionarlo.

4.2.2. Rectificación del sistema estéreo

En esta sección buscaremos establecer una geometría entre las imágenes que nos permita realizar búsquedas en una sola dirección. Esta característica se conoce como *geometría epipolar* y el proceso para encontrarla lo llamaremos rectificación del sistema estéreo.

La rectificación de un sistema de visión estéreo, nos facilitará llegar a la solución de los problemas de correspondencia y de reconstrucción. En la figura 4.4 se ilustra lo que en visión se conoce como *geometría epipolar*. En esta figura, un punto P es proyectado sobre las imágenes Π y Π' , en los puntos p y p' respectivamente. Ya que se trata de una proyección, podemos ver que todos los puntos sobre la línea $F - P$ quedan representados por p . La proyección de $F - P$ sobre el plano Π' se denomina línea epipolar y representa todas las posibles correspondencias de p en Π' . La intersección de las líneas epipolares con la línea base $F - F'$ se denominan epipolos de una imagen con relación a la otra. Por último, el plano formado por $PF F'$ se denomina plano epipolar, el cual corta a las dos imágenes Π y Π' en las líneas epipolares [24] y [8].

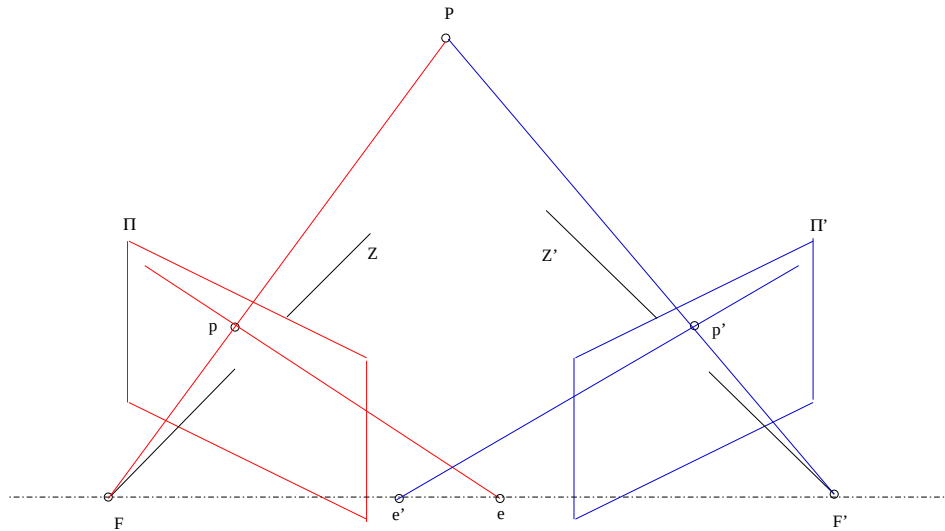


Figura 4.4: Línea y plano epipolar

Podemos concluir, que con excepción del epipolo, solamente una línea epipolar va a través de cualquier punto de una imagen y todas las líneas epipolares de una imagen pasan a través del epipolo. De esta forma, *si queremos buscar correspondencias para un punto p de una imagen deberíamos hacerlo sobre la línea epipolar de la otra imagen.*

En un sistema estéreo estándar, donde las cámaras tienen ejes ópticos paralelos y sus planos de imagen son ortogonales a los ejes ópticos, la búsqueda de correspondencias se reduce a buscar sobre un mismo renglón de píxeles. El proceso de conversión de imágenes tomadas con una geometría dada a las correspondientes imágenes obtenidas por un sistema estéreo estándar se llama *Rectificación*.

Proceso de rectificación de un sistema estéreo

La solución buscada estriba entonces en encontrar la geometría epipolar para desarrollar búsquedas de correspondencia más eficientes. Una solución está dada por una transformación proyectiva aplicada a cada imagen de tal forma que las líneas epipolares queden perfectamente colineales.

En la sección 3.1 vimos como el proceso de calibración, a través del registro de imágenes, nos permitió corregir los efectos de la distorsión radial de las lentes de cada una de las cámaras. Sin embargo, para obtener una calibración estéreo, en donde no solamente se corrija la distorsión, sino que también se construya una geometría epipolar, es necesario hacer las siguientes consideraciones:

1. En primer lugar tenemos que establecer la *geometría estéreo*, ya que en base a ella se realizarán algunos procesos de calibración y reconstrucción. Conocer la geometría estéreo implica saber la relación geométrica que guardan las cámaras entre sí (ver figura 3.6 de la página 56). En nuestro caso, se consideró a la cámara izquierda como el eje coordenado origen $c_i(x_i = 0, y_i = 0, z_i = 0)$ y se buscó en forma cuidadosa que las coordenadas de la cámara derecha $c_d(x_d, y_d, z_d)$ mantuvieran la relación $(x_d = x_i)$ y $(z_d = z_i)$. De la misma forma, que la cámara superior $c_s(x_s, y_s, z_s)$ permaneciera, en lo posible, con $(y_s = y_i)$ y $(z_s = z_i)$ (ver figura 4.5).

En otras palabras, se busca que las cámaras izquierda y derecha queden, hasta donde nos sea posible, a la misma altura y con sus ejes de proyección paralelos. La cámara superior deberá quedar encima de la cámara izquierda y también conservar su eje de proyección paralelo al de las otras dos cámaras. Para los valores dX y dY , también será necesario llevar a cabo mediciones en forma manual. Para este trabajo, se utilizaron distancias aproximadas a $10cm$, tanto para dX como para dY .

2. En base a la idea de que las tres cámaras son de características similares y a que la distorsión en el centro de las imágenes tiende a ser nula, se procedió entonces a crear un patrón de alineación que nos permitiera configurar las tres cámaras, para que sus ejes ópticos (los ejes Z) se mantuvieran paralelos.

El patrón de alineación está basado en la figura 4.5 y lo mostramos en la figura 4.6. La idea es simple, colocamos el patrón de alineación al frente de las tres cámaras, sobre un plano paralelo al plano de las cámaras y a cierta distancia (unos 3 mts.

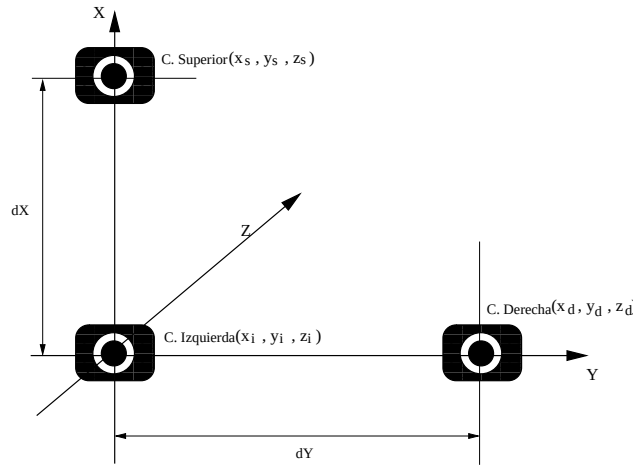


Figura 4.5: La geometría estereo. La relación que guardan las cámaras en el sistema estereo debe ser conocida con tanta exactitud como no los permitan los instrumentos de medición. Entre mejor sean las mediciones en esta etapa, tendremos menos ajustes en el proceso de reconstrucción.

es suficiente). Ahora a prueba y error tendremos que ir direccionando el eje óptico de cada una de las cámaras con su respectiva marca. Dicho de otra forma, cada una de las marcas tendrá que ser encontrada por las coordenadas del centro de la imagen de la cámara a que corresponda. La figura 4.7 ilustra este proceso.

3. Ahora se llevará a cabo la adquisición de las imágenes del patrón de calibración, pero teniendo el cuidado de utilizar el mismo patrón para las tres cámaras. El patrón de calibración es el que utilizamos anteriormente y está ilustrado en la figura 3.3(a) de la página 54. El patrón debe ser centrado con el eje óptico de una de las cámaras (nosotros utilizamos la cámara izquierda) y entonces se adquiere la imagen origen para esa cámara. Posteriormente, y tomando como referencia la posición inicial del patrón de calibración, desplazar éste en dY para la cámara derecha y dX para la cámara superior, llevando a cabo en ambos pasos, la adquisición de las imágenes origen para cada una de ellas. El desplazamiento del patrón de calibración al frente de las cámaras debe hacerse tal y como se estableció en la figura 4.5.
4. Por último, aplicamos el método de registro visto en la sección 3.1, para cada una de las imágenes origen y la imagen sintética del patrón de calibración.

Como resultado, obtendremos los parámetros necesarios para corregir la distorsión introducida por las lentes, pero también obtendremos los parámetros que nos aproximan en gran medida a un sistema estereo estándar, ya que las tres cámaras son empatadas con un mismo patrón de calibración. En otras palabras, lo que las cámaras están viendo,

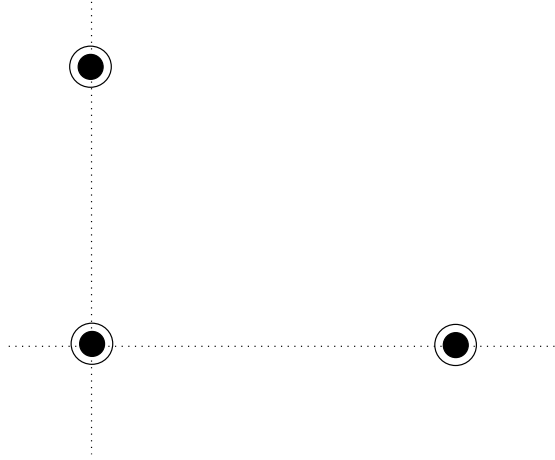


Figura 4.6: El patrón de alineación consiste únicamente en tres marcas que guarden una misma relación geométrica que las cámaras del sistema estéreo.



Figura 4.7: Los ejes de proyección de las cámaras (z_s, z_d, z_i) deberán quedar alineados con el centro de las imágenes. Aunque sabemos que el centro de una imagen no necesariamente coincide con su centro de distorsión, queremos aprovechar que ambos valores son muy cercanos entre sí.

independientemente de los posibles errores geométricos que se pudieron haber cometido previamente, es ajustado para que, por ejemplo, los elementos en las imágenes adquiridas por la cámara izquierda estén a la misma altura en las imágenes captadas por la cámara derecha. Para el caso de la cámara superior sucederá algo parecido, en relación a la cámara izquierda, sus elementos quedarán alineados sobre una misma columna en ambas imágenes.

Afortunadamente la metodología aplicada en la sección 3.1 nos permite llegar a una solución aceptable, ya que las correspondencias entre las imágenes quedan prácticamente colineales. En la figura 4.8 mostramos un par de imágenes a las cuales se les aplicó la corrección por registro y donde se resalta con líneas discontinuas la colinealidad de algunas correspondencias.

Una vez solucionado el problema de rectificación, estamos en condiciones de imple-

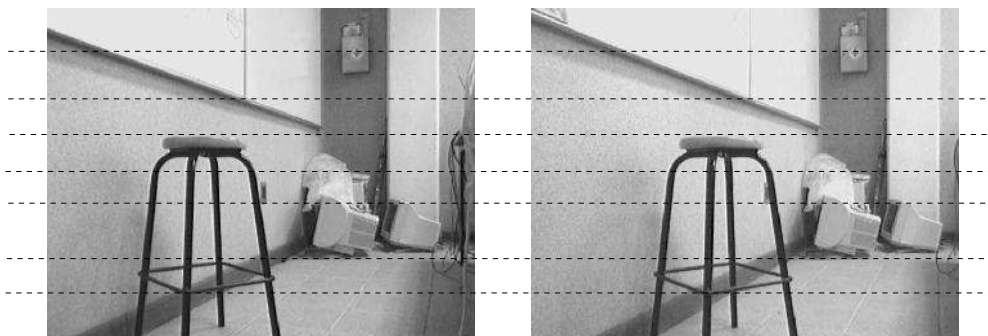


Figura 4.8: Ambas imágenes fueron adquiridas en un sistema estéreo calibrado obteniendo una colinealidad entre sus correspondencias bastante aceptable. Las imágenes de izquierda a derecha corresponden a la cámara derecha e izquierda respectivamente.

mentar algún algoritmo que nos permita inferir información de imágenes estéreo y con ello, llevar a cabo tareas de reconstrucción.

A continuación abordaremos el método de correlación, el cual es uno de los más utilizados debido en gran parte a su sencillez y facilidad para ser implementado, además de que nos puede obtener la información suficiente para llevar a cabo tareas de reconstrucción.

4.3. La correlación

En esta sección se explicarán los detalles que nos parecieron más importantes, sobre los métodos basados en correlación. Explicaremos como funcionan y los problemas a los que se enfrentan. En esta sección, implementamos dos estrategias de correlación de las cuales presentamos los resultados obtenidos.

Comenzaremos diciendo que la correlación es una metodología utilizada ampliamente para encontrar las correspondencias entre pares de imágenes estéreo.

Los sistemas estéreo determinan la profundidad a partir de dos o más imágenes de la misma escena, las cuales son tomadas al mismo tiempo desde dos diferentes puntos de vista. La tarea más importante y que más tiempo consume es la identificación de píxeles correspondientes [10].

En el libro de Trucco[23], clasifican los algoritmos de correspondencia en dos clases, los métodos basados en *correlación* y aquellos que se basan en *características*. La diferencia primordial es que los métodos de correlación se aplican sobre todos y cada uno de los píxeles de la imagen, mientras que los métodos basados en características intentan establecer una correspondencia entre conjuntos esparcidos de características de la imagen.

A través de un método de correlación podemos encontrar aquellos pixeles que representen al mismo punto del mundo real en ambas imágenes.

En la búsqueda para encontrar las correspondencias entre las imágenes, el método de correlación, buscan empatar un área de tamaño fijo de una imagen llamada ventana, con otra área del mismo tamaño en la otra imagen. La primera de estas áreas permanece fija y alrededor del pixel que se está evaluando. La segunda cambia de posición continuamente desplazándose sobre la segunda imagen en lo que se conoce como región de búsqueda. Para llevar a cabo esta tarea, se hace necesario establecer un criterio de similaridad entre las ventanas de las dos imágenes, lo cual determina una medida de correspondencia de las mismas.

Un algoritmo de correlación básico utiliza áreas de búsqueda fijas a través de las líneas epipolares y utiliza ventanas de tamaño fijo. A continuación describimos detalladamente este algoritmo:

Algoritmo de correlación basado en áreas de búsqueda y ventanas de tamaño fijo

1. En primer lugar definimos a la imagen derecha como I_d y a la imagen izquierda como I_i . Ambas imágenes guardan una geometría epipolar que permite que las correspondencias entre las mismas se mantengan colineales. Se define también una región de búsqueda y una ventana para medir la similaridad. Una región de búsqueda r_b razonable, puede ser de 80 pixeles ($r_b = 80$) y el tamaño de la ventana t_v puede establecerse de 5×5 pixeles ($t_v = 5$). Es necesario definir un criterio, para determinar cuando el algoritmo ha encontrado una correspondencia así como también un umbral, para determinar cuando la comparación entre dos ventanas ha producido un match. El criterio de correspondencia será, en este caso, quedarnos con el match que arroje el error más pequeño (el mejor match).
2. Para todos los pixeles de la imagen derecha buscaremos una posible correspondencia con su imagen izquierda. Sea $I_d(x, y)$ el pixel a evaluar, entonces habrá que resolver la siguiente expresión para $i = 1$ hasta r_b

$$m_c(x, y)_i = \sum_{m=-t_v/2}^{m=t_v/2} \sum_{n=-t_v/2}^{n=t_v/2} (I_d(x+m, y+n) - I_i(x+m, y+i+n))^2 \quad (4.1)$$

y almacenar el valor $m_c(x, y)_i$ que nos arroje el valor más pequeño.

3. Si el valor almacenado $m_c(x, y)_i$ está por debajo del umbral previamente definido, podemos establecer que $I_d(x, y)$ ha empatado con $I_i(x, y+i)$, por lo que almacenaremos el valor de i en $I_{disp}(x, y)$, donde a I_{disp} la denominaremos *imagen de disparidad* o *mapa de disparidad*. Si el valor de $m_c(x, y)_i$ está por encima del umbral, entonces $I_{disp}(x, y) = 0$ en i .

El resultado final será un mapa de disparidad I_{disp} con valores para cada pixel en un rango de 0 a r_b , donde valores de I_{disp} cercanos a r_b nos describan objetos muy cercanos

y los valores cercanos a cero describan objetos lejanos. En caso de ser cero, $I_{disp}(x, y)$ representa elementos a los cuales no se les pudo asignar una correspondencia.

