



CENTRO DE INVESTIGACIÓN Y DE ESTUDIOS
AVANZADOS
DEL INSTITUTO POLITÉCNICO NACIONAL

Unidad Zacatenco

Departamento de Computación

**Reconstrucción visual 3D basada en evolución
diferencial**

Tesis que presenta

Heriberto Cruz Hernández

para obtener el Grado de

Maestro en Ciencias

en Computación

Director de la Tesis

Dr. Luis Gerardo de La Fraga

México, D.F.

Octubre 2012

Resumen

El proceso de reconstrucción 3D consiste de obtener las coordenadas de puntos en el espacio a partir de sus coordenadas proyectadas en un conjunto de imágenes.

En visión por computadora, este problema es atacado mediante la obtención incremental de información tridimensional acompañada de ajustes frecuentes por métodos no lineales. Dichos métodos son propensos a quedar atrapados en mínimos locales, además de que son sensibles al ruido, el cual es frecuente en problemas de la vida real. En este trabajo de tesis se propone un procedimiento para resolver el problema de reconstrucción, el cual utiliza evolución diferencial para obtener mejores soluciones iniciales que los métodos tradicionales.

El procedimiento necesita como entrada un conjunto de imágenes de una escena. Se comienza por calibrar la cámara utilizando las primeras tres imágenes del conjunto mediante evolución diferencial, con esto se obtienen las características de la cámara con la que fueron generadas las imágenes de entrada, así como una reconstrucción parcial del entorno que permite procesar las imágenes faltantes. Para cada imagen del conjunto se estima la posición y orientación con que fue generada mediante evolución diferencial, lo cual permite triangular nuevos puntos que van siendo agregados, permitiendo así una reconstrucción incremental. La calidad de la reconstrucción es monitoreada con cada imagen que se procesa, de tal forma que cuando se sobrepasa un umbral de incertidumbre, la reconstrucción es sometida a un ajuste por mínimos cuadrados no lineales, lo que permite reducir la incertidumbre y por lo tanto facilitar el procesamiento de más imágenes.

Mediante esta solución se busca reducir la frecuencia con que se llevan a cabo los ajustes por métodos no lineales, evitar los mínimos locales y hacer el proceso de reconstrucción en dos y tres dimensiones menos sensible al ruido.

Abstract

The 3D reconstruction problem, asks to get the 3D coordinates of a set of points in a scene from the point's projections in a set of images.

Computer vision deals with this problem through incremental computing of 3D information from the set of images along with frequent non-linear adjustments. Frequently these kind of adjustments are susceptible to find local minima and are sensible to noise too. In this thesis work, a new procedure is proposed to solve the reconstruction problem which is based on differential evolution in order to get better initial solutions than traditional methods.

Our proposal receives as input a sequence of images taken from a scene. The process starts with the camera calibration, this stage is done using the three first input sequence images and differential evolution. This calibration stage allows to get camera intrinsic parameters along with an initial reconstruction which allows us to process new images. For the other input images we estimate position and orientation as the image was originally generated, this step allows us to triangulate new points getting so an incremental reconstruction. Reconstruction quality is always checked every time a new image is processed in order to adjust the reconstruction when an uncertainty threshold is overpassed. These adjustments are made by a non-linear least squares method what reduces both the errors and the uncertainty.

With our proposal we try to reduce the number of non-linear adjustments, to avoid local minima and to make 2D and 3D reconstruction problem more robust against noise.

Agradecimientos

Quiero agradecer profundamente a mis papás que siempre están allí para apoyarme en todas las decisiones de mi vida y por todo el amor que siempre me dan. Gracias, nunca encontraré la forma de pagarles todo lo que hacen por mí.

Agradezco también a mi hermano Edgar, mi tía Rosa Icela, a mi primas Ginna y Diana porque siempre están alegrándome la vida.

A mi novia Cinthya por todo lo que hace por mí . Gracias por siempre estar para mí en los todos los momentos sin importar si son buenos o malos y por toda la felicidad que me brindas.

Agradezco de manera muy especial mi asesor de tesis el Dr. Luis Gerardo de la Fraga por haberme enseñado tantas cosas desde el primer día que inició mi aventura en el CINVESTAV, así como por haber sido un asesor con mano firme pero paciente en todo momento. De verdad le agradezco de todo corazón.

Agradezco a todos mis profesores por siempre compartir con entusiasmo un poquito de su conocimiento científico como de la vida además de ser para mí un excelente ejemplo a seguir.

Especialmente agradezco al Dr. Carlos Artemio Coello Coello y al Dr. José Gabriel Ramírez Torres por que además de haber sido excelentes profesores invirtieron parte de su tiempo en la revisión de mi tesis para hacer valiosos comentarios y correcciones que me sirven de mucho.

Agradezco al CINVESTAV por brindarme la oportunidad de estudiar una maestría en tan importante institución y permitirme conocer tantas cosas increíbles y personas geniales.

También agradezco al CONACYT por la beca que se me otorgó durante toda mi estancia en el CINVESTAV y al proyecto 168357 de Ciencia básica del CONACYT por todos los apoyos recibidos.

Índice general

Resumen	III
Abstract	v
Agradecimientos	VII
Índice de figuras	x
Índice de tablas	xII
1. Introducción	1
1.1. Motivación	1
1.2. Planteamiento del problema	3
1.3. Objetivos generales y particulares	5
1.4. Estructura del documento	5
2. Fundamentos Teóricos	7
2.1. Transformaciones	7
2.1.1. Traslación	8
2.1.2. Rotación	9
2.1.3. Rotación y traslación	10
2.1.4. Escalamiento	10
2.2. Métodos de estimación	10
2.2.1. Optimización lineal por mínimos cuadrados	11
2.2.2. Optimización no lineal por mínimos cuadrados	12
2.2.3. Optimización no lineal con algoritmos evolutivos	14
2.3. Conceptos de Visión por Computadora	17
2.3.1. Modelo de la cámara oscura	17
2.3.2. Calibración de la cámara	19
2.3.3. Triangulación	22
2.3.4. Modelo de distorsión	24
2.4. Reconstrucción	25
2.5. Calidad de la reconstrucción	27
2.6. Generación de la imagen que vería una cámara móvil	30

3. Estado del arte	33
3.1. Calibración	33
3.2. Localización	39
3.3. Construcción y actualización del mapa	41
3.3.1. Errores y ajustes	42
4. Propuesta de solución	45
4.1. Descripción general	45
4.2. Auto-calibración	47
4.3. Localización	47
4.4. Ajuste de la escala	49
4.5. Eliminación de puntos atípicos	50
4.6. Cálculo de la incertidumbre	51
4.7. Consideraciones sobre la cámara móvil	53
5. Experimentos y resultados	55
5.1. Configuraciones mínimas	55
5.2. Reconstrucción de entornos planos con datos reales	58
5.3. Reconstrucción de entornos tridimensionales	59
5.4. Número de evaluaciones y tiempos de ejecución	65
6. Conclusiones y trabajo a futuro	69
A. Cálculo de derivadas parciales para el cálculo de la incertidumbre	73
Bibliografía	77

Índice de figuras

1.1. Luz reflejada y captada por la vista.	1
1.2. Requerimientos comunes de visión por computadora.	3
1.3. Cámara móvil en un entorno plano.	4
1.4. Cámara móvil en un entorno tridimensional.	5
2.1. Punto en el espacio.	8
2.2. Objeto formado por puntos.	8
2.3. Traslación de un punto en el espacio.	9
2.4. Ejemplo de rotación de un punto sobre el eje y	9
2.5. Ejemplo de escalamiento.	11
2.6. Etapas de la evolución diferencial.	15
2.7. Generación de imágenes y recuperación de información 3D.	17
2.8. Modelo clásico de la cámara oscura.	17
2.9. Modelo de la cámara oscura.	18
2.10. Ejemplos de fotografías usadas para la calibración de la cámara.	20
2.11. Triangulación de una correspondencia.	23
2.12. Efectos de la distorsión radial.	25
2.13. Corrección de la distorsión de una imagen.	26
2.14. Proceso de reconstrucción.	26
2.15. Incremento de incertidumbre durante el proceso de reconstrucción.	27
2.16. Ilustración del “cerrado de ciclo” durante el proceso de reconstrucción.	27
2.17. Error de reproyección.	28
2.18. Residuales de las observaciones.	29
2.19. Error cuadrático medio.	29
3.1. Ejemplo de mapa reconstruido con acumulación de errores.	34
3.2. Patrones planos comúnmente utilizados para calibración.	34
3.3. Puntos sobre un plano con $z = 0$	35
3.4. Agregando una nueva vista.	38
3.5. Localización de una cámara móvil.	40
3.6. Ciclo de reconstrucción.	42
4.1. Procedimiento general.	46
4.2. Proceso de localización de una imagen.	48

4.3. Ejemplo de soluciones.	49
4.4. Ajuste de la escala.	49
4.5. Detección de punto atípico.	50
4.6. Incertidumbre y la escala.	53
4.7. Escena con cámara estática.	54
4.8. Escena con cámara móvil.	54
5.1. Entornos planos utilizados para encontrar configuraciones mínimas.	56
5.2. Tipos de movimiento.	56
5.3. Porcentajes de éxito obtenidos con distintas configuraciones para auto-calibración 2D.	56
5.4. Porcentaje de éxito de distintas configuraciones para localización 2D.	57
5.5. Porcentaje de éxito de configuraciones de movimiento para auto-calibración 2D.	58
5.6. Imágenes del conjunto de datos utilizando por Zhang.	59
5.7. Porcentajes de éxito para las soluciones iniciales de reconstrucción para los datos de Zhang.	60
5.8. Imágenes usadas para calibrar la cámara con el método de Zhang.	61
5.9. Porcentajes de éxito para el número de puntos en la auto-calibración 3D.	63
5.10. Porcentajes de éxito para el número de puntos en la localización 3D.	64
5.11. Imágenes ejemplo del conjunto de datos del dinosaurio de Oxford (fotografía y puntos de interés).	65
5.12. Vistas del modelo reconstruido.	65

Índice de tablas

3.1. Trabajos y métodos usados para calibración.	39
3.2. Soluciones al problema de localización.	41
3.3. Trabajos y métodos usados para localización.	41
4.1. Límites y parámetros para ejecutar la evolución diferencial utilizados para realizar la auto-calibración.	48
5.1. Grupos de imágenes disponibles para reconstrucción de rejilla de cuadros.	60
5.2. Solución final de los parámetros de la cámara para la rejilla de cuadros.	60
5.3. Parámetros intrínsecos de la cámara usada para la generación de las imágenes de la figura de cuboides.	61
5.4. Parametros de la cámara y error RMS para el modelo construido.	62
5.5. Ángulos de rotación y vectores de traslación para las imágenes del modelo construido.	62
5.6. Fotografías y su reproducción.	62
5.7. Vistas del modelo reconstruido.	63
5.8. Parametros de la cámara y error RMS para el modelo construido.	64
5.9. Características de la computadora.	65
5.10. Tiempos y número de evaluaciones para las etapas de auto-calibración 2D y localización 2D.	66
5.11. Tiempos y número de evaluaciones para las etapas para el caso 3D de auto-calibración y localización.	67

Capítulo 1

Introducción

1.1. Motivación

En la naturaleza los seres vivos hemos sido dotados de la capacidad de interactuar con nuestro entorno. Esta capacidad nos permite satisfacer necesidades propias de los seres vivos: alimento, protección, aprendizaje, etc. Sin embargo para que el proceso de interacción se lleve a cabo es imprescindible la presencia de dos subprocesos, por un lado la obtención de información del exterior (entorno) o capacidad sensorial y por otro lado la manipulación del mismo. Ambas actividades se necesitan mutuamente para poder funcionar eficientemente.

Los seres humanos obtenemos información de nuestro entorno a través de los cinco sentidos: vista, olfato, oído, tacto y gusto. Sin embargo de los cinco sentidos, probablemente del que más información obtenemos de manera consciente es la vista [1, 2]. Gracias a este sentido podemos realizar desde actividades básicas diarias que implican la búsqueda y reconocimiento de lugares, la identificación de personas, la manipulación de objetos, hasta actividades intelectuales de entretenimiento como mirar una puesta de sol, leer un libro o una revista, etc.

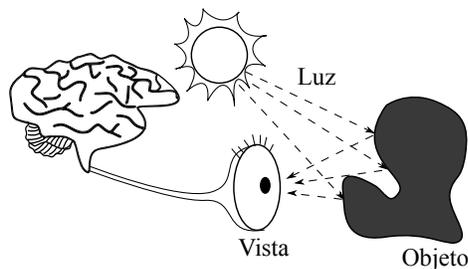


Figura 1.1: Luz reflejada y captada por la vista.