En la figura 4.9 mostramos un mapa de disparidad obtenido con este algoritmo. Las zonas más claras indican elementos que se encuentran más cerca de las cámaras. Los elementos más lejanos aparecen con tonalidades oscuras. Las partes negras de la imagen, indican que el algoritmo no pudo asignar correspondencias para dichos elementos. El umbral seleccionado fue de 4 unidades en tono de gris, en una escala de 0 a 255. Esto quiere decir que para considerar a $m_c(x, y)$ como un match, éste deberá satisfacer la condición $m_c(x, y) < 4^2 t_v^2$.

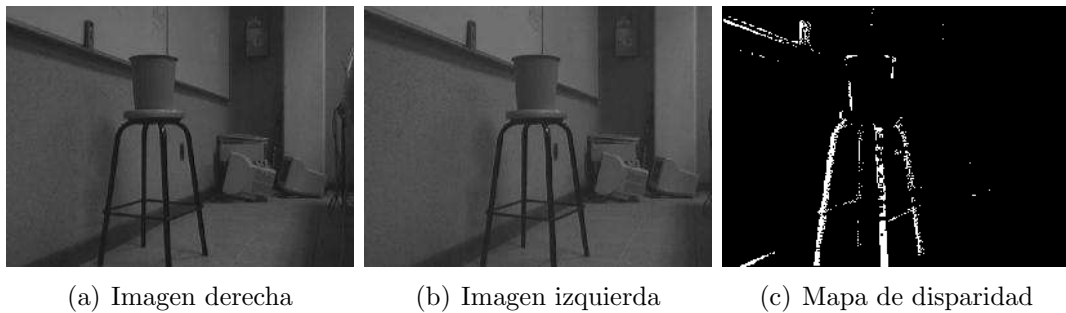


Figura 4.9: Ejemplo de un mapa de disparidad (c), a partir de dos imágenes estéreo.

A partir de este algoritmo de correlación, se pueden hacer una gran cantidad de modificaciones que permitan encontrar mapas de disparidad de mejor calidad, es decir, con menos errores (correspondencias falsas), con más información (mayor cantidad de correspondencias verdaderas) y en un menor tiempo. En [10] se describen algunos algoritmos que permiten encontrar correspondencias siguiendo diferentes estrategias y se describen sus limitaciones y bondades.

A continuación enumeramos algunos de los problemas que se tienen que resolver para generar un mapa de disparidad que sea confiable.

Falsas correspondencias

El origen de una falsa correspondencia suele ser muy variado y es casi imposible eliminar o contemplar todas aquellas circunstancias que las provocan. El caso más común es cuando el algoritmo se enfrenta a imágenes que presentan pocos cambios. Ejemplo de ello pueden ser las paredes, la superficie de un pizarrón, el piso, etc. En tales casos los algoritmos tendrán una gran cantidad de empates para cada elemento que se esté evaluando. El problema es determinar ¿cuál de esos matches es el correcto? En el ejemplo del algoritmo de correlación previamente descrito decidimos elegir el mejor match, lo cual no garantiza que se trate de una correspondencia.

Otro tipo de escenarios que arrojan falsas correspondencias está ejemplificado en la figura 4.10 el cual, a diferencia del caso anterior, presenta un escenario repleto de elementos muy parecidos, como son las patas de las mesas y sillas. El resultado serán una gran cantidad de matchs, en donde el mejor de ellos no necesariamente será la solución. Por lo tanto, el mapa de disparidad (ver figura (c)), presentará una gran cantidad de errores producto de las falsas correspondencias.

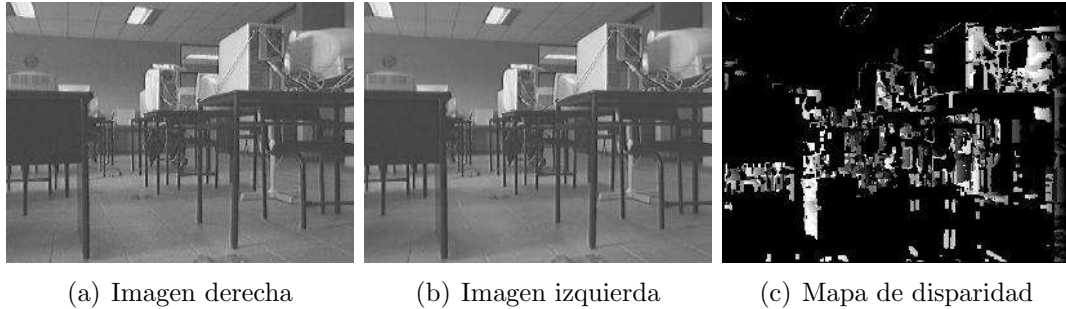


Figura 4.10: En este caso, el mapa de disparidad generado por el algoritmo de correlación es poco confiable. La gran cantidad de elementos similares hacen de éste uno de los peores escenarios para los algoritmos de correlación.

Oclusión

La oclusión se presenta cuando por alguna circunstancia, alguno de los elementos de una imagen no tiene correspondencia en la otra imagen. En la figura 4.1 de la página 61 ya se habían ejemplificado con triángulos algunos casos de oclusión. Objetos cercanos ocultan elementos para una cámara, pero la otra si los puede capturar (ver figura 4.2 de la página 62). Factores como la iluminación y la propia naturaleza de la geometría estéreo, pueden provocar que en las imágenes no aparezcan los mismos elementos.

Iluminación

Los efectos de la iluminación suelen representar problemas para los algoritmos de correlación, ya que por ejemplo, al capturarse las imágenes pueden transcurrir décimas de segundo entre una y otra, lo que produce diferencias en la iluminación y esto se traduce en diferentes tonalidades en las imágenes. El diferente punto de vista que guardan las cámaras también provoca diferentes percepciones de la iluminación. Aún más, las cámaras por si mismas mantienen una calibración que les permite ajustarse a la cantidad de luz disponible, lo que conduce a que las imágenes presenten tonalidades diferentes.

Tiempo de procesamiento

El número de operaciones que se necesitan para evaluar cada uno de los píxeles, depende directamente del tamaño de la ventana de correlación, así como de la región de búsqueda. Aún siendo un sistema estéreo estándar, los algoritmos de correlación requieren de una gran cantidad de operaciones. Tener regiones de búsqueda pequeñas, nos permite considerar correspondencia para elementos que se encuentren relativamente lejos de las cámaras. Sin embargo, los objetos cercanos serán descartados, ya que éstos necesitarían una mayor región de búsqueda. Por otra parte, ventanas de correlación pequeñas aceleran el proceso de búsqueda de correspondencias, pero a costa de una mayor cantidad de errores (falsos matches).

A continuación describiremos un algoritmo de correlación que, por sus características, fue elegido como parte de esta tesis para llevar a cabo parte de la construcción de los mapas de disparidad. Estamos hablando del algoritmo de Shirai[13].

4.3.1. Algoritmo de Shirai

Un algoritmo de correlación que nos dio buenos resultados en la construcción del mapa de disparidad, pero sobre todo en el tiempo de procesamiento, es el desarrollado por Yoshiaki Shirai. La idea central de este algoritmo, es calificar desde un principio que tan factible es que un pixel pueda tener correspondencia. De esta forma, el algoritmo de Shirai selecciona solamente los bordes de una imagen como firmes candidatos a una búsqueda de correspondencias. Por lo tanto, el algoritmo de Shirai descarta todos aquellos píxeles de la imagen que no sean detectados como bordes.

En la sección 3.7 se describió como a través de las derivadas de una imagen podemos detectar bordes, ya sean verticales u horizontales o ambos.

Una forma confiable para detectar bordes, independientemente de la dirección de las derivadas, es a través de la magnitud del gradiente de la imagen. La magnitud del gradiente de un punto (x, y) en la imagen I se define como:

$$|\nabla I(x, y)| = \sqrt{\left(\frac{\partial I(x, y)}{\partial x}\right)^2 + \left(\frac{\partial I(x, y)}{\partial y}\right)^2} \quad (4.2)$$

En donde se deberá definir un umbral para $\nabla I(x, y)$ que nos ayude a determinar cuando se ha detectado un borde.

Otra consideración que se hace en el algoritmo de Shirai, es manejar ventanas de correspondencia de diferentes tamaños, comenzando por una ventana de 3×3 y hasta llegar a un tamaño máximo previamente definido. De la misma forma, también se definen diferentes tamaños para la región de búsqueda, comenzando con la región de búsqueda más grande, la cual irá decreciendo conforme la ventana de correspondencia vaya aumentando. Otro aspecto importante que considera el algoritmo de Shirai, es el criterio para

determinar si un match es una correspondencia o se trata de un posible error. La idea es poder distinguir entre varios matches, aquel que sea notablemente menor que los demás.

A continuación presentamos el algoritmo de Shirai descrito en [13].

Algoritmo de Shirai

```

inicializar MAX_VENTANA inicializar d1, d2 y d3 para cada pixel
R(p) de la imagen derecha
  si R(p) es un borde entonces
    inicializar el parámetro de ventana K
    inicializar la región de búsqueda I
    mientras K < MAX_VENTANA
      inicializar el tamaño de la ventana 2(K)+1
      para cada pixel L(q) de la región de búsqueda I en la imagen izquierda
        calcular el error entre las ventanas con la función SIMILARIDAD(R(p),I(q))
      si existe un error único mínimo más pequeño que d1 entonces
        fijar el valor de disparidad para el punto R(p) en el mapa de disparidad
        abortar ciclo
      si todos los valores de error son mayores que d2 entonces
        no existe correspondencia
        abortar ciclo
      incrementar K en uno
      decrementar I utilizando d3, I=I-d3
    fin ciclo mientras
  fin ciclo para

```

Donde $d1$ es el umbral para determinar cuándo las ventanas alrededor de $R(p)$ y $L(q)$ son lo suficientemente similares para hacer match, lo cual no garantiza que se trate de una correspondencia, ya que puede haber varios matches para un mismo punto. El umbral $d2$ se utiliza para determinar una posible oclusión, o que el pixel que se está analizando corresponde a un elemento que se encontraba muy cerca de las cámaras, y por lo tanto no se deberá seguir tratando de encontrar un empate con ventanas más grandes. El parámetro $d3$ se utiliza para modificar la región de búsqueda I , la cual va decreciendo conforme la ventana va creciendo. De esta forma, para áreas de búsquedas muy grandes, de uno 120 pixeles por ejemplo, utilizaremos la menor de las ventanas, por ejemplo 3×3 , y cuando necesitemos ventanas muy grandes, el área de búsqueda deberá ser razonablemente muy pequeña, digamos de unos quince o veinte pixeles. En el peor de los casos, la ventana crecerá a un tamaño máximo preestablecido, lo cual indicará que es poco probable que exista un match que satisfaga la función de similaridad $SIMILARIDAD(R(p), I(q))$. La función de similaridad puede ser la misma que utilizamos anteriormente (ver la ecuación 4.1 de la página 69).

Por otra parte, el algoritmo de Shirai no confía en el criterio del mejor match para determinar una correspondencia, a excepción de que ese match sea único o que sea bastante mejor que otros candidatos.

Enseguida presentamos una nueva estrategia para determinar una correspondencia entre un grupo de candidatos (matches), la cual deberá incluirse en el mapa de disparidad. La estrategia consiste en agrupar los diferentes matches arrojados por el método de Shirai, obtener su media y su desviación estándar y determinar que tan alejado se encuentra nuestro match de la media. En nuestros experimentos, elegir matches suficientemente alejados de la media disminuyó una gran cantidad de errores a que si elegimos simplemente el mejor match.

Estrategia para elegir una correspondencia

1. Almacenamos todos los matches obtenidos para un mismo tamaño de la ventana y de la región de búsqueda.
2. Si existe un único match, querrá decir que se trata de una correspondencia, por lo que actualizamos nuestro mapa de disparidad y continuamos con la búsqueda de la correspondencia para el siguiente pixel.
3. Si existen varios matches, obtenemos la media μ de ellos,

$$\mu = \frac{1}{n} \sum_{i=1}^{i=n} m_i$$

donde n es la cantidad de matches obtenidos y m_i es el resultado obtenido de la función de similaridad para cada uno de los n matches obtenidos.

4. Obtenemos la desviación estándar σ de los datos obtenidos

$$\sigma = \frac{1}{n} \sqrt{\sum_{i=1}^{i=n} (m_i - \mu)^2}$$

5. Si el mejor match se encuentra lo suficientemente alejado de cualquiera de los demás, por ejemplo a más de 2σ , entonces lo consideramos para el mapa de disparidad, ya que es muy probable que se trate de una correspondencia. De lo contrario podemos seguir buscando con una ventana más grande y una área de búsqueda más pequeña.

Al utilizar la desviación estándar como una medida para elegir una correspondencia, se nos permite eliminar una gran cantidad de errores a que si lo hiciéramos simplemente eligiendo el mejor match. Por lo tanto utilizando este criterio generalmente no encontraremos más correspondencias, si en cambio las que encontremos generarán mapas de disparidad con menos errores.

Además, como una optimización al algoritmo de Shirai, si consideramos que el principal criterio para buscar correspondencias es a través de la detección de bordes, entonces,

al momento de buscar empates con la otra imagen, podemos indicarle al algoritmo que solamente compare con aquellos que también sean bordes, es decir, solo evaluaremos cuando tanto $(p, E_d(p))$ como $(q, E_i(q))$ sean bordes. Esto ahorrará comparaciones inútiles ya que es muy poco probable que un borde detectado en la imagen derecha haga match con algo que no sea borde en la imagen izquierda. Por lo tanto mantendremos la calidad del mapa de disparidad y lo calcularemos más rápido.

Por otra parte, las operaciones que involucra calcular la desviación estándar afectan muy poco al algoritmo, ya que dichas operaciones solo se llevan a cabo cuando se cumplen las condiciones de bordes y de similaridad.

Un ejemplo del algoritmo de Shirai se muestra en la figura 4.11, donde $\sigma = 2$ y utilizamos regiones de búsqueda de 80 pixeles y ventanas máximas de 7×7 .

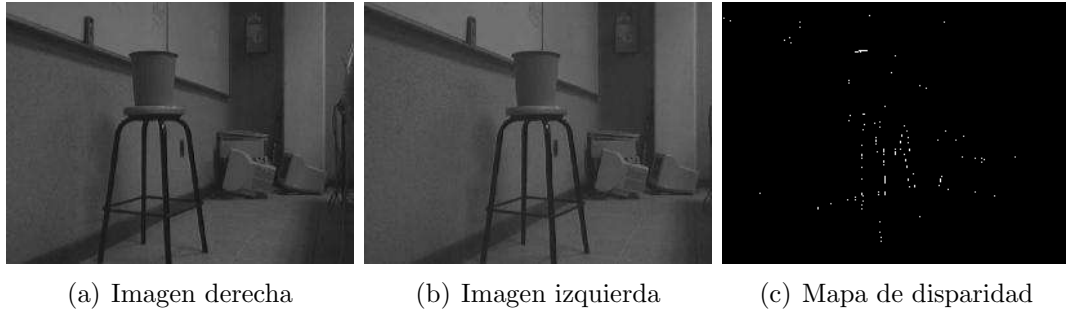


Figura 4.11: El mapa de disparidad generado por el algoritmo de correlación de Shirai se comporta adecuadamente, ya que solamente detecta la disparidad en los bordes de la imagen, evitando así los errores debido a superficies homogéneas.

Los principales problemas que resuelve el algoritmo de Shirai comparado con el método de correlación de ventanas y áreas de búsqueda fijas, son la reducción de errores en el mapa de disparidad, y el tiempo que tarda en procesar las imágenes para calcularlo. Haciendo una comparación entre ambos algoritmos de correlación, y considerando condiciones de operación similares, es decir, con regiones de búsqueda de hasta 80 pixeles, con una ventana máxima de 7×7 y las mismas imágenes estéreo de 640×480 pixeles, obtuvimos que el tiempo de cálculo para el algoritmo de Shirai fue de menos de 2 segundos (ver figura 4.11(c)), mientras que el algoritmo de correlación que utiliza ventanas y regiones de búsqueda fijas tardó más de 40 segundos (ver figura 4.9(c)). Además del tiempo, el mapa de disparidad del algoritmo de Shirai presenta menos errores que el mapa de disparidad generado por el otro método.

Sin embargo, el algoritmo de Shirai solamente aportará información al mapa de disparidad cuando se trate exclusivamente de bordes. Esto producirá grandes áreas del mapa de disparidad sin información, problema que se agravará más cuando se trate de escenas con amplias superficies homogéneas, como por ejemplo paredes o pisos.

En el siguiente capítulo se presenta una estrategia para solucionar el problema de las

superficies homogéneas a través de algoritmos de correlación y propagación. Se introduce el uso de un telémetro láser como una herramienta para generar mapas de disparidad, en conjunto con el algoritmo de Shirai. Antes de ello, en la siguiente sección, haremos una pequeña introducción a la solución del problema de la reconstrucción.