La vista es el sentido que nos permite captar la luz que es reflejada en los objetos de tal forma que ésta pueda ser procesada por el cerebro (véase figura 1.1). Los seres humanos somos capaces no sólo de conocer la forma de los cuerpos, sino que también somos capaces de detectar aspectos como: las texturas, los colores, el movimiento,

de tal forma que es imposible cuantificar la cantidad de información que se puede obtener a partir de la vista.

Dada la importancia de la vista para nuestra vida cotidiana y siendo conscientes de las ventajas y las potenciales actividades que podemos llevar a cabo con la información obtenida de este sentido, nos resulta natural pensar en imitar el funcionamiento de la visión natural, para generar mecanismos artificiales que permitan agregar en medida de lo posible estas capacidades a objetos de uso cotidiano que hagan más fácil y productiva nuestra vida.

En los inicios de la visión por computadora situados alrededor de 1970 [3], la visión fue considerada sólo un componente más para poder dotar a las máquinas de inteligencia, es decir, comportamiento similar al humano. Se pensó en su momento que la visión era una tarea sencilla comparándola con la difícil tarea de dotar a las máquinas de razonamiento. Hoy en día estamos conscientes de la dificultad que representa el dotar a una computadora con una cámara y lograr que comprenda lo que ve. No sólo se desea poder aplicar filtros que permitan y mejoren la visualización de la información presente en una imagen, sino que además se desea recuperar la información tridimensional de la escena dónde se tomó la imagen. En robótica existe el problema de *localización y reconstrucción simultánea* (LRS) también conocido como SLAM [4] (*simultaneous localization and mapping*) por sus siglas en inglés. El problema LRS tiene como objetivo obtener la estructura un entorno desconocido al mismo tiempo que se debe de saber en donde se encuentra un robot en el mismo mapa que se va obteniendo. El problema de LRS puede abordarse utilizando distintos instrumentos entre los que se encuentran: rayos láser [5], visión monocular [6] y visión estereoscópica [7]. Cuando el instrumento utilizado es una cámara, el problema es conocido como *localización y reconstrucción simultánea visual* (LRSV) o VSLAM (*visual simultaneous localization and mapping*) por sus siglas en inglés. El problema de reconstrucción está muy relacionado con el problema LRSV ya que ambos requieren de la recuperación de la información tridimensional de una escena, es por ello que en este trabajo de tesis ambos problemas son mencionados continuamente.

El deseo del entendimiento 3D por parte de las máquinas fue el impulsor de una serie de trabajos, enfocados en obtener un mejor tratamiento en la detección de bordes [8], esquinas [9], líneas [8] y correspondencias [10] en imágenes estereoscópicas (véase figura 1.2). Gracias a estos avances hoy en día es posible la integración de módulos capaces de recuperar parte de la información en una escena, permitiendo así acercarnos pero sin lograr aún la comprensión total de la información presente en una escena tridimensional.

Con los avances que se tienen en el campo de visión se han logrado aplicaciones en campos como la industria, la medicina y la robótica. En la industria, la aplicación se ha enfocado en ubicar una cámara en una línea de producción, de tal forma que se realiza una inspección visual a los productos en búsqueda de partes faltantes, defectos de fabricación, etc. Este tipo de actividades anteriormente podían llevarse a cabo sólo por seres humanos. Debido a lo repetitivo de la tarea, estas actividades eran propensas a fallas. En medicina las aplicaciones pueden apreciarse en actividades como cirugía

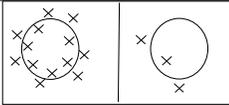
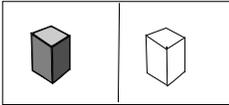
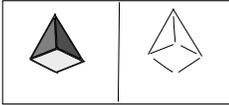
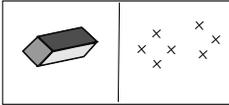
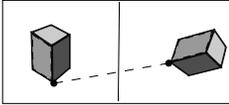
Procedimiento	Ilustración
Reducción de ruido	
Detección de bordes	
Extracción de líneas	
Detección de esquinas	
Detección de correspondencias	

Figura 1.2: Requerimientos comunes de visión por computadora.

asistida por computadora y recuperación de imágenes con información tridimensional del interior del cuerpo. En este campo la decisión de lo que se va a hacer sigue siendo una tarea humana. Sin embargo, debido a lo delicado de las tareas, es conveniente utilizar sistemas computarizados que alerten en caso de detectar un posible problema en el caso de la inspección visual, o bien que permitan realizar movimientos con más precisión en una cirugía. En robótica se desea dotar a las máquinas de la capacidad de transitar por sí solas en espacios de estructura desconocida, esta actividad requiere de la comprensión de la estructura tridimensional de la escena para evitar que los robots colisionen o pasen innecesariamente por un mismo lugar dos veces. Este tipo de aplicaciones pueden verse en los robots móviles que son enviados a otros planetas para realizar tareas de exploración. En este escenario el control a distancia por parte de un ser humano no es una opción, pues el tiempo de envío y recepción de información a lugares tan lejanos aún a la velocidad de la luz es de minutos, por esta razón es necesaria la exploración casi autónoma por parte del robot.

Con lo ya mencionado es necesario enfatizar que la visión por computadora sigue siendo un campo activo de investigación, en el que se requiere mejorar los resultados y procedimientos ya obtenidos para acercarnos más al sueño del que inicialmente surgió.

1.2. Planteamiento del problema

El tema principal del trabajo de tesis es la reconstrucción del modelo tridimensional de un entorno a partir de una secuencia de imágenes tomadas por una misma cámara. En la literatura científica [11, 12, 13, 14, 15] se puede encontrar que esta tarea tiene asociados algunos problemas, como son:

- **Acumulación de errores.** La acumulación de errores en la reconstrucción incremental dificulta la reconstrucción de entornos amplios.
- **Óptimos locales y costo de ajustes.** Cuando se hacen ajustes con técnicas de programación matemática como Levenberg-Marquardt, si la solución obtenida inicialmente no es lo suficientemente buena, el algoritmo converge a un óptimo local, lo que significa que el mapa obtenido no es el más aproximado. Para incrementar la calidad del mapa suele utilizarse un gran número de ajustes mediante el método Levenberg-Marquardt, lo que se traduce en costos computacionales altos. Algunos trabajos como [16] y [17] sacrifican la calidad del mapa evitando el uso de este tipo de técnicas.
- **Mapa poco preciso.** El hecho de obtener un mapa no tan preciso entorpece la fase de localización en trabajos que requieren el mapa para lograr localización en tiempo real.

El problema será abordado para el caso de un entorno plano, es decir asumiendo que el terreno por el cual transita una cámara móvil es plano (véase figura 1.3), así como para entornos con puntos en un sistema coordenado de tres dimensiones (véase figura 1.4).

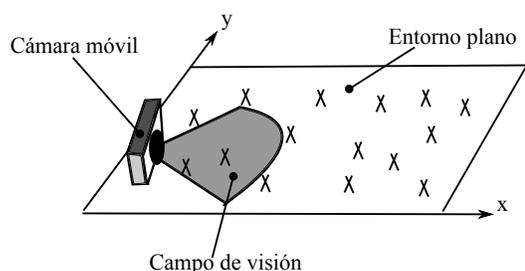


Figura 1.3: Cámara móvil en un entorno plano.

Con este trabajo de tesis se pretende reducir el número de veces que se utiliza el algoritmo Levenberg-Marquardt proporcionando una mejor solución inicial. Con esto se busca disminuir los efectos causados por los problemas comunes como: acumulación de errores, óptimos locales y mapas poco precisos que están presentes en algunos trabajos como [11, 12, 13, 14, 15].

Hipótesis

Es posible realizar la reconstrucción de un entorno utilizando visión monocular y un procedimiento basado en evolución diferencial que proporcione una buena solución inicial, de tal manera que sea posible reducir la cantidad de ajustes por Levenberg-Marquardt necesarios, al mismo tiempo de que se obtienen reconstrucciones más precisas.

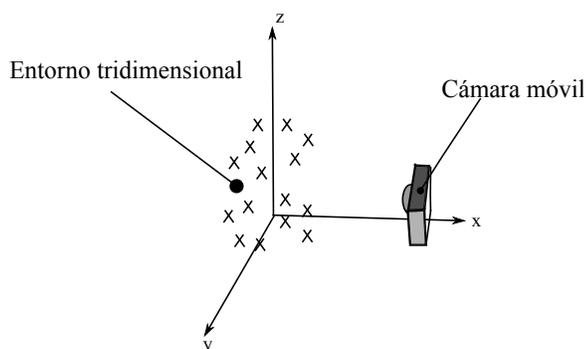


Figura 1.4: Cámara móvil en un entorno tridimensional.

1.3. Objetivos generales y particulares

Desarrollar un procedimiento que resuelva el problema de reconstrucción en entornos planos así como tridimensionales utilizando evolución diferencial, de forma tal que se logre una mejor calidad en la reconstrucción y que se requiera de menos ajustes.

Particulares

1. Desarrollar una metodología para agregar una nueva vista a partir de m puntos, utilizando un procedimiento basado en evolución diferencial [18].
2. Establecer un criterio para decidir cuándo aplicar un ajuste total tanto de la reconstrucción como de las poses de las cámaras, mediante el método Levenberg-Marquardt.
3. Extender la metodología para planos a puntos con tres dimensiones.
4. Establecer un criterio para decidir cuándo aplicar un ajuste total, tanto de la reconstrucción como de las poses de las cámaras en el caso 3D.
5. Probar los procedimientos propuestos para casos simulados así como reales.

1.4. Estructura del documento

El tema principal de este trabajo es la reconstrucción de escenas planas y tridimensionales. El capítulo 2 contiene el conocimiento teórico sobre el que esta tesis se basa; en donde se explican conceptos como el modelo de la cámara, la calibración de la misma, triangulación, el proceso de reconstrucción, conceptos de optimización así como métricas de calidad en las reconstrucciones. En el capítulo 3 se presentan otros trabajos con el fin de mostrar los enfoques con que se ha abordado este problema. El capítulo 4 presenta la propuesta de solución para el caso de entornos planos y tridimensionales, que se refiere a aspectos que van desde el proceso de autocalibración, hasta el procedimiento para la obtención de la reconstrucción y eliminación de

puntos atípicos y su aplicación para localizar un robot móvil. El capítulo 5 muestra los experimentos y resultados obtenidos tanto en simulaciones como en casos reales. El capítulo 6 presenta las conclusiones, así como las mejoras y extensiones que podrían aplicarse a este trabajo de tesis con base en los resultados obtenidos. Por último, en el apéndice [A](#) se presenta a detalle la forma en que se realizó el cálculo de las derivadas parciales usadas en este trabajo de tesis.

Capítulo 2

Fundamentos Teóricos

En este capítulo se presentan los aspectos teóricos en los que se basa este trabajo de tesis. Los conceptos de transformaciones lineales que permiten cambiar de posición y orientación a los objetos en el espacio tridimensional son tratados en la sección 2.1. En la sección 2.2 se presentan los aspectos de optimización que fueron considerados en la realización de esta tesis: optimización lineal por mínimos cuadrados, optimización no lineal con Levenberg-Marquardt y computación evolutiva. Así mismo se presentan aspectos relacionados con visión por computadora (sección 2.3) tales como: el modelo de la cámara, la generación de imágenes, la forma en que se modela la distorsión de la lente y la forma de estimar un punto en el espacio a partir de imágenes en las que aparece por medio de la triangulación. En la sección 2.4 se presenta lo relacionado al proceso de reconstrucción. Para poder medir la calidad de una reconstrucción es necesario contar con indicadores que nos permitan cuantificarla. Algunos ejemplos de estos cuantificadores son el error cuadrático medio y la incertidumbre, los cuales se presentan en la sección 2.5.

2.1. Transformaciones

Tanto en visión como en robótica es fundamental representar la posición y orientación de los objetos en un entorno. Cuando hablamos de objetos podemos hablar de robots, cámaras, obstáculos, caminos, etc.

Un punto en el espacio puede ser visto o representado como un vector de coordenadas. Este vector de coordenadas representa la traslación o desplazamiento de un punto con respecto a otro punto de referencia [19]. Generalmente, el punto de referencia es el origen del sistema coordenado (ver figura 2.1).