4.4. Problema de Reconstrucción

Una vez obtenido el mapa de disparidad y conociendo la geometría estéreo del sistema, podemos enfocar nuestra atención en el problema de reconstrucción. Si la geometría del sistema estéreo es conocida, entonces el mapa de disparidad puede ser convertido a un mapa 3-D de la escena que se está observando [23].

También de [23] tomamos la figura 4.12 con la cual ilustramos la idea básica de la reconstrucción estéreo. En esta figura se muestra una vista desde arriba de un sistema estéreo compuesto por dos cámaras. El plano de la imagen izquierda l_i es coplanar al plano de la imagen derecha l_d . O_i y O_d son los centros de proyección y los ejes ópticos son paralelos. Ahora, para determinar la posición de P en el mundo 3-D utilizaremos algo de trigonometría. Consideremos a p_i y p_d las proyecciones de P en las imágenes izquierda y derecha respectivamente. Si (p_d, p_i) se considera una correspondencia entonces podemos decir que P es la intersección de los rayos $O_i p_i - O_d p_d$. Si T es la distancia entre los centros de proyección O_i y O_d (línea base), entonces. Sea x_i y x_d las coordenadas de p_i y p_d con respecto a los puntos principales c_i y c_d , f la distancia focal de ambas cámaras y Z la distancia entre P y la línea base, podemos formar los triángulos similares (p_i, P, p_d) y (O_i, P, O_d) . Con lo cual podemos establecer que²:

$$\frac{T + x_i - x_d}{Z - f} = \frac{T}{Z} \quad (4.3)$$

Si establecemos que la disparidad es $d = x_d - x_i$, entonces podemos expresar 4.3 como:

$$Z = \frac{fT}{d} \quad (4.4)$$

En la ecuación 4.4 se observa que la profundidad Z es inversamente proporcional a la disparidad d . También podemos apreciar que para la resolución del problema de reconstrucción están en juego varios elementos: f, T, c_i, c_d y que resolviendo sus valores podemos resolver el problema de reconstrucción.

La distancia focal f está determinada por el tipo de lente que utilicemos en la cámara y es un dato que generalmente viene especificado por el fabricante. Sin embargo, de no tenerlo, lo podemos calcular a partir de la propia ecuación 4.4. A continuación proporcionamos una estrategia útil para calcular la distancia focal en base a un sistema estéreo.

²Debemos considerar que las coordenadas de x_i y x_d pueden ser positivas o negativas, dependiendo si se encuentran a la derecha o a la izquierda del centro de la imagen c_i y c_d respectivamente. En este caso en particular, x_i es negativa y x_d es positiva.

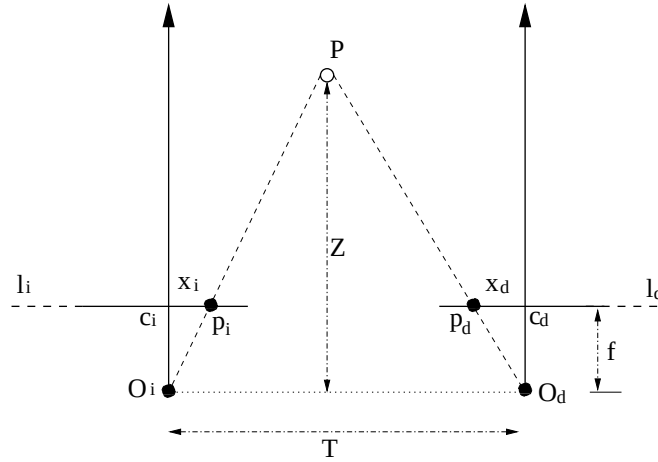


Figura 4.12: Conociendo la geometría estereográfica (T) y los parámetros intrínsecos (f, c_l, c_r) podemos calcular la profundidad Z de un punto P utilizando la disparidad que se genera con ambas imágenes.

Estimación de la distancia focal f

Para estimar la distancia focal a partir de la ecuación 4.4 nos apoyaremos en la figura 4.13. En esta imagen se aprecian dos cámaras cuya distancia T es conocida. La distancia Z no es conocida ya que ignoramos la posición exacta del punto focal de cada cámara. Sin embargo, sin importar la distancia Z , podemos utilizar una marca en frente de las cámaras, la cual la desplazamos en intervalos conocidos Δ_z y en cada uno de ellos obtenemos un par de imágenes estereográficas. Entonces, utilizando la ecuación 4.4 podemos obtener:

$$Z - \Delta_z = \frac{fT}{d_0} \quad (4.5)$$

Donde d_0 es la nueva disparidad generada con la marca a la nueva distancia $Z - \Delta_z$.

Repetiendo el procedimiento, desplazamos la marca nuevamente una distancia Δ_z . De esta forma podemos decir que:

$$Z - 2\Delta_z = \frac{fT}{d_1} \quad (4.6)$$

Ahora de 4.5 y 4.6 obtenemos la igualdad:

$$\frac{fT}{d_0} + \Delta_z = \frac{fT}{d_1} + 2\Delta_z \quad (4.7)$$

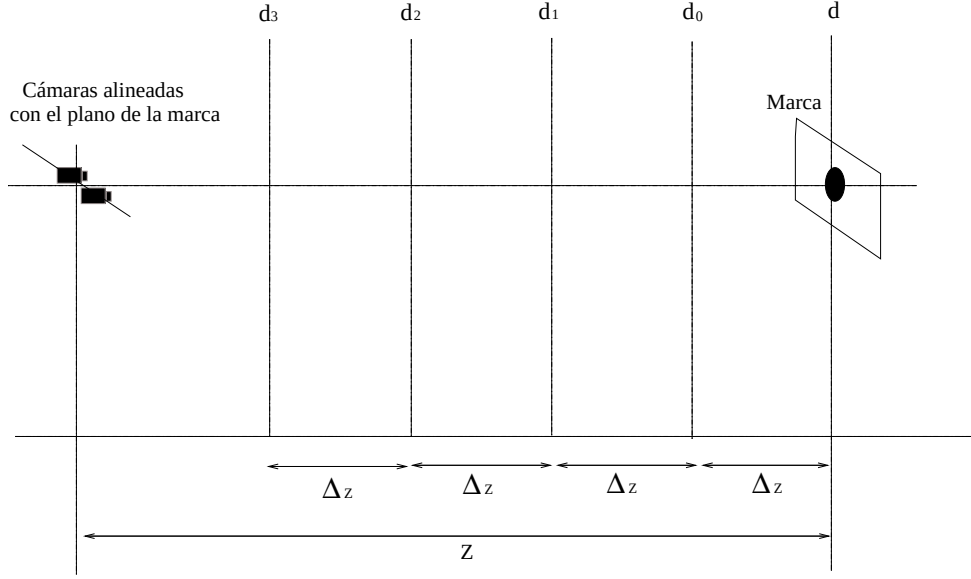


Figura 4.13: Para estimar el valor de la distancia focal de las lentes, lo podemos hacer a partir de varias imágenes estereo de un mismo objeto, el cual varía su distancia entre imagen e imagen con incrementos conocidos.

Resolviendo para f podemos obtener:

$$\frac{fT}{d_0} - \frac{fT}{d_1} = 2\Delta_z - \Delta_z$$

$$fT \left(\frac{1}{d_0} - \frac{1}{d_1} \right) = \Delta_z$$

$$f = \frac{\Delta_z}{T \left(\frac{1}{d_0} - \frac{1}{d_1} \right)} \quad (4.8)$$

Podemos calcular varias disparidades $d_0, d_1, d_2, \dots, d_n$, con sus respectivas variaciones Δ_z . Con dichas disparidades y con la ecuación 4.8 podemos estimar hasta $n(n-1)/2$ distancias focales ($f_1, f_2, \dots, f_{n(n-1)/2}$). Esto lo podemos hacer cruzando las mediciones (d_0 con d_1 , d_0 con d_2 , d_0 con d_3 y así sucesivamente), por lo que tendremos una cantidad suficiente de datos, a partir de unos pocos desplazamientos δ_z , que nos permitirán hacer una buena estimación.

En nuestro caso, aunque ya conocíamos que la distancia focal de nuestras lentes era de 0.4cm, decidimos corroborarlo a través del ejercicio ya descrito. En la imagen 4.14 se muestran algunas de las tomas realizadas a la marca de calibración. En total se llevaron

a cabo 10 diferentes mediciones y en cada una se tomó un par de imágenes, por lo que pudimos obtener 10 valores de disparidad ($d_0, \dots, d_9$). Utilizando la ecuación 4.8 y las correctas combinaciones entre las disparidades, llevamos a cabo 45 estimaciones para el valor de f . Las mediciones y las estimaciones se muestran al final de esta sección.

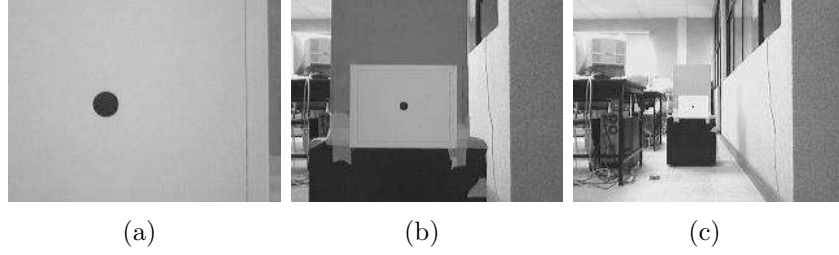


Figura 4.14: Algunas de las tomas realizadas durante la estimación de la distancia focal. Las tres tomas corresponden a la cámara derecha. En total se llevaron a cabo 10 mediciones variando $\Delta_z = 50cm$ y $Z \simeq 6$ mts.

Para mejores resultados, se hizo necesario calcular el valor de d a nivel subpixel. Para ello fue necesario calcular los centros de masa de las marcas de cada una de las imágenes. El centro de masa de la marca en la imagen se obtiene sacando un promedio de los pixeles que conforman la marca y pesándolos por el valor de su tono de gris. En nuestro caso los valores más altos corresponderán a los tonos más oscuros y los pixeles con tonalidades claras tenderán a cero. La ecuación que describe como calcular el centro de masa c_m es la siguiente:

$$c_m = \frac{1}{\sum_{i=m_{xi}}^{m_{xf}} \sum_{j=m_{yi}}^{m_{yf}} (p(i, j))} \sum_{i=m_{xi}}^{m_{xf}} \sum_{j=m_{yi}}^{m_{yf}} i(p(i, j)) \quad (4.9)$$

donde (m_{xi}, m_{yi}) y (m_{xf}, m_{yf}) son las esquinas superior izquierda e inferior derecha de una ventana que encierra completamente a la marca (en este caso el círculo negro de la figura 4.14).

Los resultados obtenidos durante las mediciones de la disparidad, utilizando los centros de masa de las marcas, son mostrados en la siguiente tabla:

Mediciones para desplazamientos			
	Cámara Derecha	Cámara Izquierda	Disparidad
Δ_z	302.5355	317.3595	14.8240
$2\Delta_z$	302.2476	318.8175	16.5699
$3\Delta_z$	302.1471	320.9088	18.7617
$4\Delta_z$	301.5909	323.2756	21.6847
$5\Delta_z$	298.7872	324.5759	25.7887
$6\Delta_z$	297.6862	329.1455	31.4593
$7\Delta_z$	296.6116	337.0735	40.619
$8\Delta_z$	293.1384	349.7264	56.5880
$9\Delta_z$	281.7120	376.3703	94.6583
$10\Delta_z$	227.5904	500.9182	148.78

En seguida mostramos el cálculo de la distancia focal en base a las mediciones anteriores. La distancia focal F_x expresada en pixeles. La distancia F expresada en centímetros. La equivalencia de pixeles a centímetros es 1 pixel = 0.00056cm y es un dato proporcionado por el fabricante de la cámara.

$F_x=710.556562$, $F=0.397912$
 $F_x=713.442300$, $F=0.399528$
 $F_x=709.916490$, $F=0.397553$
 $F_x=704.357821$, $F=0.394440$
 $F_x=707.927324$, $F=0.396439$
 $F_x=708.950036$, $F=0.397012$
 $F_x=710.101913$, $F=0.397657$
 $F_x=710.165491$, $F=0.397693$
 $F_x=712.458803$, $F=0.398977$
 $F_x=716.351573$, $F=0.401157$
 $F_x=709.596887$, $F=0.397374$
 $F_x=702.315539$, $F=0.393297$
 $F_x=707.273052$, $F=0.396073$
 $F_x=708.629602$, $F=0.396833$
 $F_x=710.026195$, $F=0.397615$
 $F_x=710.109658$, $F=0.397661$
 $F_x=712.697299$, $F=0.399110$
 $F_x=702.968394$, $F=0.393662$
 $F_x=695.501785$, $F=0.389481$
 $F_x=704.297798$, $F=0.394407$
 $F_x=706.725052$, $F=0.395766$
 $F_x=708.774500$, $F=0.396914$

$F_x=709.079899$, $F=0.397085$
 $F_x=712.178301$, $F=0.398820$
 $F_x=688.192123$, $F=0.385388$
 $F_x=704.964388$, $F=0.394780$
 $F_x=707.986209$, $F=0.396472$
 $F_x=710.241044$, $F=0.397735$
 $F_x=710.314975$, $F=0.397776$
 $F_x=713.736799$, $F=0.399693$
 $F_x=722.574606$, $F=0.404642$
 $F_x=718.316477$, $F=0.402257$
 $F_x=717.908040$, $F=0.402029$
 $F_x=716.069728$, $F=0.400999$
 $F_x=719.075000$, $F=0.402682$
 $F_x=714.108241$, $F=0.399901$
 $F_x=715.597287$, $F=0.400734$
 $F_x=713.927384$, $F=0.399799$
 $F_x=718.205389$, $F=0.402195$
 $F_x=717.092556$, $F=0.401572$
 $F_x=713.836990$, $F=0.399749$
 $F_x=719.581572$, $F=0.402966$
 $F_x=710.610851$, $F=0.397942$
 $F_x=720.832571$, $F=0.403666$
 $F_x=731.352651$, $F=0.409557$

Al obtener el promedio de los cálculos realizados, estimamos el valor de $f = 0,39833$ lo cual representa una buena aproximación. El error de hecho es de 2.98 píxeles.

4.5. Conclusiones

Una vez corregida la distorsión en las cámaras de vídeo se hace posible utilizar diferentes herramientas que nos permitan inferir información de distancia utilizando dos o más imágenes de una misma escena. En este capítulo se nos explicó la importancia de conocer la geometría estéreo para resolver los problemas de la visión estéreo. Utilizando la geometría estéreo podemos estimar la distancia focal, la cual a su vez, es necesaria para resolver el problema de reconstrucción. Se observó también en este capítulo que considerar diferentes estrategias para determinar la correspondencia entre imágenes, nos permitirá generar mapas de correspondencia confiables, aunque con un costo importante en tiempo de procesamiento y que reducir este costo implicó reducir la confiabilidad y utilidad de los mapas de disparidad.

En el siguiente capítulo se describirá un proceso completo para llevar a cabo la reconstrucción de estructuras 3-D a partir de un mapa de disparidad. Introduciremos el uso

del telémetro láser para solucionar el problema de las superficies homogéneas, utilizando un algoritmo de propagación, el cual también será descrito. El resultado final serán los mapas del entorno 3-D correspondiente a la escena de las imágenes estéreo.

Capítulo 5

Reconstrucción estéreo

En los capítulos anteriores se describieron características importantes sobre los sistemas de visión estéreo. En base a ello, determinamos que es necesario conocer los parámetros intrínsecos y extrínsecos del sistema para poder generar estimaciones confiables sobre el mundo 3-D. También se ha descrito la importancia de la rectificación.

En el capítulo anterior se realizaron algunos ejemplos con el algoritmo de Shirai. Hemos destacado que dicho algoritmo, al igual que la mayoría de los algoritmos de correlación, presenta dificultades para resolver el problema de las superficies homogéneas.

En este capítulo proponemos una estrategia nueva capaz de aprovechar y propagar la información de distancia generada por un telémetro láser, para construir un mapa de disparidad que nos arroje más información del entorno a que si solamente utilizáramos la correlación. En [2] ya han integrado un telémetro láser y un sistema de visión, aunque solamente utilizan una cámara y básicamente su objetivo es resolver el problema de localización de un robot móvil.

Con la propagación podemos encontrar información en áreas donde los algoritmos de correlación sufren, y que de hecho, el algoritmo de Shirai descarta completamente. Estamos hablando de las superficies homogéneas.

Posteriormente, a partir del mapa de disparidad se generan mapas del entorno 3-D, los cuales serán utilizados por un robot móvil para llevar a cabo tareas de navegación.