Frecuentemente, un objeto es visto como un conjunto de puntos. Si el objeto es rígido, los puntos que lo conforman mantienen posiciones constantes relativas a un punto de referencia local del objeto. Con esta representación es posible definir la posición y orientación (pose) del objeto en general en lugar de definir la posición para cada uno de los puntos que lo conforman (ver figura 2.2).

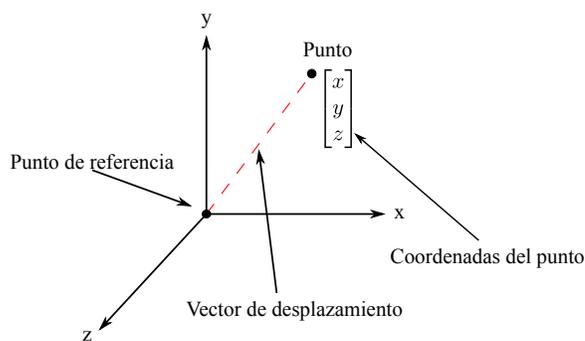


Figura 2.1: Punto en el espacio.

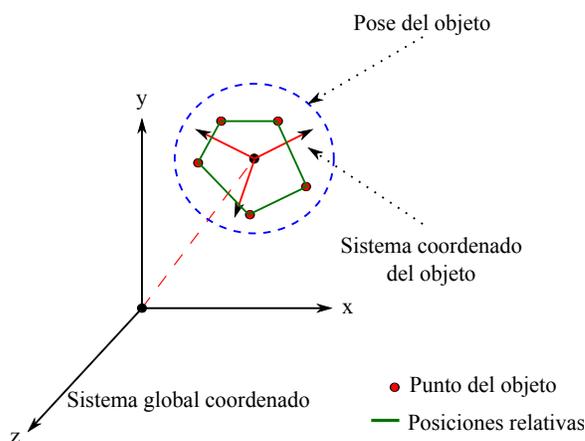


Figura 2.2: Objeto formado por puntos.

2.1.1. Traslación

La traslación es la operación que permite cambiar de posición a un punto o a un objeto en el espacio (ver figura 2.3). El operador de traslación se define como:

$$\mathbf{p}_t = \mathbf{p} + \mathbf{v}$$

$$\begin{bmatrix} x + t_x \\ y + t_y \\ z + t_z \end{bmatrix} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix} \quad (2.1)$$

La traslación también puede ser expresada utilizando puntos tridimensionales homogéneos:

$$\begin{bmatrix} x + t_x \\ y + t_y \\ z + t_z \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \quad (2.2)$$

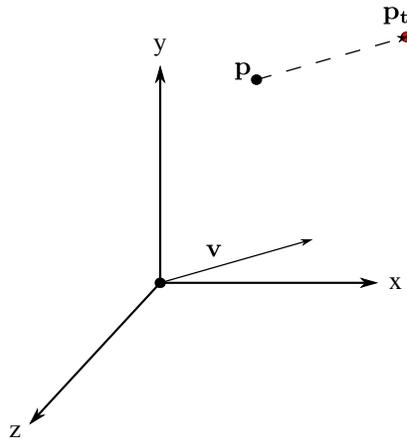


Figura 2.3: Traslación de un punto en el espacio.

2.1.2. Rotación

La rotación es la transformación que permite cambiar de orientación a los objetos en el espacio. Esta transformación ocurre alrededor de los ejes principales del sistema de coordenadas. En la figura 2.4 se muestra un punto rotando alrededor del eje y .

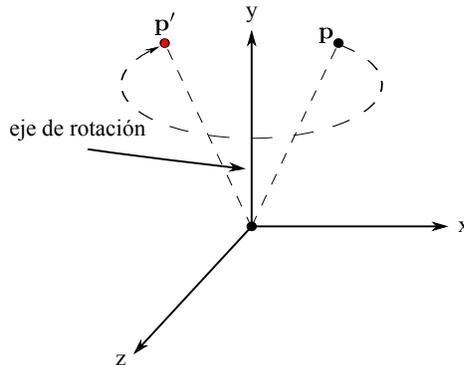


Figura 2.4: Ejemplo de rotación de un punto sobre el eje y .

Existen diversas formas para representar la rotación, cada una por ejemplo [20]:

- Ángulos de Euler.
- Cuaterniones.
- Ángulo-vector.

Cada una de estas representaciones tiene ventajas sobre las demás para ciertas aplicaciones. Hartley y Zisserman en [21] recomiendan usar la representación de ángulo vector, sin embargo en este trabajo de tesis se utilizan los ángulos de Euler debido a que no se reporta ningún problema con su uso en el trabajo de de la Fraga [18] que es el trabajo en el cual se basa esta tesis.

La representación en ángulos de Euler define la rotación a partir de tres ángulos θ_1 , θ_2 y θ_3 alrededor de los ejes principales del sistema de coordenadas. Existe una gran variedad de combinaciones de ejes para llevar a cabo esta rotación [20], sin embargo en este trabajo de tesis R se define utilizando la convención en la ecuación (2.3)

$$R = R_z(\theta_3) \cdot R_y(\theta_2) \cdot R_z(\theta_1) \quad (2.3)$$

donde:

$$R_z(\theta) = \begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad R_y(\theta) = \begin{bmatrix} \cos \theta & 0 & -\sin \theta \\ 0 & 1 & 0 \\ \sin \theta & 0 & \cos \theta \end{bmatrix} \quad (2.4)$$

2.1.3. Rotación y traslación

Es posible expresar la traslación y la rotación utilizando una matriz de transformación. Dado el vector de traslación $\mathbf{t} = [t_x, t_y, t_z]^T$ y la matriz de rotación R la matriz de transformación T se define como:

$$T \cdot R = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.5)$$

Es importante tomar en cuenta que al aplicar la matriz de transformación, se aplica primeramente la transformación de rotación y después la de traslación.

2.1.4. Escalamiento

La operación de escalamiento consiste en multiplicar por un escalar e los componentes del vector de traslación que definen a un punto. Al aplicar este operador sobre todos los puntos que conforman a un objeto, las proporciones, es decir las distancias entre ellos, también son escaladas (ver figura 2.5).