Comenzaremos hablando de la propagación láser y de los elementos que en ella participan.

5.1. Propagación láser

En esta sección se llevará a cabo la integración de la información generada por un telémetro láser a la geometría del sistema estéreo. En otras palabras, a partir de ciertos datos sobre la ubicación de algunos elementos del entorno 3-D dadas por el telémetro láser, los integraremos a las imágenes y los propagaremos en el mapa de disparidad.

Por lo tanto, los problemas a resolver en esta sección son:

1. Incluir al telémetro láser en la geometría del sistema de visión estéreo.
2. Convertir las lecturas de distancia generadas por el láser a coordenadas de la imagen en las cámaras.
3. Propagar la información, obtenida a partir de las lecturas del telémetro láser, en un mapa de disparidad.

Comenzaremos describiendo algunas generalidades sobre el telémetro láser.

5.1.1. El telémetro láser

Un telémetro láser es un instrumento de gran precisión para medir distancias. Básicamente funciona emitiendo una señal láser, la cual al ser reflejada, es capturada por un sensor previamente calibrado con la señal que se está generando. En base a un análisis de la señal emitida y la señal reflejada, es posible determinar con gran exactitud la distancia a la que se encuentra el emisor de el reflector en base al tiempo transcurrido desde que la luz salió y hasta que regresó.

Para nuestro trabajo utilizamos un telémetro láser de la marca SICK, el cual es capaz de generar hasta 361 lecturas, en un rango de 180° , sobre un plano horizontal (ver figura 5.1). De esta forma, podemos tener información de distancia cada medio grado.

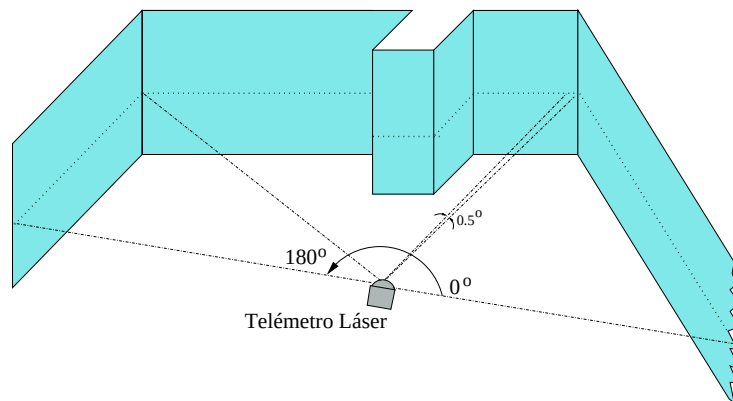


Figura 5.1: El telémetro láser utilizado en este trabajo puede cubrir 180° a través de 361 lecturas, es decir genera una lectura cada medio grado, pero funciona sobre un mismo plano.

El telémetro láser que se utilizó se presenta en la figura 5.2.

El primer paso para poder utilizar la información generada por el telémetro láser en la construcción de un mapa de disparidad, será integrar el mismo a la geometría estéreo del sistema de visión. A continuación describimos en que consiste este proceso y como lo hemos llevado a cabo.



Figura 5.2: El telémetro láser. La imagen corresponde al modelo SICK PLS 101-312.

5.1.2. Integración del telémetro láser a la geometría estéreo

El problema de integrar el telémetro láser, a la geometría estéreo, consiste en conocer la posición de dicho dispositivo con respecto a la referencia utilizada por el sistema de visión estéreo. Al solucionar este problema, estaremos en posibilidad de trasladar la información de distancia, generada por el telémetro láser, al plano de la imagen de alguna de las cámaras y consecuentemente reflejarla en el mapa de disparidad.

Al igual que con las cámaras, necesitamos establecer perfectamente las coordenadas del telémetro láser con respecto al origen coordenado utilizado en la geometría estéreo, para nuestro caso, el centro de la cámara izquierda. En la figura 5.3 se describen los

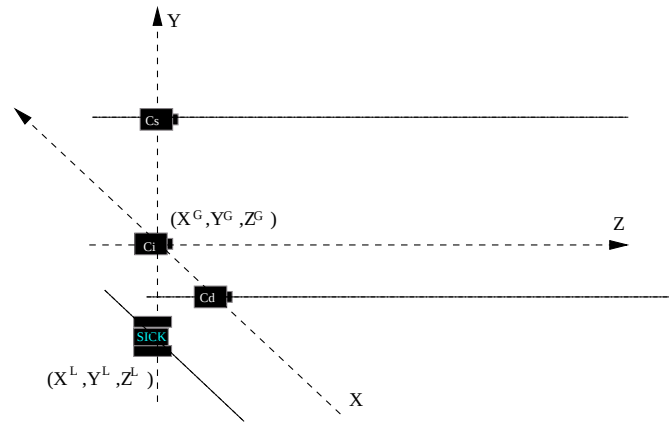


Figura 5.3: Integración del telémetro láser a la geometría estéreo.

elementos que debemos considerar para integrar el telémetro láser a la geometría estéreo. Necesitamos conocer las coordenadas (X^L, Y^L, Z^L) del telémetro láser. Como se ilustra en la misma figura, es importante que tanto el telémetro láser como las cámaras apunten en la misma dirección, es decir, que las lecturas para 0° y 180° del telémetro láser mantengan la misma dirección del eje X del sistema estéreo.

En la figura 5.4 se muestra la geometría que incluye a las tres cámaras del sistema de

visión y al telémetro láser. Para determinar la posición del telémetro láser necesitamos establecer que áreas sobre las imágenes de las cámaras son las que nos interesan. Es necesario que las lecturas generadas por el láser, correspondan a elementos que sean detectados por las tres cámaras, de lo contrario dicha información no será de utilidad. De esta forma, estaríamos descartando de inicio una gran parte de las lecturas láser, ya que la apertura de las cámaras es mucho menor que los 180° que logra cubrir el telémetro láser. La integración del láser al sistema estéreo se hace en forma manual por lo que se trata de una calibración mecánica.

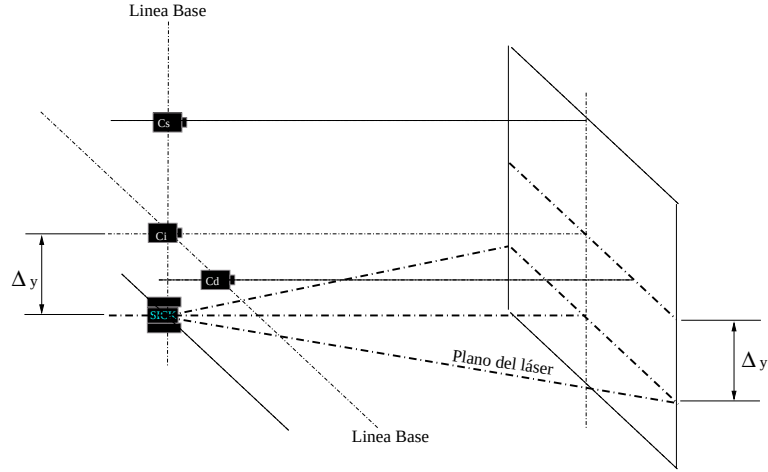


Figura 5.4: En la integración del telémetro láser al sistema estéreo es importante mantener las lecturas láser sobre un plano paralelo a los ejes de proyección de las cámaras. Con esto se busca mantener la misma coordenada Δ_y para todas las lecturas del telémetro láser.

Por otra parte, es importante mantener el plano de las lecturas láser paralelo a los ejes de proyección de las cámaras, ya que esto nos facilitará los cálculos para determinar las coordenadas 3-D que describen a cada una de las lecturas del telémetro láser.

Por lo tanto, para obtener las coordenadas (X_i^L, Y_i^L, Z_i^L) de cada una de las lecturas L_i , del telémetro láser, con respecto a las coordenadas (X^L, Y^L, Z^L) podemos establecer las siguientes ecuaciones:

$$\begin{aligned} X_i^L &= L_i [\cos(i/2)] \\ Y_i^L &= 0 \\ Z_i^L &= L_i [\sen(i/2)] \end{aligned} \tag{5.1}$$

En donde i toma valores de $0, \dots, 360$, con los cuales cubre un rango de $0^\circ, \dots, 180^\circ$. Por lo tanto, las funciones $\sen()$ y $\cos()$ de esta ecuación se calculan utilizando grados.

Finalmente, para calcular las coordenadas (X_i^G, Y_i^G, Z_i^G) que son las coordenadas 3-D de cada una de las lecturas, con respecto al origen utilizado en la geometría estéreo, bastará con sumar a la ecuación 5.1 las coordenadas del telémetro láser:

$$\begin{aligned} X_i^G &= X_i^L + X^L \\ Y_i^G &= Y^L \\ Z_i^G &= Z_i^L + Z^L \end{aligned} \quad (5.2)$$

De esta forma solucionamos el problema de integrar el telémetro láser a la geometría estéreo asumiendo que la calibración mecánica logró que el plano del láser sea paralelo a los ejes ópticos de las cámaras. A continuación se abordará el problema de llevar elementos del mundo 3-D, cuya posición es conocida, al plano de la imagen de una de las cámaras.

5.1.3. Del láser a la imagen

Una vez integrado el telémetro láser a la geometría estéreo, nuestra tarea ahora es convertir cada una de las distancias L_i generadas por el telémetro láser, cuyas coordenadas (X_i^G, Y_i^G, Z_i^G) ya conocemos, a coordenadas (x_i, y_i) en el plano de la imagen de una de las cámaras.

Gracias a que ya corregimos la distorsión en las lentes de las cámaras, a que las imágenes están rectificadas y a que podemos conocer las coordenadas 3-D de las lecturas láser, es que podemos echar mano del llamado *modelo en perspectiva* [23], el cual nos permite convertir coordenadas del mundo 3-D a coordenadas en la imagen.

El modelo en perspectiva

El modelo en perspectiva supone la ausencia de cualquier tipo de distorsión, por lo que también se le conoce como modelo de “pinhole”. Consiste de un plano π (ver figura 5.5) que es el plano de la imagen, y un punto O que representa al centro de proyección. La distancia entre π y O es la distancia focal f . La intersección entre el eje óptico Z y el plano de la imagen es el centro de la imagen o . Si tenemos un punto P , cuyas coordenadas (X_P, Y_P, Z_P) son conocidas y si establecemos que p es la proyección de P en el plano de la imagen, entonces podemos establecer la relación:

$$\begin{aligned} \frac{x}{f} &= \frac{X_P}{Z_P} \\ \frac{y}{f} &= \frac{Y_P}{Z_P} \end{aligned} \quad (5.3)$$

Por lo que podemos obtener:

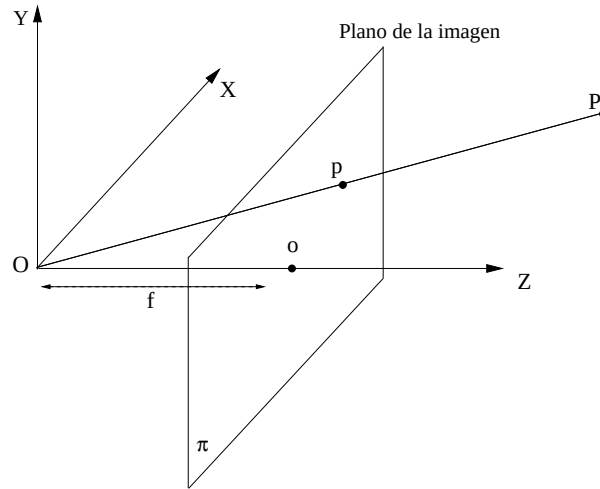


Figura 5.5: El modelo en perspectiva o también llamado modelo de “pinhole”.

$$\begin{aligned}
 x &= f \frac{X_P}{Z_P} \\
 y &= f \frac{Y_P}{Z_P} \\
 d &= \left. \vphantom{\begin{aligned} x &= f \frac{X_P}{Z_P} \\ y &= f \frac{Y_P}{Z_P} \end{aligned}} \right\} \frac{fT}{Z}
 \end{aligned} \tag{5.4}$$

donde x y y son las coordenadas de p en la imagen π y la disparidad (d) se obtiene a partir de la ecuación 4.4.

De esta forma podemos ubicar perfectamente en la imagen la información generada por el telémetro láser.

En la figura 5.6 mostramos dos ejemplos de las conversiones de las lecturas láser a coordenadas en la imagen utilizando la ecuación 5.4. En ambas imágenes se superponen pixeles con un tono de gris contrastante para poder ser apreciados. Cada pixel representa una de las lecturas obtenidas con el telémetro láser.

A continuación se describe como, a partir de la información inicial generada por el telémetro láser, podemos construir un mapa de disparidad.

5.1.4. Propagación en el mapa de disparidad.

En esta sección utilizaremos la información generada por el telémetro láser dada en las “semillas de disparidad”, las cuales se propagarán por todo el mapa de disparidad,



Figura 5.6: Dos ejemplos de conversión de las lecturas láser al plano de la imagen.

en base a un criterio de similaridad entre los vecinos de las mismas en la imagen¹.

La idea de la propagación está basada en que los pixeles contiguos, sobre todo cuando se trata de zonas homogéneas de la imagen, muy probablemente pertenecen a una misma superficie por lo que resulta conveniente que la disparidad sea contagiada y que de no ser así, deberán estar separados por un borde, el cual puede ser detectado, lo que impedirá que la propagación se extienda más allá de lo deseado. Aunque esto no siempre sucede, en la mayoría de los casos podemos aprovechar esta característica para expandir o propagar la información generada por el láser.

El proceso de propagación que implementamos consiste en almacenar las semillas en una lista. A continuación cada semilla irá generando más elementos para la lista y la forma en que serán almacenados y accedidos estos nuevos elementos ejemplifica una búsqueda en anchura en una estructura tipo árbol[19].

Partiendo de la idea de que nuestro sistema de visión estéreo es trinocular, es decir que contamos con tres tomas de la misma escena (ver figura 5.4), y que además conocemos la disparidad de un conjunto de puntos en la imagen, generados por el láser. Podemos enumerar el proceso de propagación de la siguiente forma:

1. Colocamos en una lista L los puntos $p_1, p_2, p_3, \dots, p_n$ generados con el láser, almacenando sus coordenadas (x_i, y_i) , su tono original en la imagen izquierda (t_i) y su disparidad (d_i).
2. A continuación consideraremos una matriz M del mismo tamaño de las imágenes, inicializando todos sus elementos en cero. En nuestro caso utilizamos imágenes de 640×480 pixeles.

¹En nuestras pruebas, de 50 a 70 lecturas láser resultaban de utilidad y este número varía notablemente de imagen a imagen (ver figura 5.6)

3. Tomamos un punto p_i de la lista L y evaluamos si alguno de sus vecinos v_p se le parece lo suficiente como para considerarlo un candidato para la propagación. Por lo tanto deberemos establecer un criterio de similaridad entre ellos. Diferencias de hasta 15 unidades (en una escala de 0 a 255) pueden arrojar buenos resultados. Por lo tanto nuestra primer prueba de propagación la llamaremos *prueba de textura* y será superada siempre y cuando $|t_p - t_v| < Cs$, donde Cs es el criterio de similaridad definido previamente y t_p y t_v son el tono de gris de la actual semilla y de su pixel vecino respectivamente. En caso de no superar esta prueba, el pixel v_p es desechado para la propagación de p_i y se continua con el siguiente vecino de p_i . En nuestro caso consideramos hasta cuatro vecinos (arriba, abajo, izquierda y derecha).
4. En caso de que v_p supere la prueba de similaridad, será sometido a una segunda prueba, la cual llamaremos *prueba de correspondencia horizontal*. Considerando que la proyección del láser la hicimos sobre la imagen izquierda, entonces utilizaremos d_i para buscar la correspondencia de v_p en la imagen derecha. En la sección 4.3 de la página 68 ya describimos como llevar a cabo este proceso. La diferencia es que ahora ya conocemos la ubicación de la posible correspondencia a través de la disparidad d_i , por lo que solamente tendremos que verificar que satisfaga los requerimientos para hacer match. En caso de no cumplir este segundo criterio, el pixel v_p es desechado y se continua con el siguiente vecino. En caso de ya no haber vecinos se continúa con el siguiente elemento de la lista L .
5. Una vez superado el criterio de similaridad y el criterio de correspondencia optamos por una tercera etapa la cual llamaremos *prueba de correspondencia vertical*, y con ella trataremos de confirmar que v_p debe ser contagiado por p_i . En esta tercera etapa se hace otra prueba de correspondencia pero ahora con la cámara superior. El criterio es el mismo aunque quizá sea necesario recalcular el valor para d_i . Tanto en esta etapa como en la anterior se puede establecer una región de búsqueda alrededor del punto marcado por d_i , con ello reduciremos el impacto producto de los errores de medición de los cuales difícilmente estamos exentos. Aquellos puntos que no superan esta tercera etapa son desechados para la propagación de p_i y se continúa con el siguiente vecino de p_i , en caso de no haber se continúa con el siguiente elemento de la lista L .
6. Finalmente, si v_p ha superado la tercera etapa entonces es contagiado por p_i , y se marca la posición (x_v, y_v) de la matriz M para ya no volverlo a evaluar como vecino y se agrega v_p contagiado con la disparidad de p_i al final de la lista L .
7. Se continúan evaluando los vecinos de p_i o en su defecto se continúa con el siguiente de la lista. El proceso termina con el último elemento de la lista L .