El escalamiento puede ser aplicado al multiplicar el vector que define al punto por la matriz con e en la diagonal. Así el escalamiento de un punto se define como:

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} e & 0 & 0 \\ 0 & e & 0 \\ 0 & 0 & e \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} \quad (2.6)$$

2.2. Métodos de estimación

En la ciencia es común encontrarse con el problema de encontrar o estimar el conjunto de parámetros que hacen que un modelo matemático se ajuste lo más posible a una serie de observaciones. Si se tiene el número exacto de observaciones para poder

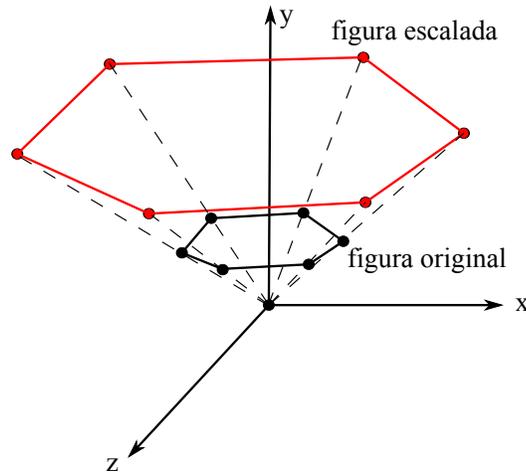


Figura 2.5: Ejemplo de escalamiento.

estimar el modelo, entonces el modelo se ajustará a la perfección a las observaciones con las que se estimó. Es común contar con un número de observaciones mayor al mínimo necesario pero en presencia de ruido, en este caso se opta por encontrar el conjunto de parámetros que mejor ajustan el modelo a las observaciones dadas aunque éste no se ajuste perfectamente a todas las observaciones. Ésto da origen al problema de resolver un sistema sobre-determinado de ecuaciones. Existen varias formas de afrontar el problema. En esta sección se describen aquellos enfoques considerados para la realización de este trabajo de tesis.

2.2.1. Optimización lineal por mínimos cuadrados

El problema de encontrar los parámetros que mejor ajustan un modelo a un conjunto de observaciones puede ser escrito en forma de matrices como sigue: dado el vector $\mathbf{b} \in \mathbb{R}^m$ y la matriz $A \in \mathbb{R}^{m \times n}$ con $m > n$, se desea encontrar un vector $\mathbf{x} \in \mathbb{R}^n$ tal que $A\mathbf{x}$ sea la aproximación a \mathbf{b} en la ecuación (2.7) [22] que minimice la norma el vector error formado por $\epsilon = A\mathbf{x} - \mathbf{b}$.

$$A\mathbf{x} \cong \mathbf{b} \quad (2.7)$$

Solución mediante ecuaciones normales

Aún cuando el sistema representado en la ecuación (2.7) es sobre-determinado, es posible encontrar una solución a \mathbf{x} . Dado que A pertenece al espacio $\mathbb{R}^{m \times n}$ la estimación de \mathbf{x} en $\mathbf{x} = A^{-1}\mathbf{b}$ no es factible pues A no es una matriz cuadrada y no se puede calcular su inversa, sin embargo una aproximación de \mathbf{x} puede obtenerse por mínimos cuadrados mediante ecuaciones normales [23] como se muestra en la ecuación (2.8).

$$\mathbf{x} = (A^T A)^{-1} A^T \mathbf{b} \quad (2.8)$$

Solución mediante el cálculo de la matriz pseudo inversa

Una opción para evitar el cálculo de $(A^T A)^{-1} A^T$ en la ecuación (2.8) es mediante el cálculo de la matriz pseudo inversa [24] A^+ que puede llevarse a cabo mediante la descomposición en valores singulares.

La matriz A puede ser descompuesta como en la ecuación (2.9)

$$A = U D V^T \quad (2.9)$$

donde U y V son dos matrices ortogonales y D es una matriz diagonal.

La matriz pseudo inversa A^+ es definida en la ecuación (2.10) y es la matriz que permite la estimación de \mathbf{x} en la ecuación (2.11)

$$A^+ = V D^+ U^T \quad (2.10)$$

donde: D^+ es el resultado de cambiar cada elemento de D por su recíproco si el elemento es diferente de cero.

$$\mathbf{x} = A^+ \mathbf{b} \quad (2.11)$$

2.2.2. Optimización no lineal por mínimos cuadrados

Volvamos al problema de encontrar los parámetros que mejor ajustan un modelo a un conjunto de observaciones. Si utilizamos los métodos de la sección 2.2.1, la estimación del modelo será buena en tanto la naturaleza del problema sea del tipo lineal. Cuando nos enfrentamos a problemas no lineales es necesario utilizar métodos que han sido desarrollados para tratar con ellos. El algoritmo Levenberg-Marquardt es un método iterativo. Este algoritmo es una extensión del algoritmo Gauss-Newton [21] diseñado para tener una convergencia más rápida además de tener un mecanismo de regulación para el caso de problemas sobre-determinados. El algoritmo puede verse como una mezcla entre el algoritmo de descenso de gradiente y el método iterativo de Newton, el cual funciona bien en la práctica y ha llegado a convertirse en un estándar en la solución al problema de mínimos cuadrados no lineales.

Supongamos una relación funcional $\mathbf{x} = f(\mathbf{p})$ donde: \mathbf{x} es un vector de observaciones y \mathbf{p} es un vector de parámetros, en espacios \mathbb{R}^n y \mathbb{R}^m respectivamente. Se cuenta con una medición próxima a la medición verdadera $\bar{\mathbf{x}}$ y deseamos encontrar el vector $\hat{\mathbf{p}}$ que más satisface la relación funcional. En otras palabras se busca el vector $\hat{\mathbf{p}}$ que satisface la ecuación $\mathbf{x} = f(\hat{\mathbf{p}}) - \epsilon$ donde: ϵ representa el vector error cuya norma es minimizada. Para resolver el caso donde f es no-lineal, debemos comenzar en un valor inicial \mathbf{p}_0 , el cual puede ser obtenido de resolver el problema con alguna técnica por mínimos cuadrados lineales (sección 2.2.1). \mathbf{p}_0 será refinado bajo la suposición de que la función f es localmente lineal. Sea $\epsilon_0 = f(\mathbf{p}_0) - \mathbf{x}$, asumimos que la función f es aproximada en \mathbf{p}_0 por $f(\mathbf{p}_0 + \Delta) = f(\mathbf{p}_0) + J\Delta$, donde: $J = \partial f / \partial \mathbf{p}$ es la matriz jacobiana.

El punto \mathbf{p}_0 pasa a \mathbf{p}_1 con $\mathbf{p}_1 = \mathbf{p}_0 + \Delta$, lo cual minimiza la ecuación (2.12).

$$f(\mathbf{p}_1) - \mathbf{x} = f(\mathbf{p}_0) + J\Delta - \mathbf{x} = \epsilon_0 + J\Delta \quad (2.12)$$

Se requiere minimizar $\epsilon + J\Delta$ sobre Δ , lo cual es un problema de minimización lineal. El vector Δ se calcula mediante las siguientes ecuaciones sabiendo que:

$$J^T J\Delta = -J^T \epsilon_0 \quad (2.13)$$

o bien usando la pseudo-inversa $\Delta = -J^+ \epsilon_0$. Así se obtiene el vector $\hat{\mathbf{p}}$ comenzando en el punto \mathbf{p}_0 y calculando las aproximaciones sucesivas de acuerdo a la fórmula de la ecuación (2.14).

$$\mathbf{p}_{i+1} = \mathbf{p}_i + \Delta_i \quad (2.14)$$

donde Δ_i es la solución al problema lineal por mínimos cuadrados $J\Delta_i = \epsilon_i$, J es la matriz jacobiana $\partial f/\partial \mathbf{p}$ evaluada en \mathbf{p}_i y $\epsilon_i = f(\mathbf{p}_i) - \mathbf{x}$.

Se quisiera que el algoritmo converja a la solución por mínimos cuadrados $\hat{\mathbf{p}}$. Desafortunadamente, es posible que lo haga a un óptimo local o bien que no converja, esto depende de que tan cercano esté \mathbf{p}_0 de $\hat{\mathbf{p}}$.

El método Newton y la matriz Hessiana

Consideremos el problema de encontrar el mínimo de funciones con muchas variables. Supongamos la función $g(\mathbf{p})$ donde: \mathbf{p} es un vector de parámetros. Supongamos también que $g(\mathbf{p})$ tiene un valor mínimo bien definido y que conocemos un punto inicial \mathbf{p}_0 razonablemente cercano al mínimo. Es necesario expandir $g(\mathbf{p})$ en \mathbf{p}_0 mediante series de Taylor para obtener:

$$g(\mathbf{p}_0 + \Delta) = g + g_p \Delta + \Delta^T g_{pp} \Delta / 2 + \dots \quad (2.15)$$

donde: el subíndice p denota diferenciación y el lado derecho es evaluado en \mathbf{p}_0 .

Deseamos minimizar esta cantidad respecto a Δ . Para esto diferenciamos con respecto a Δ e igualamos a cero, llegando a la ecuación $g_p + g_{pp} \Delta = 0$ ó:

$$g_{pp} \Delta = -g_p \quad (2.16)$$

En esta ecuación g_{pp} es la matriz de segundas derivadas (la Hessiana de g), cuyo i - j -ésimo elemento es $\partial^2 g / \partial \mathbf{p}_i \partial \mathbf{p}_j$, con \mathbf{p}_i y \mathbf{p}_j como los i -ésimo y j -ésimo parámetros respectivamente.

El vector g_p es el gradiente de g . El método de Newton consiste en iniciar en un valor \mathbf{p}_0 e iterativamente calcular los incrementos Δ usando la ecuación (2.16) hasta la convergencia.

Cambiamos el tipo de función costo que surge de la minimización por mínimos cuadrados, así $g(\mathbf{p})$ es ahora la norma de una función error:

$$g(\mathbf{p}) = \frac{1}{2} \|\epsilon(\mathbf{p})\|^2 = \frac{\epsilon(\mathbf{p})^T \epsilon(\mathbf{p})}{2} \quad (2.17)$$

donde $\epsilon(\mathbf{p}) = f(\mathbf{p}) - \mathbf{x}$ es una función del vector \mathbf{p} , el factor $\frac{1}{2}$ está presente para simplificar los cálculos siguientes.

El vector gradiente g_p es fácilmente calculado como $\epsilon_p^T \epsilon$. Diferenciando $g_p = \epsilon_p^T \epsilon_p$ por segunda vez obtenemos la fórmula del Hessiano de g :

$$g_{pp} = \epsilon_p^T \epsilon_p + \epsilon_{pp}^T \epsilon \quad (2.18)$$

Bajo la suposición de que $f(\mathbf{p})$ es lineal, el segundo término a la derecha se elimina teniendo así que $g_{pp} = \epsilon_p^T \epsilon_p$. Sustituyendo el Hessiano y el gradiente en la ecuación (2.16) obtenemos $J^T J \Delta = -J^T \epsilon$ la cual corresponde a la ecuación (2.13), lo que significa que llegamos al mismo proceso iterativo de Newton bajo la suposición de que la ecuación (2.13) es una aproximación razonable de la matriz Hessiana de la función $g(\mathbf{p})$. Este procedimiento donde $J^T J$ es usado como una aproximación de la matriz Hessiana es conocido como el método Gauss-Newton [21].

Levenberg-Marquardt

El método Levenberg-Marquardt es una variación del método Gauss-Newton donde la ecuación (2.16) es reemplazada por la ecuación aumentada:

$$(J^T J + \lambda I) \Delta = J^T \epsilon, \quad (2.19)$$

para algún valor de λ que varía de iteración en iteración. I es la matriz identidad, que toma como valor típico inicial $\lambda = 10^{-3}$ veces el promedio de los elementos de la diagonal $N = J^T J$. Si el valor de Δ , obtenido de resolver la ecuación (2.19) lleva a una reducción del error, entonces el incremento es aceptado y λ es dividido por un factor (típicamente 10) antes de la siguiente iteración. Por otro lado, si el valor de Δ obtenido lleva a un incremento del error, λ es multiplicado por el mismo factor y la ecuación (2.19) es resuelta nuevamente. Este proceso es repetido hasta que se encuentre el valor Δ que lleva a la reducción del error. El proceso de resolver repetidamente la ecuación (2.19) para diferentes valores de λ hasta obtener un Δ aceptable es considerado como una iteración del algoritmo Levenberg-Marquardt [21].