En la figura 5.7 mostramos un ejemplo de la propagación utilizando 8 y 4 vecinos para cada pixel. Utilizar 4 vecinos para cada semilla permite que la propagación se haga más

equilibrada lo que produce diferencias notorias en la dirección que sigue la propagación así como en la superficie cubierta. Para esta investigación resultó más conveniente utilizar solamente 4 vecinos, ya que la propagación se extendió en forma de capas lo que puede traducirse en información de distancia, en cambio al utilizar 8 vecinos la propagación se extendió caprichosamente ramificándose de tal forma que dificultaría posteriormente la etapa de reconstrucción.

El método de propagación utilizando el telémetro láser es una herramienta ideal para combatir el problema de las superficies homogéneas. Sin embargo, al utilizar el criterio de textura, los píxeles que aparecen como bordes quedan excluidos de cualquier análisis. De hecho los bordes funcionan como contenedores de la propagación y por tal motivo se reflejan como elementos sin información en el mapa de disparidad.

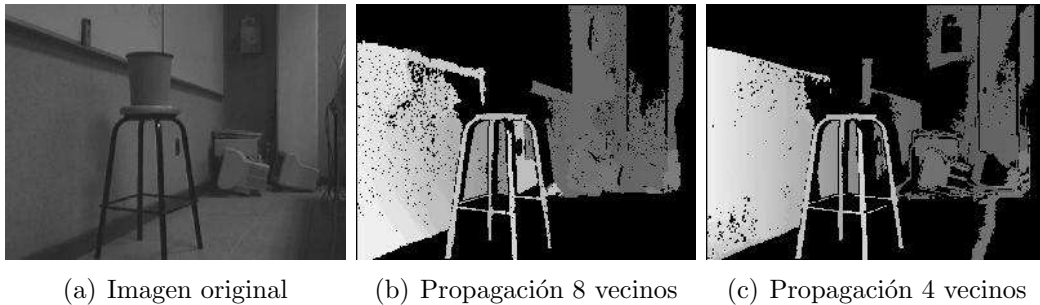


Figura 5.7: Resultados de la propagación considerando solamente dos de las tres etapas mencionadas anteriormente, la prueba de verificación no se llevó a cabo. El comportamiento del algoritmo considerando sus 8 vecinos inmediatos (b) es apreciablemente diferente a si solamente consideramos a cuatro de ellos (c).

Sin embargo, el problema de la detección de disparidad en los bordes se analizó en la sección 4.3.1 de la página 72, en donde se utilizó el algoritmo de Shirai para solucionarlo.

A continuación presentamos la estrategia final que utilizamos para generar un mapa de disparidad, la cual esta integrada por la suma de los mapas generados con el algoritmo de Shirai y el método de propagación.

5.1.5. Propagación y Correlación

Analizando las ventajas y desventajas que presentan los métodos de propagación y correlación, podemos aprovecharlas para integrar un único mapa de disparidad que combine la información de ambas. El algoritmo de Shirai se encargará de detectar la disparidad para aquellos elementos que se consideren como bordes en la imagen y por otra parte, el método de propagación aprovecha una pequeña cantidad de información inicial para propagarse por todas aquellas regiones que presenten cierta similitud.

De esta forma, al combinar los resultados de ambos métodos en un mismo mapa de disparidad, podemos obtener la información necesaria para llevar a cabo las tareas de reconstrucción. En la figura 5.8 mostramos dos mapas de disparidad generados por cada uno de los métodos mencionados y una tercera imagen donde encajamos ambos mapas los cuales son excluyentes, es decir, la información generada en el mapa de disparidad obtenido por el método de Shirai corresponde a elementos sin información en el mapa de disparidad que se obtuvo en el método de propagación y viceversa.

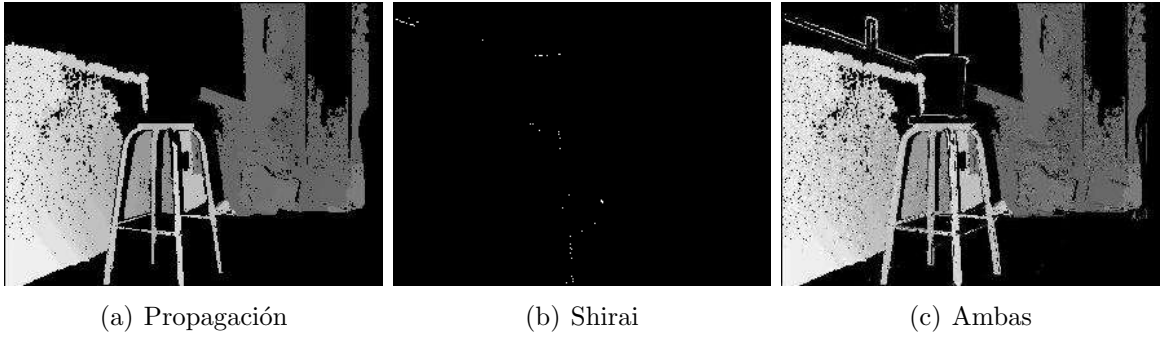


Figura 5.8: Aprovechando las ventajas de las estrategias de propagación láser (a), y de la correlación de Shirai (b), se puede obtener un mapa de disparidad con mayor información (c).

Por último, mencionaremos que fue introducida una modificación en el algoritmo de Shirai para reducir la cantidad de errores en el mapa de disparidad. Esta modificación, considera que una vez que ya se detectó el match m_1 , volvemos a aplicar el mismo procedimiento para detectar un nuevo match pero ahora se hace la búsqueda en sentido contrario y con las imágenes intercambiadas, es decir, si m_1 indicaba cuantos pixeles se había desplazado un punto $p_1(x, y)$ en la imagen derecha a un punto $p_2(x, y + m_1)$ en la imagen izquierda, ahora buscaremos la correspondencia de p_2 en la imagen izquierda con su correspondiente mejor match p'_2 en la imagen derecha. Si p'_2 y p_1 son el mismo punto estamos hablando de que m_1 es una disparidad correcta.

Con la unión del método de propagación y del algoritmo de Shirai, terminamos los procesos relacionados con la construcción de mapas de disparidad. A continuación, utilizaremos los mapas de disparidad para generar mapas del entorno 3-D, este proceso es conocido como *reconstrucción* y representa la parte final de esta tesis.

5.2. Reconstrucción

En esta sección llevaremos a cabo la construcción de mapas del entorno 3-D, a partir de la información contenida en un mapa de disparidad. Los mapas del entorno serán

construidos en base a las necesidades de un robot móvil y con el objetivo de que éste pueda realizar tareas de navegación, como por ejemplo, evadir obstáculos.

5.2.1. De la imagen al mundo real

Una vez que ya contamos con el mapa de disparidad y que ya conocemos los parámetros intrínsecos y extrínsecos de nuestro sistema estéreo, estamos en condiciones de reconstruir el entorno 3-D, del cual fueron obtenidas las imágenes. Al contrario de como lo hicimos con el láser en la sección anterior, ahora partiremos de un punto P_i en la imagen con coordenadas (x_i, y_i, d_i) en donde d_i es la disparidad de p_i y buscaremos encontrar las coordenadas (X_P, Y_P, Z_P) del objeto correspondiente.

Para ello utilizaremos las ecuaciones 4.4 y 5.4 de las cuales podemos obtener las siguientes relaciones:

$$\begin{aligned} Z_P &= f \frac{T}{d_i} \\ X_P &= (o_x - x_i) \frac{Z_P}{f} \\ Y_P &= (o_y - y_i) \frac{Z_P}{f} \end{aligned} \tag{5.5}$$

En donde T es la distancia entre los centros de proyección de las cámaras, d_i es la disparidad para el punto (x_i, y_i) y o_x y o_y son el centro de la imagen.²

La figura 5.9 nos muestra como las coordenadas (X_P, Y_P, Z_P) de un punto P , están relacionadas con su proyección p_i en el plano de la imagen y con el centro de proyección f de una de las cámaras.

Utilizando las ecuaciones 5.5 estamos en condiciones de reconstruir el entorno 3-D, partiendo de la información extraída de las imágenes estéreo. La construcción de un mapa 3-D puede resultar una tarea compleja si consideramos los aspectos gráficos que esto involucra. Sin embargo, considerando que el objetivo de estos mapas es ayudar en las tareas de navegación de un robot, y más específicamente, en proporcionarle la información necesaria para evadir obstáculos, podemos considerar solamente aquellos elementos que sean útiles para este fin.

A continuación se llevará a cabo la construcción de un mapa en dos dimensiones que permita, a un robot móvil, tomar decisiones para moverse en su entorno sin chocar.

5.2.2. Construcción de un mapa del entorno

La principal preocupación de la robótica móvil, es contar con los elementos necesarios para llevar a cabo la construcción de mapas confiables y poder navegar a partir de

²Recordar que estas ecuaciones están en función del modelo de “pinhole” en el cual no existe distorsión.

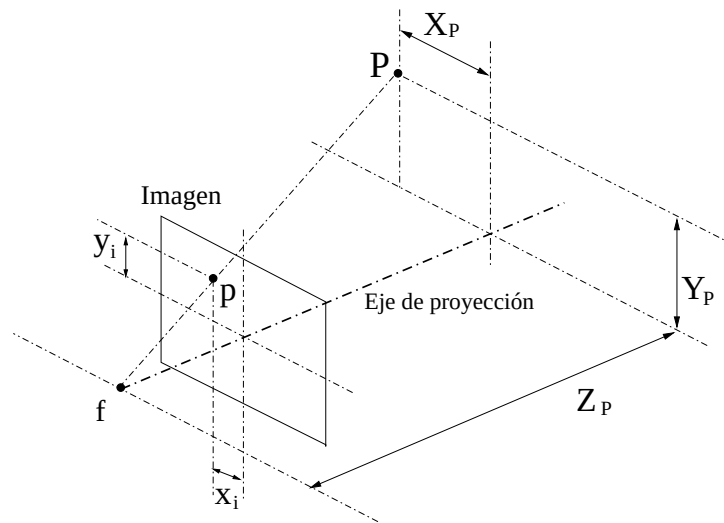


Figura 5.9: Las coordenadas (X_P, Y_P, Z_P) de P , pueden obtenerse a partir de las coordenadas de su proyección en la imagen $p_i(x, y)$ y de la disparidad generada para este punto.

la información que obtiene de ellos. En nuestro caso, los recursos que utiliza el robot para llevar a cabo la construcción de un mapa, el cual hemos estado llamando *mapa del entorno*, son el sistema de visión estéreo y un telémetro láser. El mapa del entorno es una imagen, que refleja la posición de los elementos del ambiente que rodean al robot, la cual se modifica cada vez que el robot se mueve y toma nuevas imágenes. No es necesario que el robot sepa donde ha estado con anterioridad sino más bien a qué distancia se encuentran los obstáculos ahora. Un mapa de estas características es suficiente para que el robot pueda llevar a cabo su tarea, la cual implica que se pueda mover sin colisionar. Podemos citar algunas aplicaciones que justifican la elaboración de mapas del entorno [16] :

- Vehículos inteligentes, ya sea en la forma de asistentes para el conductor del vehículo o vehículos sin operador humano
- Vigilancia e inspección en la industria, por ejemplo en el monitoreo de áreas radioactivas de plantas nucleoelectricas;
- En ambientes de difícil acceso, por ejemplo en minas,
- En el espacio o bajo el agua, por ejemplo el robot Spirit 5.10 llegó al planeta Marte en enero del 2004, este robot puede ser teleoperado desde la tierra a partir de reconstrucciones 3-D que realiza utilizando visión estéreo.
- En la desactivación de bombas y minas

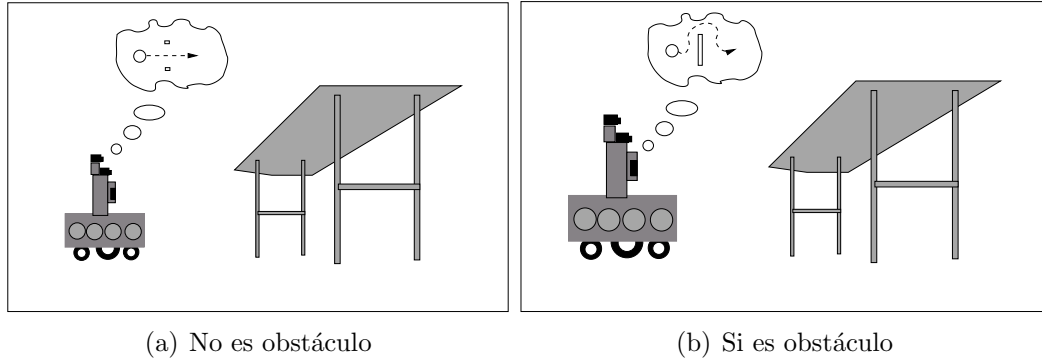


Figura 5.11: El robot deberá reflejar en el mapa del entorno solamente aquellos objetos, o las partes de ellos, que representan un obstáculo tomando en cuenta la geometría del robot.

2. Se define un vector V de n elementos, en el cual almacenaremos las distancias de los obstáculos más cercanos, en donde Z_P es la distancia que se almacenará, Y_P determinará a que altura está el elemento y por lo tanto si se trata de un obstáculo o no y X_P será mapeado a uno de los n elementos del vector V y determinará la posición de Z_P en dicho vector, en donde por ejemplo, $V(n/2)$ contendrá la distancia a la que se encuentra un obstáculo que esté directamente enfrente del robot. En la figura 5.13 se explica como hacer esto.
3. En el caso de que Y_P quede por encima o por debajo de lo que llamaremos *la zona de choque del robot* (ver figura 5.12), $P(X_P, Y_P, Z_P)$ es descartado para el mapa del entorno. La zona de choque está definida por Y_{max} y Y_{min} que representan el punto más alto del robot y el piso respectivamente.
4. Si se generan dos o más valores para una misma posición en V , siempre se elegirá la más pequeña, ya que esto indicará que dos objetos comparten el mismo parámetro X_P , aunque posiblemente con diferentes alturas Y_P y diferentes distancias Z_P , por lo que elegiremos aquel que esté más cerca. Considerando las dimensiones del robot, una distancia Z_P demasiado pequeña indicará la presencia de un obstáculo, por lo que el robot tendrá que aplicar alguna estrategia para evadirlo.

El resultado final es una especie de vista aérea del entorno del robot, a través de la cual podremos aplicar alguna estrategia de navegación. La construcción de nuestro mapa del entorno puede verse como si aplastáramos todos aquellos objetos que están dentro de la zona de choque, por lo que nuestro mapa se ve reducido a un plano bidimensional mediante el cual se hace más fácil la navegación. En la figura 5.14 se muestra un ejemplo de un mapa generado utilizando este principio. El mismo mapa es detallado y explicado en

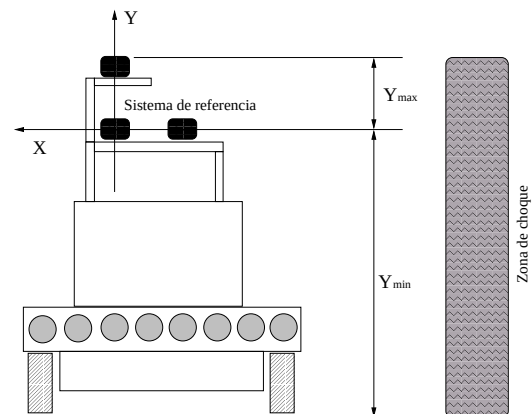


Figura 5.12: Para delimitar la zona de choque, los valores Y_{max} y Y_{min} se establecen en base un sistema de referencia que en este caso es una de las cámaras.

la figura 5.15. El mapa bidimensional permite al robot aplicar estrategias de navegación confiables al conocer los obstáculos del entorno.

En el siguiente capítulo se presentan las conclusiones acerca de los resultados obtenidos durante la calibración del sistema de visión estereó y durante la etapa de construcción de los mapas de disparidad y del entorno. También se presentan las sugerencias para trabajos futuros.

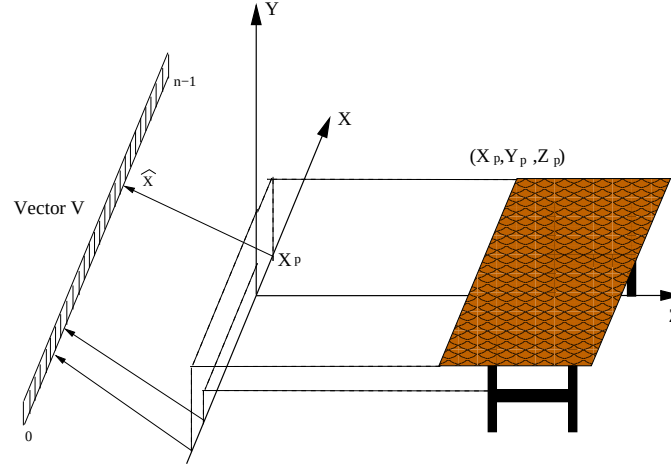


Figura 5.13: Los elementos del entorno que comparten la misma posición X , utilizando la misma referencia que la geometría estéreo, son discretizados y enviados al mismo elemento $V(\hat{X})$ del vector de distancias, en donde se elige al más cercano. $\hat{X}$ es X_p después de haber sido mapeada a un valor discreto que utilizaremos como índice de V y por lo tanto deberá cumplir $0 \leq \hat{X} \leq (n - 1)$.



(a) Imagen original



(b) Mapa de disparidad



(c) Mapa del entorno

Figura 5.14: A partir de tres imágenes de la misma escena de las cuales incluimos una de ellas (a), se genera en primer lugar un mapa de disparidad (b) a través del proceso de propagación y correspondencia. Por último se obtiene un mapa del entorno(c).

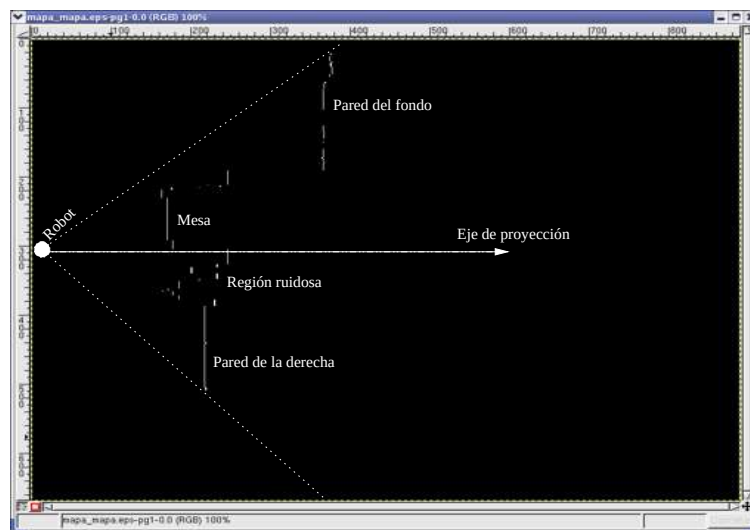


Figura 5.15: El mapa se muestra a detalle y está basado en la escena de la figura 5.14(a). El robot tiene la suficiente información para decidir cuando está en peligro de hacer colisión.

Capítulo 6

Conclusiones

En esta tesis se han descrito algunos de los problemas a los que se enfrentan los sistemas de visión estéreo, por un lado la calibración, que incluye la corrección de la distorsión y por otro lado, el problema de reconstrucción en el que se incluyó el problema de la correspondencia entre imágenes y la forma de integrar las mediciones de un telémetro láser. En este capítulo se presentan las conclusiones de esta investigación y se incluyen también algunas sugerencias para extenderla.

6.1. Conclusiones

En esta investigación, se abordaron los problemas de calibración de un sistema de visión estéreo trinocular y de reconstrucción, utilizando la visión estéreo y un telémetro láser. El resultado final es la obtención de un mapa del entorno que le pueda ser útil a un robot para llevar a cabo tareas de navegación. Para llevar a cabo esta investigación utilizamos un robot móvil con capacidad de movimiento el cual esta equipado con un sistema de visión compuesto de tres cámaras de tipo industrial y un telémetro láser de gran precisión con capacidad de generar lecturas en un rango de 180° . Durante este trabajo utilizamos dos tipos de lentes para las cámaras, unos lentes de 0,4 cm de distancia focal, y otros lentes de 0,21 cm. de distancia focal.

Se llevó a cabo la implementación de dos métodos para corregir la distorsión, uno denominado autocalibración y otro basado en el registro de imágenes. Se utilizó correlación para generar mapas de disparidad con ayuda de un método de propagación basado en el telémetro láser. A partir del mapa de disparidad se generaron mapas del entorno, útiles para que el robot pudiera llevar a cabo una sencilla tarea de navegación consistente en navegar en interiores sin colisionar.

Podemos resaltar las siguientes contribuciones llevadas a cabo en esta investigación:

1. Se integró el telémetro láser a un sistema estéreo. Utilizando las lecturas de distancia, generadas por el láser, como semillas de propagación se llevó a cabo la construcción de los mapas de disparidad.

2. En el caso de utilizar lentes de mayor apertura, se hizo necesario integrar un tercer coeficiente de distorsión al algoritmo presentado por Leonardo Romero y Félix Calderón [17], evitando así el tener que ajustar el algoritmo con valores iniciales. De la misma forma se mejoraron los resultados al agregar el tercer coeficiente de distorsión a que si solamente se utilizan dos.
3. Se definió una estrategia para estimar el valor de la distancia focal en base a imágenes estéreo.
4. Se hicieron modificaciones al algoritmo propuesto por Shirai en los criterios de búsqueda de correspondencias, lo que permitió reducir el tiempo del proceso.
5. Se estableció una rectificación de las imágenes estéreo a partir del algoritmo de calibración presentado en [17].

En resumen, hemos presentado un método de calibración, que ha sido capaz de generar las condiciones apropiadas para llevar a cabo tareas de reconstrucción a partir de imágenes estéreo, destacando la introducción del telémetro láser como una herramienta útil para resolver algunos problemas de correspondencia de imágenes.

6.2. Trabajos futuros

Algunas de las líneas de investigación, que estamos en condiciones de sugerir, para extender o completar nuestra investigación son:

- Desarrollar estrategias de navegación más robustas, ya que en nuestro caso, la tarea que lleva a cabo el robot es muy sencilla al simplemente evitar chocar con obstáculos. Por ejemplo, si el robot almacena un historial de los lugares en los que ya ha estado permitirá una navegación continua al no tener que recalcular los mapas de navegación. Incluso, en ambientes estáticos, podría guardar los movimientos necesarios para ir de un lugar a otro realizando un mínimo de cálculos.
- Integrar al sistema de visión un mecanismo para detectar cristales, ya que ni las cámaras ni el telémetro láser nos garantizan que los puedan incluir en el mapa de disparidad. Una alternativa que se puede considerar es el uso de un anillo de sonares, los cuales pueden detectar este tipo de materiales.
- Construir un mecanismo de autocalibración que le permita al robot calibrarse por si solo, actualmente este proceso se lleva a cabo en forma manual, teniendo que desplazar al robot y alinearlos frente a un patrón de calibración. Una propuesta es que el robot pudiera localizar por si solo el patrón de calibración y alinearse frente a este para llevar a cabo el proceso de calibración. También sería útil un mecanismo que le permita al robot autoajustar sus cámara dependiendo del tipo de iluminación del entorno en el que se encuentre.

- Definir una estrategia que le permita al robot detectar desniveles en pisos uniformes, como por ejemplo escaleras. Actualmente el robot deshecha al piso como posible obstáculo lo que hace riesgosa su navegación.
- Optimizar el tiempo para la generación de los mapas del entorno. Una sugerencia es implementar algunas rutinas utilizando instrucciones del tipo SIMD (Single Instructions Multiple Data) que implementan los procesadores para agilizar los cálculos multimedia. Actualmente toda la programación esta hecha en Ansi C y para llevar a cabo un mapa completo del entorno se toma de 10 a 15 segundos dependiendo del entorno en el que se encuentre. Se espera que con la codificación a bajo nivel se pueda reducir a menos de la cuarta parte.
- Utilizando los mapas del entorno que va generando el robot al moverse, construir un mapa global que le permitiera al robot poder recordar lugares y por lo tanto tomar mejores decisiones en la navegación.
- Ampliar el alcance a ambientes dinámicos sincronizando la captura de las cámaras. En este trabajo se asumieron solamente ambientes interiores estáticos. El problema de los ambientes dinámicos es que generan mucho ruido durante la construcción de los mapas de disparidad.
- Agregar una tercera cámara al robot permite construir mapas con una mayor cantidad de información, pero también implica prácticamente duplicar los cálculos para generar dichos mapas. Un trabajo futuro es fortalecer los algoritmos para generar mapas de navegación utilizando únicamente dos imágenes y el telémetro láser.

Referencias

- [1] Unibrain S. A. Unibrain the 1394 innovators. <http://www.unibrain.com/>, August 2003.
- [2] K. O. Arras, N. Tomatis, and R. Siegwart. Multisensor on-the-fly localization using laser and vision. In *Proceedings of the 2000 IEEE/RSJ International Conference on Intelligent Robots and Systems*, May 2000.
- [3] F. Calderón. Curso de procesamiento de imágenes. Maestría en Ingeniería Eléctrica, January 2003.
- [4] S. C. Chapra and R. P. Canale. *Métodos numéricos para ingenieros*. Mc. Graw Hill, Oct 2000.
- [5] A. Chávez. *Fórmulas de matemáticas*. SEP-SEIT, 1995.
- [6] F. Devernay and O. Faugeras. Automatic calibration and removal of distortion from scenes of structured environments. In *SPIE*, volume 2567, San Diego, CA, July 1995.
- [7] Real Academia Española. Diccionario de la lengua española, vigésima segunda edición. <http://www.rae.es>, Agosto 2005.
- [8] D. A. Forsyth and J. Ponce. *Computer Vision - A modern approach*. Pearson Education Inc., 2003.
- [9] D. Hearn and M. P. Baker. *Gráficas por computadora*. Prentice Hall Hispanoamericana S.A., 1995.
- [10] H. Hirschmüller, P. R. Innocent, and J. Garibaldi. Real-time correlation-based stereo vision with reduced border error. In *International Journal of Computer Vision*, volume 47, pp. 229-246, Centre for Computational Intelligence, De Montfort University, Leicester , U.K., LEI 9BH, 2002.
- [11] S.L. INFAIMON. Criterios de calidad de las opticas. <http://www.infaimon.com/catalogo/industria/optica/teoricos/criterios.htm>, August 2003.

- [12] L. Iocchi and K. Konolige. A multiresolution stereo vision system for mobile robots. In *AIIA (Italian AI Association) Workshop*, September 1998.
- [13] R. Klette, K. Schlüns, and A. Koschan. *Computer Vision, Three-dimensional Data from Images*. Springer-Singapore, 1998.
- [14] L. Monje. Curso de fotografía. <http://www.difo.uah.es/curso/>, August 2003.
- [15] E. J. Purcell and D. Varberg. *Cálculo con Geometría Analítica*. Prentice Hall Hispanoamericana S.A., 1987.
- [16] L. Romero. *Construcción de mapas y localización de robots móviles: un enfoque probabilista*. ITESM, campus Cuernavaca, Mayo de 2002.
- [17] L. Romero and F. Calderón. Correcting radial lens distortion using image and point correspondences. Proceedings of the 8th Iberoamerican Congress on Pattern Recognition, CIARP2003, November 2003.
- [18] Leonardo Romero, Adrián Núñez, Sergio Bravo, and L. E. Gamboa. Fusing a laser range finder and a stereo vision system to detect obstacles in 3d. In *Iberamia 2004*, volume pp. 555-561, Springer Verlag, LNAI 3315, 2004.
- [19] S. Russell and P. Norving. *Artificial Intelligence: A Modern Approach*. Prentice Hall, 2002.
- [20] R. Szeliski. Video mosaics for virtual environments. *IEEE Computer Graphics and Applications* pp. 22-30, March 1996.
- [21] T. Tamaki, T. Yamamura, and N. Ohnishi. An automatic camera calibration method with image registration technique. In *IIIS Proc. of SCI2000*, volume V, pages 317–322, Nagoya Japan, July 2000.
- [22] T. Tamaki, T. Yamamura, and N. Ohnishi. Correcting distortion of image by image registration. In *ACCV2002, The 5th Asian Conference on Computer Vision*, Melbourne, Australia, January 2002.
- [23] E. Trucco and A. Verri. *Introductory Techniques for 3-D Computer Vision*. Prentice Hall, 1998.
- [24] G. Olague y R. Ramírez. Síntesis de imágenes a partir de fotografías. In *Conacyt México*, volume contrato 35267-A, Departamento de ciencias de la computación, División de Física Aplicada, Ensenada Baja California, October 2001.

Apéndice A

Transformaciones espaciales

Las transformaciones espaciales aplicadas a imágenes nos permiten que un punto con coordenadas (x, y) se traslade a una nueva posición en la imagen (x', y') . Cuando hablamos de imágenes estas transformaciones se pueden definir como *transformaciones bidimensionales* [9].

Las transformaciones espaciales se pueden dividir en:

1. Traslación
2. Rotación
3. Escalamiento
4. Compuesta

Enseguida se describirán brevemente las transformaciones de traslación, rotación, escalamiento y las transformaciones compuestas de las tres primeras. Sin embargo, antes de abordar las transformaciones compuestas se introducirán las coordenadas homogéneas, herramienta matemática muy útil para modelar de forma simple la composición de transformaciones utilizando transformaciones matriciales.

A.1. Traslación

La traslación de un punto $P(x, y)$ a una nueva posición $P'(x', y')$ se puede expresar como:

$$\begin{aligned}x' &= x + T_x \\ y' &= y + T_y\end{aligned}\tag{A.1}$$

En donde T_x y T_y representan la traslación en los ejes X y Y respectivamente. Lo anterior se puede representar en una sola ecuación matricial utilizando vectores para representar las posiciones de coordenadas y el vector de traslación:

$$P' = \begin{bmatrix} x' \\ y' \end{bmatrix}, \quad P = \begin{bmatrix} x \\ y \end{bmatrix}, \quad T = \begin{bmatrix} T_x \\ T_y \end{bmatrix} \quad (\text{A.2})$$

De tal forma que las dos ecuaciones A.1 se pueden expresar en forma matricial como:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} T_x \\ T_y \end{bmatrix} \quad (\text{A.3})$$

O bien, utilizando una notación más compacta, como:

$$P' = P + T \quad (\text{A.4})$$

La figura A.1 ilustra la traslación de un punto sobre un plano.

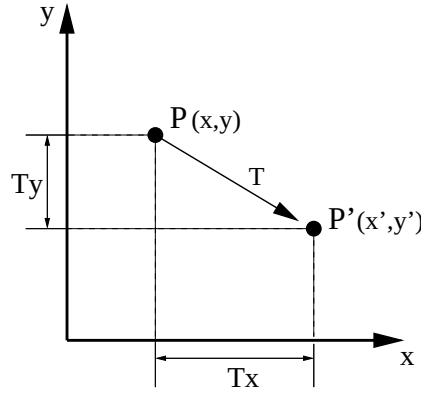


Figura A.1: La traslación se puede representar mediante desplazamientos T_x y T_y en los ejes X y Y respectivamente, o simplemente a través del vector de traslación T .

A.2. Rotación

Se aplica una rotación bidimensional en un objeto al cambiar su posición a lo largo de la trayectoria de una circunferencia en el plano de XY [9].

Para una rotación es necesario especificar un ángulo de rotación θ y la posición (x_r, y_r) de un punto de rotación en torno al cual se gira el objeto.

La rotación de un punto $P(x, y)$ utilizando como centro de rotación al origen $O(0, 0)$ nos lleva a un nuevo punto $P'(x', y')$ en donde la transformación ahora se puede expresar como una medida angular θ . En la figura A.2 se ilustra lo anterior.

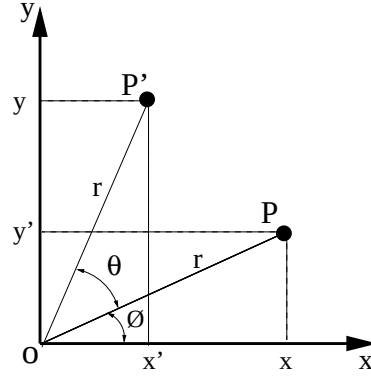


Figura A.2: La rotación

En este caso r es la distancia común de los puntos P y P' al origen O .

De la figura se infiere:

$$\begin{aligned} x' &= r \cos(\theta + \phi) = r \cos(\theta) \cos(\phi) - r \sin(\theta) \sin(\phi) \\ y' &= r \sin(\theta + \phi) = r \cos(\theta) \sin(\phi) + r \sin(\theta) \cos(\phi) \\ r &= \sqrt{x^2 + y^2} \end{aligned} \quad (\text{A.5})$$

Donde:

$$x = r \cos(\phi) \quad (\text{A.6})$$

$$y = r \sin(\phi) \quad (\text{A.7})$$

Sustituyendo A.6 y ?? en A.5 nos da:

$$\begin{aligned} x' &= x \cos(\theta) - y \sin(\theta) \\ y' &= x \sin(\theta) + y \cos(\theta) \end{aligned} \quad (\text{A.8})$$

Si utilizamos la notación vectorial para los puntos P' y P , entonces A.8 puede expresarse también en forma matricial:

$$P' = R \cdot P \quad (\text{A.9})$$

Donde la matriz de rotación R es

$$R = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \quad (\text{A.10})$$

Por lo que finalmente, la ecuación para la rotación utilizando una representación matricial es:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \quad (\text{A.11})$$

A.3. Escalamiento

Escalar un punto $P(x, y)$ para obtener un nuevo punto $P'(x', y')$ implica multiplicar por un factor s (factor de escala) a cada uno de sus componentes. Esto es:

$$\begin{aligned} x' &= s_x x \\ y' &= s_y y \end{aligned} \quad (\text{A.12})$$

Donde el factor de escala s_x escala objetos en dirección x y el factor de escala s_y lo hace en la dirección y . También se pueden expresar las ecuaciones de transformación A.12 en la forma matricial[9]:

$$P' = S \cdot P \quad (\text{A.13})$$

Por lo que podemos expresar:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

A continuación se introducirá el concepto de coordenadas homogéneas, esto con el objetivo integrar las transformaciones espaciales como un producto matricial.

A.4. Coordenadas homogéneas

En esta sección se plantea la necesidad de formular nuevamente las ecuaciones para las transformaciones espaciales, de modo que se puedan procesar de manera eficiente.

Se pueden combinar las transformaciones espaciales en una sola representación de matriz, esto se logra ampliando las representaciones de matrices de 2 por 2 a matrices de 3 por 3. De esta forma podemos expresar las transformaciones de traslación, rotación y escalamiento como multiplicaciones de matriz, siempre y cuando también ampliemos las representaciones de matriz para las coordenadas de P y P' [9].

Al utilizar coordenadas homogéneas en lugar de coordenadas cartesianas ($P(x, y)$), se hace necesario agregar un parámetro más. El punto P en coordenadas homogéneas está representado como $P(x, y, h)$ y la relación entre las coordenadas homogéneas y las coordenadas reales se define como:

$$\begin{aligned} x &= \frac{x_h}{h} \\ y &= \frac{y_h}{h} \end{aligned} \quad (\text{A.14})$$

Donde h puede tomar cualquier valor diferente de cero, por lo que existen múltiples representaciones de un mismo punto. Sin embargo, por simplicidad se utiliza $h = 1$.

Expresar posiciones en coordenadas homogéneas nos permite expresar todas las ecuaciones de transformaciones espaciales como multiplicaciones de matrices [9].

En base a lo anterior, las matrices de transformación utilizando coordenadas homogéneas, quedarían de la siguiente forma:

Para la traslación:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & T_x \\ 0 & 1 & T_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Para la rotación alrededor del origen:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Para la transformación escalar:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Nótese que la traslación está expresada ahora como una multiplicación matricial y antes era una suma.

A.5. Transformación compuesta

Una transformación compuesta es una combinación de otras transformaciones como pueden ser la traslación, la rotación o el escalamiento. Un ejemplo de ello es la rotación con respecto a un punto arbitrario (figura A.3), para hacerlo primeramente tendríamos que trasladar el centro de rotación o_r al origen de sistema de referencia o , rotar el punto $P(x, y)$, y posteriormente trasladar el centro de rotación a su posición original $o_r(x_r, y_r)$.

Si utilizamos la representación matricial de cada transformación, podemos representar la transformación compuesta como un triple producto matricial:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & x_r \\ 0 & 1 & y_r \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -x_r \\ 0 & 1 & -y_r \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Incluso se pueden premultiplicar las matrices de transformación y obtener una sola que las represente. Dicha multiplicación arrojaría el siguiente resultado:

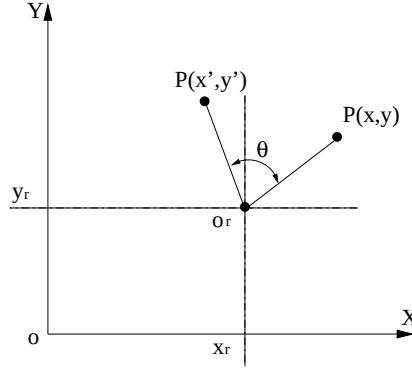


Figura A.3: La rotación sobre un punto arbitrario es una combinación de tres transformaciones

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & x_r(1 - \cos(\theta)) + y_r \sin(\theta) \\ \sin(\theta) & \cos(\theta) & -x_r \sin(\theta) + y_r(1 - \cos(\theta)) \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

En [9] se menciona que al producto de matrices de transformación también se le conoce como concatenación o composición de matrices.

A.6. Transformación Afín

Una transformación afín es cualquier combinación lineal de las transformaciones básicas como la rotación, la traslación y el escalamiento[9].

Una de las propiedades de las transformaciones afín es que las líneas paralelas se transforman en líneas paralelas y los puntos finitos en puntos finitos [9].

La transformación afín se puede representar a través de la siguiente ecuación:

$$\hat{r} = Ar + b \quad (\text{A.15})$$

la cual es también conocida como el *Modelo Afín*
en donde:

$$A = \begin{bmatrix} \theta_0 & \theta_1 \\ \theta_3 & \theta_4 \end{bmatrix} \quad b = \begin{bmatrix} \theta_2 \\ \theta_5 \end{bmatrix} \quad r = \begin{bmatrix} x \\ y \end{bmatrix} \quad \hat{r} = \begin{bmatrix} x' \\ y' \end{bmatrix}$$

De esta forma a través de 6 parámetros $\theta_0, \theta_1, \theta_2, \theta_3, \theta_4$ y θ_5 se puede modelar cualquier transformación espacial básica ya sea de traslación, rotación y/o escalamiento.

Podemos expresar la ecuación A.15 en coordenadas homogéneas:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} \theta_0 & \theta_1 & \theta_2 \\ \theta_3 & \theta_4 & \theta_5 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (\text{A.16})$$

Con estas herramientas, ahora podríamos aplicarle una transformación afín a cualquier imagen I_1 para generar una nueva imagen I_2 .

Apéndice B

Optimización multivariable

En este capítulo se describe el método de Newton para el caso de una variable y para el caso de múltiples variables.

B.1. Optimización para una variable

El método de Newton es muy utilizado para aproximar las soluciones de la ecuación $f(x) = 0$. Se trata de uno de los métodos más populares para obtener las raíces de las ecuaciones no lineales. El método de Newton parte de un valor inicial x_0 para el valor de la raíz y se calcula una aproximación x_1 que generalmente es mejor que x_0 . De esta forma, y ahora partiendo de x_1 , se obtiene una mejor solución x_2 y así sucesivamente[3][9]. El método de Newton lleva a cabo la aproximación de la raíz con la siguiente fórmula:

$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)} \quad (\text{B.1})$$

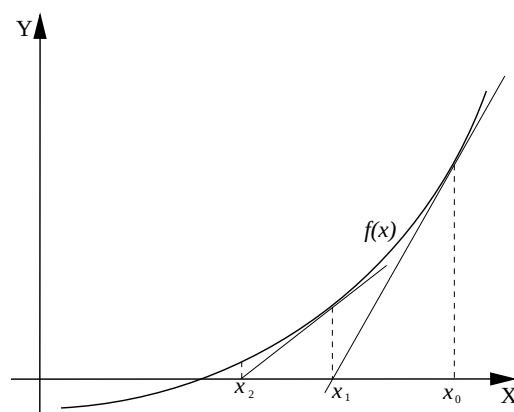


Figura B.1: Interpretación geométrica del Método de Newton

Si el algoritmo de Newton converge en una raíz, convergerá más pronto que cualquier otro método para obtener raíces. Pero quizá no converja siempre [9].

La interpretación geométrica del método de Newton (ver figura B.1), consiste en aproximar la función $f(x)$ con una línea recta en cada paso de la iteración. Se inicia con un valor inicial para la raíz x_0 , luego se calcula la siguiente aproximación de la raíz como x_1 al determinar la posición en que la línea de la tangente de x_0 cruza el eje de las x . En x_0 la pendiente de la curva es la primera derivada de la función $f(x)$ y se expresa como:

$$f'(x_0) = \frac{f(x_0)}{x_0 - x_1} \quad (\text{B.2})$$

Por lo tanto, la siguiente aproximación de la raíz está dada por:

$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)} \quad (\text{B.3})$$

que es la ecuación B.1 [9].

A continuación se presenta el caso en que el resultado de la función $f(x)$ depende de varias variables.

B.2. Optimización multivariable

Ahora nos referiremos a la función $E(\theta)$ como una función multivariable. En donde θ es un vector formado por n variables $(\theta_1, \theta_2, \theta_3, \dots, \theta_n)$. En este caso, la optimización busca encontrar los valores de θ que hagan:

$$E(\theta) = 0 \quad (\text{B.4})$$

Como parte importante para desarrollar la función $E(\theta)$ con el método de Newton, utilizaremos la serie de Taylor. De hecho, la ecuación B.1 puede ser derivada a partir de la serie de Taylor.

Aproximación por serie de Taylor

Si suponemos que $E(\theta)$ es una función continua y de clase 2 entonces podemos utilizar la Serie de Taylor[3] y aproximarla como :

$$E(\theta^{t+1}) \approx E(\theta^t) + [\nabla E(\theta^t)](\theta^{t+1} - \theta^t) + \frac{1}{2!}[\nabla^2 E(\theta^t)](\theta^{t+1} - \theta^t)^2 \quad (\text{B.5})$$

En donde la primera derivada de la función $\nabla E(\theta^t)$ es conocido como gradiente y la segunda derivada de la función $\nabla^2 E(\theta^t)$ se le conoce también como el Hessiano de la función.

Y si consideramos que el mínimo de una función diferenciable $E(\theta)$ de N variables reales $\theta = (\theta_1, \theta_2, \dots, \theta_N)$ puede encontrarse cuando su gradiente es igual a cero:

$$\frac{\partial E(\theta^{t+1})}{\partial \theta^{t+1}} = 0 \quad (\text{B.6})$$

Así es que al derivar los dos primeros elementos de la sumatoria de B.5, obtenemos:

$$\nabla E(\theta^t) + \nabla^2 E(\theta^t)(\theta^{t+1} - \theta^t) = 0 \quad (\text{B.7})$$

por último despejando θ^{t+1} de B.7 obtenemos:

$$\theta^{t+1} = \theta^t - [\nabla^2 E(\theta^t)]^{-1} \nabla E(\theta^t) \quad (\text{B.8})$$

A continuación describiremos, paso a paso, como aplicar el método de Newton a funciones multivariantes.

B.3. Solución con el método de Newton

El objetivo del método de Newton será encontrar los parámetros que permitan minimizar una función de la forma $E(\theta) = 0$.

Primeramente se enumeran en forma general los pasos del método de Newton y posteriormente son tratados por separado.

La solución para minimizar la ecuación B.8 implica los siguientes pasos:

1. Dar un valor inicial para θ^0
2. Calcular el gradiente $\nabla E(\theta^t)$
3. Calcular el Hessiano $\nabla^2 E(\theta^t)$
4. Calcular θ^{t+1} utilizando la ecuación B.8
5. Terminar si $|E(\theta^{t+1}) - E(\theta^t)| < \varepsilon$, si no regresar a 2

B.3.1. Valor inicial

Es necesario asignar valores iniciales a cada uno de los elementos de $\theta = (\theta_0, \theta_1, \dots, \theta_N)$. Los valores iniciales determinan la cantidad de iteraciones para para que la función converja o incluso, con valores iniciales muy alejados de la solución, determinar si ésta converge o no.

B.3.2. Cálculo del Gradiente

Para llevar a cabo el cálculo del gradiente utilizaremos como ejemplo la ecuación de error 3.1 de la página 44, la cual considera 6 parámetros θ . De esta forma, el cálculo del gradiente $\nabla E(\theta^t)$ aplicado sobre la ecuación 3.1 es el primer paso para obtener los nuevos valores de θ^{t+1} . El resultado final será un vector de 6 elementos ya que habrá que derivar con respecto $\theta_0, \dots, \theta_5$ respectivamente.

Partiendo de las ecuaciones 3.1 y 3.3 establecemos:

$$E(\theta) = \sum_{i=1}^{i=N} e_i(\theta) = \sum_{i=1}^{i=N} [I_1(\hat{r}_i(\theta)) - I_2(r_i)]^2 \quad (\text{B.9})$$

Ahora obtenemos el gradiente (∇) de B.9:

$$\nabla E(\theta) = \frac{\partial E(\theta)}{\partial \theta} = 2 \sum_{i=1}^{i=N} e_i(\theta) \frac{\partial I_1(\hat{r}_i(\theta))}{\partial(\theta)} \quad (\text{B.10})$$

Desarrollando B.10 para θ_0 :

$$\frac{\partial E(\theta)}{\partial \theta_0} = 2 \sum_{i=1}^{i=N} e_i(\theta) \frac{\partial I_1(\hat{r}_i(\theta))}{\partial(\theta_0)} \quad (\text{B.11})$$

Recordemos de la ecuación 3.2 que $\hat{r}$ está expresada como:

$$\hat{r} = \begin{bmatrix} \hat{x}(\theta) \\ \hat{y}(\theta) \end{bmatrix} = \begin{bmatrix} x & y & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & x & y & 1 \end{bmatrix} \begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \\ \theta_3 \\ \theta_4 \\ \theta_5 \end{bmatrix}$$

En donde hacemos notar que θ_0 solamente afectará a $\hat{x}(\theta)$

Utilizando la regla de la cadena[15], el término $\frac{\partial I_1(\hat{r}_i(\theta))}{\partial(\theta_0)}$ se puede expresar como:

$$\frac{\partial I_1(\hat{r}(\theta))}{\partial(\theta_0)} = \frac{\partial I_1(\hat{x}(\theta), \hat{y}(\theta))}{\partial(\theta_0)} = \frac{\partial I_1(\hat{x}(\theta), \hat{y}(\theta))}{\partial(\hat{x}(\theta))} \frac{\partial \hat{x}(\theta)}{\partial \theta_0} + \frac{\partial I_1(\hat{x}(\theta), \hat{y}(\theta))}{\partial(\hat{y}(\theta))} \frac{\partial \hat{y}(\theta)}{\partial \theta_0} = \frac{\partial I_1(\hat{x}(\theta), \hat{y}(\theta))}{\partial(\hat{x}(\theta))} (x) \quad (\text{B.12})$$

Resolviendo de igual forma para $\theta_1, \dots, \theta_5$ los resultados finalmente serían:

$$\begin{aligned}
\frac{\partial I_1(\hat{r}(\theta))}{\partial(\theta_0)} &= \frac{\partial I_1(\hat{x}(\theta), \hat{y}(\theta))}{\partial(\hat{x}(\theta))}(x) \\
\frac{\partial I_1(\hat{r}(\theta))}{\partial(\theta_1)} &= \frac{\partial I_1(\hat{x}(\theta), \hat{y}(\theta))}{\partial(\hat{x}(\theta))}(y) \\
\frac{\partial I_1(\hat{r}(\theta))}{\partial(\theta_2)} &= \frac{\partial I_1(\hat{x}(\theta), \hat{y}(\theta))}{\partial(\hat{x}(\theta))} \\
\frac{\partial I_1(\hat{r}(\theta))}{\partial(\theta_3)} &= \frac{\partial I_1(\hat{x}(\theta), \hat{y}(\theta))}{\partial(\hat{y}(\theta))}(x) \\
\frac{\partial I_1(\hat{r}(\theta))}{\partial(\theta_4)} &= \frac{\partial I_1(\hat{x}(\theta), \hat{y}(\theta))}{\partial(\hat{y}(\theta))}(y) \\
\frac{\partial I_1(\hat{r}(\theta))}{\partial(\theta_5)} &= \frac{\partial I_1(\hat{x}(\theta), \hat{y}(\theta))}{\partial(\hat{y}(\theta))}
\end{aligned} \tag{B.13}$$

Los 6 resultados obtenidos integran uno de los N vectores que se generarán a lo largo de un solo cálculo de θ . Donde N es el número de píxeles de la imagen. Así pues podemos decir que en cada proceso para calcular un nuevo θ se generará una matriz completa de $N \times 6$ la cual se conoce como el Jacobiano de la función de error:

$$J(\theta) = \frac{\partial E(\theta)}{\partial(\theta)} = \begin{bmatrix} \frac{\partial E_1(\theta)}{\partial \theta_0} & \frac{\partial E_1(\theta)}{\partial \theta_1} & \frac{\partial E_1(\theta)}{\partial \theta_2} & \dots & \frac{\partial E_1(\theta)}{\partial \theta_5} \\ \frac{\partial E_2(\theta)}{\partial \theta_0} & \frac{\partial E_2(\theta)}{\partial \theta_1} & \frac{\partial E_2(\theta)}{\partial \theta_2} & \dots & \frac{\partial E_2(\theta)}{\partial \theta_5} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \frac{\partial E_N(\theta)}{\partial \theta_0} & \frac{\partial E_N(\theta)}{\partial \theta_1} & \frac{\partial E_N(\theta)}{\partial \theta_2} & \dots & \frac{\partial E_N(\theta)}{\partial \theta_5} \end{bmatrix} \tag{B.14}$$

B.3.3. Cálculo del Hessiano

El Hessiano es una segunda derivada de la función de error. En tal caso cada uno de los resultados obtenidos en B.13 tendrían que ser derivados nuevamente. Sin embargo, podemos optar por método de Gauss-Newton [4], aprovechando que ya se ha calculado el Jacobiano de la función de error, por lo que podemos utilizar la siguiente expresión para aproximar el Hessiano [3]:

$$\nabla^2 E(\theta) = 2J(\theta)^T J(\theta) \tag{B.15}$$

B.3.4. Condición de terminación

Al obtener el nuevo valor θ^{t+1} se puede evaluar la expresión $|E(\theta^{t+1}) - E(\theta^t)| < \varepsilon$, en donde ε representa un umbral de error previamente definido. En caso de no satisfacer

dicha condición se comenzará nuevamente en el paso 2, haciendo $\theta^t = \theta^{t+1}$.

Apéndice C

Filtros para imágenes

Para la detección de esquinas en una imagen, se utilizó un método extraído de [17], el cual es llevado a cabo a través de varios procesos que a continuación describimos y que incluye aplicar filtros de derivadas y filtros gaussianos a la imagen [3].

C.1. Filtro de derivadas

Para la detección de esquinas en las imágenes, se utilizaron dos filtros de *derivadas gaussianas*. La derivada de una imagen consiste básicamente en calcular las diferencias entre píxeles vecinos, ya sea en la dirección x o en la dirección y [3]. El efecto de derivar una imagen es que se resaltan los bordes en la misma. De esta forma, al derivar la imagen, en la dirección x estamos resaltando sus características horizontales. En forma contraria, si la dirección de la derivada es y , son los bordes verticales los que destacarán ¹. En la figura C.1 se muestra un ejemplo de derivadas en ambas direcciones (en este caso se le aplicó la derivada a la imagen de la figura 3.7(b)).

C.2. Filtro gaussiano

Por otra parte, aplicar un filtro gaussiano crea un efecto de suavizar la imagen, de esta forma estamos ensanchando los bordes de la misma. En la figura C.2 mostramos dos imágenes en la que a una de ellas se le ha aplicado un filtro gaussiano.

¹Considerando que para todas nuestras implementaciones utilizamos un sistema de referencia derecho girado, por lo que el eje x es el eje vertical y el eje y es el eje horizontal (ver apéndice A de [9]).

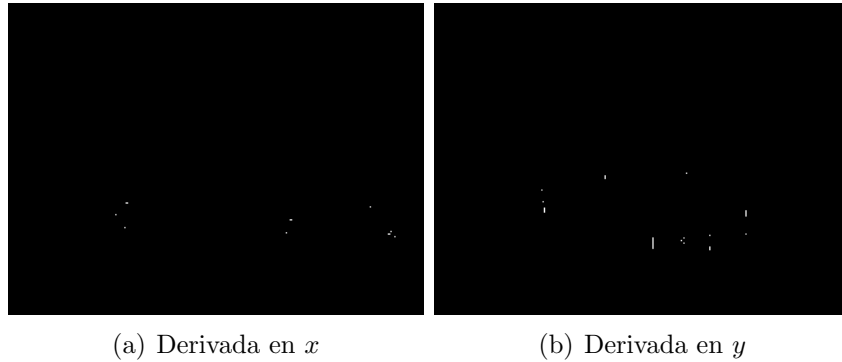


Figura C.1: La imagen es derivada primero en la dirección x (a), haciendo destacar sus bordes horizontales y por separado es derivada en la dirección y (b), destacándose ahora los bordes verticales.

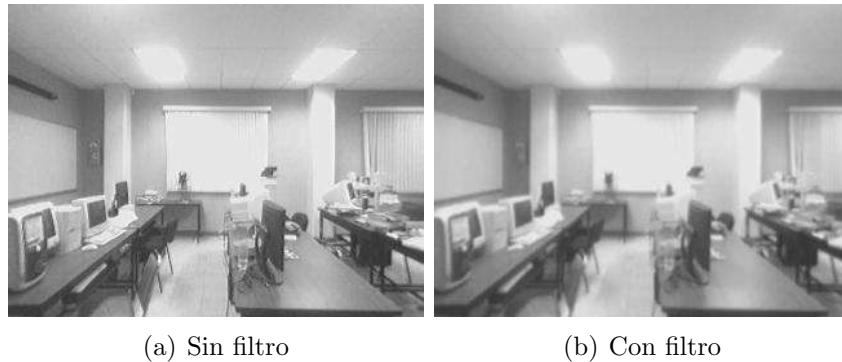


Figura C.2: Diferencia entre una imagen con filtro gaussiano (b) y otra que no lo tiene (a).

C.3. Implementación del filtro Gaussiano y filtro de Derivadas

La implementación de un filtro gaussiano se puede hacer a través de una *convolución* en 2 dimensiones.

Una convolución, es una transformación de las propiedades de una imagen por medio de un “*kernel*” previamente definido. Convolucionar una imagen se refiere a transformar el contenido de la misma, pixel por pixel, pero considerando las características de sus vecinos.

Un kernel, es el elemento que determina el tipo de convolución. El kernel puede ser unidimensional (vector) o de dos dimensiones (matriz). Para el caso muy particular de

convolucionar imágenes, podemos establecer que una imagen es un arreglo de valores enteros, de tamaño $m \times n$ (ver figura C.3), y en donde el kernel es también una matriz pero de $r \times r$ elementos.

La convolución se aplicará sobre todos y cada uno de los pixeles, por ejemplo, al multiplicar el kernel por el pixel p_{13} de la imagen, el nuevo valor p'_{13} sería:

$$p'_{13} = p_2k_9 + p_3k_8 + p_4k_7 + p_{12}k_6 + p_{13}k_5 + p_{14}k_4 + p_{22}k_3 + p_{23}k_2 + p_{24}k_1$$

De esta forma, la convolución puede ser definida como[23]:

$$p'_{(m,n)} = \sum_{i=h}^{i=h} \sum_{j=-h}^{j=h} p_{(n-i,m-j)} k_{(i,j)} \quad (C.1)$$

En donde h es la parte entera de $\frac{r}{2}$ y se aplica a todos los $m \times n$ pixeles de la imagen en donde m son las filas y n las columnas [3].

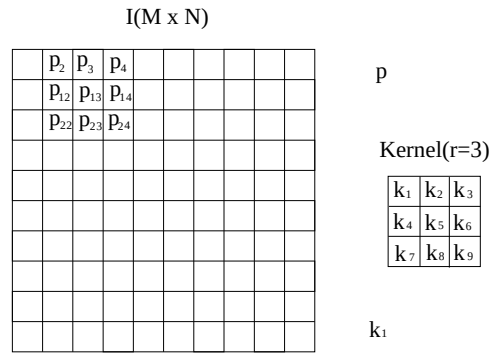


Figura C.3: La imagen I y el kernel son representados como matrices, en donde el tamaño del kernel determinará que tanta influencia de los vecinos se quiere tener en la convolución.

Aunque existen diferentes tipos de kernels, nosotros estamos interesados únicamente en dos de ellos: el kernel gaussiano y el kernel de derivadas. Durante la aplicación del método de corrección por puntos de correspondencia ha surgido la necesidad de calcular primeras y segundas derivadas a imágenes previamente suavizadas con filtros gaussianos. La convolución es una forma de llevar a cabo dichos cálculos simplemente definiendo el kernel correcto.

Algunos kernels útiles para nuestro trabajo son:

0	-1	0	0	0	0	0	0	0	0
0	1	0	-1	1	0	0	-2	0	1
0	0	0	0	0	0	0	1	0	0

$$\partial I(x, y) / \partial x \quad \partial I(x, y) / \partial y \quad \partial^2 I(x, y) / \partial x^2 \quad \partial^2 I(x, y) / \partial y^2$$

Una forma de construir el kernel gaussiano $g_{(x,y)}$ es a través de la siguiente expresión:

$$g_{(x,y)} = \left[\frac{1}{2\pi\sigma_x} e^{-\frac{(x-\mu_x)^2}{2\sigma_x^2}} \right] \left[\frac{1}{2\pi\sigma_y} e^{-\frac{(y-\mu_y)^2}{2\sigma_y^2}} \right] \quad (\text{C.2})$$

Si consideramos que $\mu_x = \mu_y = 0$ y que $\sigma_x = \sigma_y = \sigma$ entonces la ecuación C.2 queda de la siguiente forma:

$$g_{(x,y)} = \left[\frac{1}{2\pi\sigma} e^{-\frac{x^2}{2\sigma^2}} \right] \left[\frac{1}{2\pi\sigma} e^{-\frac{y^2}{2\sigma^2}} \right] \quad (\text{C.3})$$

En donde σ determina el tamaño del kernel: $r = 2\sigma + 1$.

El efecto de aplicar previamente un filtro gaussiano se muestra en la figura C.4 destacándose el grosor de los bordes detectados. El qué tan suavizada quede una imagen está determinado a través del valor de σ .

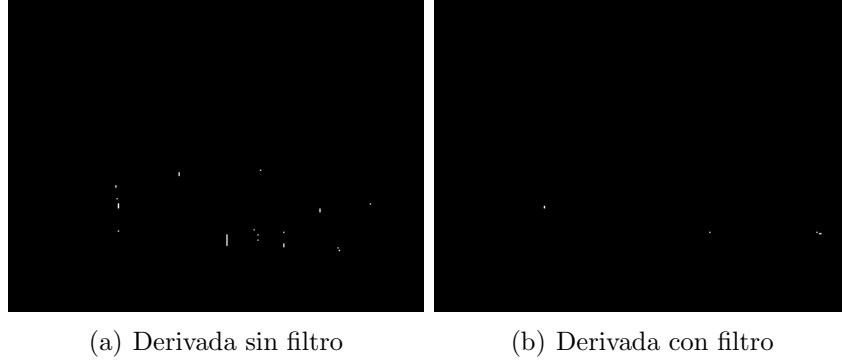


Figura C.4: Aplicar o no el filtro gaussiano por una parte parece ensanchar los bordes detectados y al mismo tiempo se difuminan otros al grado de desaparecer. La derivada de estas imágenes se ha hecho en ambas direcciones.

Por último, utilizando el filtro gaussiano con $\sigma = 2$ y aplicando derivadas en ambas direcciones podemos encontrar las esquinas de las imágenes utilizadas para la corrección de la distorsión (ver figura 3.4(c) de la página 55). La figura C.5 nos muestra parte de dicho proceso. En primer lugar (a) se muestra una imagen a la que se le aplicó una convolución utilizando kernels gaussianos y de derivadas en el sentido de las x . La figura del centro (b) se produce al aplicar la misma convolución en ambos sentidos. El resultado es un conjunto de cúmulos sobre cada una de las esquinas que se detectaron en la imagen. Finalmente se calcula el centro de masa de cada uno de los cúmulos en ambas imágenes (ver ecuación 4.9 de la página 79) y el resultado es un conjunto de puntos (c) con precisión subpixel.

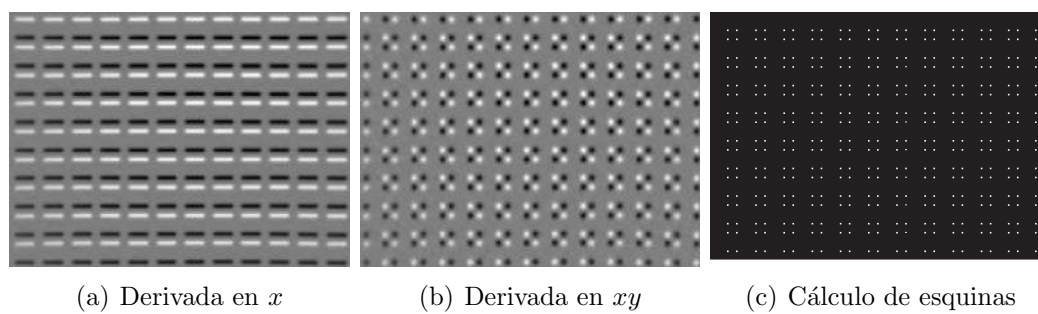


Figura C.5: Para obtener las esquinas en el patrón de calibración a nivel subpixel se hace uso de las derivadas en ambas direcciones (a) y (b) y posteriormente se calcula el centro de masa de cada cúmulo (c).

Índice alfabético

- Ambientes interiores, 5
Autocalibración, 8, 11, 14, 47

Binarización de una imagen, 18, 21, 25, 26, 33

Cálculo del gradiente, 118
Cálculo del Hessiano, 119
Cámara ideal, 3, 9, 10, 16
Cámaras de vídeo, 4
Calibración estéreo, 1, 2
Calibración externa, 15
Calibración interna, 13
calibración, el problema de la, 1, 2
Centro de distorsión, 11, 14, 16, 31, 32, 36, 50, 67
Centro de masa, 79
Centro de proyección, 46
Centro geométrico, 11, 31, 36
Coeficiente de distorsión, 7, 14, 17, 32, 50, 54, 102
Composición de matrices, 112
Concatenación de matrices, 112
Coordenadas homogéneas, 110
Corrección de la distorsión, 3, 7, 8, 11, 16, 18, 32, 35, 47, 124
Correspondencia colineal, 7
Correspondencia de imágenes, 61, 62, 102

Dispositivo fotosensible, 9, 32
Distancia focal, 7, 9, 12, 14, 54, 76, 78, 80, 87, 102
Distorsión radial, 7, 14, 17, 43, 54, 65
Dominio de Hough, 24–29, 32–34

Dominio de la imagen, 22–24, 27–29, 34

Ecuación dos puntos, 21
Ecuación pendiente ordenada al origen, 22
Ecuación punto pendiente, 22
Escalamiento, 7

Falsas lecturas, 6

Geometría epipolar, 8, 64
Geometría estéreo, 2, 8

Imagen en la cámara, 12
Imagen sintética, 10
Imagen virtual, 12

Jacobiano, 48, 119

Método de Cardano, 38
Método de Newton, 7, 115
Mapa de disparidad, 5, 8, 60, 69–72, 74–76, 81, 83–85, 88, 91–93, 95, 101, 102
Marco de referencia global, mundial, 15
Matriz de rotación, 7, 15, 107–109, 111, 112
Modelo de distorsión, 17, 47
Modelo de Pinhole, 9, 10, 87, 88, 93
Modelo de pinhole, 3

Origen de los huecos, 35

Parámetros ópticos, 12
Parámetros de calibración, 11
Parámetros de corrección, 11
Parámetros externos, extrínsecos, 12, 14, 15
Parámetros fotométricos, 12
Parámetros geométricos, 12–15

Parámetros internos, intrínsecos, 12–14, 18, 32, 35, 77, 83, 93
Patrón de calibración, 7, 18, 31, 34, 43, 47, 52, 54, 65–67, 125
Patrón sintético, 7
Problema de correspondencia, 3, 60
Problema de la reconstrucción, 1, 4, 76, 101
Punto focal, 9, 12
Puntos de control, Marcas de control, 14

Reconstrucción, 1, 8
Registro de imágenes, 6–8, 14, 43, 65, 66
Representación de una recta, 21
Representación paramétrica de una recta, 22, 24
Representación polar de una recta, 24, 27
Robot reactivo, 2
Robots móviles, 2
Rotación, 7

Shirai, 7, 102

Telémetro Láser, 3
Telémetro láser, 5
Transformación afín, 46, 112, 113
Transformación compuesta, 111
Transformaciones proyectivas, 7
Transformada de Hough, 18, 19, 34
Traslación, 7
Trasformaciones espaciales, 7

Vector de translación, 15
Vector de traslación, 107, 108, 111, 112