2.2.3. Optimización no lineal con algoritmos evolutivos

Se conocen como algoritmos evolutivos a aquellos métodos de optimización que basados en las teorías Darwinianas de la evolución aplican una serie de operadores como, recombinación, mutación y elitismo (supervivencia del mejor individuo) para obtener la mejor solución dado un problema de minimización o maximización. Entre estos algoritmos se encuentran las llamadas estrategias evolutivas desarrollados por Rechenberg en 1973 [25] y Schwefel en 1994 [26], así como los algoritmos genéticos que son atribuidos a Holland en 1962 [27] y Goldberg en 1989 [28].

Cada enfoque tiene sus diferencias así como sus ventajas. Sin embargo, en general se puede decir que las estrategias evolutivas son efectivas como optimizadores de

funciones continuas. Esto se debe en parte a que codifican los parámetros como valores reales y los manipulan mediante operadores aritméticos. En contraste, los algoritmos genéticos son presumiblemente mejores para problemas de optimización combinatoria, debido a que codifican los parámetros como cadenas de bits y los modifican mediante operadores lógicos. Modificar un algoritmo genético para poder utilizar parámetros codificados en números reales con el fin de aplicarlos en la optimización de parámetros continuos es típicamente transformarlo en un tipo de estrategia evolutiva [29, 30].

Evolución diferencial

La evolución diferencial es un tipo de algoritmos evolutivo, propuesto por Kenneth Price y Raíner Stor para optimización en espacios continuos, el cual es versátil y fácil de usar. La primera publicación de la evolución diferencial fue como reporte técnico en 1995 [31]. Desde entonces, la evolución diferencial ha probado su efectividad tanto en concursos de optimización (p. ej. la Competencia Internacional en Optimización Evolutiva de la IEEE en 1996 y 1997) así como en una gran variedad de aplicaciones del mundo real [32].

La evolución diferencial es un optimizador poblacional (la población es un conjunto de soluciones, en donde cada solución es llamada individuo). La versión clásica inicia la evolución muestreando el espacio de búsqueda con individuos generados aleatoriamente como se muestra en la figura 2.6(a). Cada individuo es indexado con un número entre 0 y $N - 1$. La evolución diferencial genera nuevos individuos que son resultado de perturbar los existentes en la población.

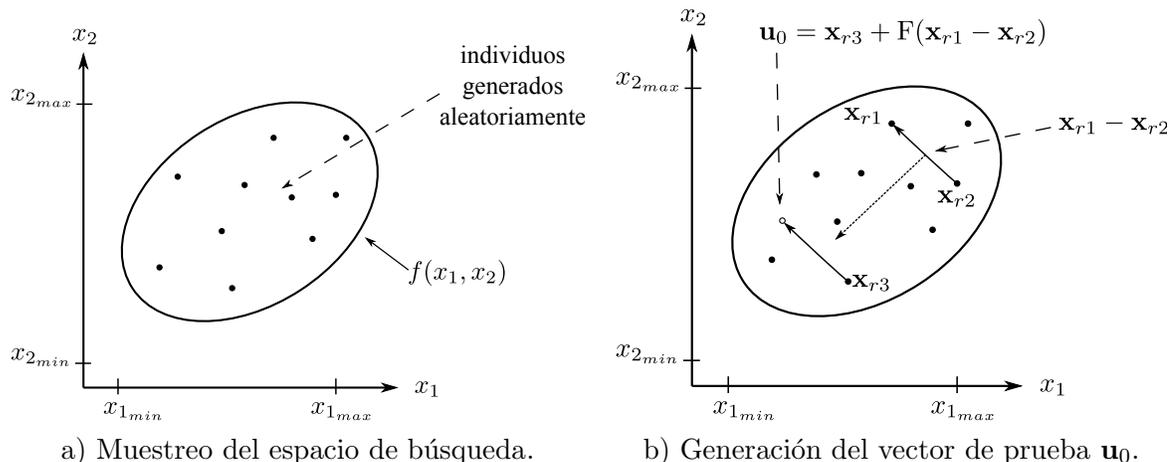


Figura 2.6: Etapas de la evolución diferencial.

En la figura 2.6(b) el vector de prueba \mathbf{u}_0 es el resultado de sumar la diferencia escalada de dos vectores seleccionados aleatoriamente de la población a un tercer vector. El vector generado \mathbf{u}_0 compite con el vector de la población con el mismo índice (en este caso, el individuo cero). Si el vector \mathbf{u}_0 tiene una mejor evaluación de

la función objetivo (aptitud) reemplaza al elemento con el que compitió. El mismo procedimiento se repite para todos los índices desde 0 hasta $N - 1$. Esto se conoce como una generación del algoritmo.

Existen varias versiones de evolución diferencial, sin embargo en este trabajo de tesis se usó la versión rand/1/bin, la cual incorpora una probabilidad de recombinación de parámetros R para la creación del individuo de prueba (ver algoritmo 1).

Entrada: Límites de búsqueda, condición de paro s . Tamaño de población N , máximo número de generaciones M , constante de diferenciación F y probabilidad de recombinación R .

Salida: Una solución al problema de minimización

```

1: Inicializar  $X = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\}$ 
2: Evaluar( $X$ )
3:  $k = 0$ 
4: repetir
5:   para  $j = 1$  hasta  $N$  hacer
6:     Seleccionar 3 números aleatorios  $r_1, r_2$  y  $r_3$  entre  $[1, N]$  con  $r_1 \neq r_2 \neq r_3$ 
7:     Elegir un número aleatorio  $i_{\text{rand}}$  entre  $[1, n]$ 
8:     para  $i = 1$  hasta  $n$  hacer
9:        $\mathbf{u}_{i,j} = \begin{cases} x_{i,r3} + F(x_{i,r1} - x_{i,r2}) & \text{si } U(0, 1) < R \text{ o } i = i_{\text{rand}} \\ x_{i,j} & \text{en otro caso} \end{cases}$ 
10:    fin para
11:     $\mathbf{u}_{n+1,j} = \text{evaluar}(\mathbf{u}_j)$ 
12:    si  $\mathbf{u}_{n+1,j} < \mathbf{x}_{n+1,j}$  entonces
13:       $\mathbf{x}_j = \mathbf{u}_j$ 
14:    fin si
15:     $\min = x_{n+1,1}$ ,  $\max = x_{n+1,1}$ 
16:    para  $i = 2$  hasta  $n$  hacer
17:      si  $x_{n+1,i} < \min$  entonces
18:         $\min = x_{n+1,i}$ 
19:      fin si
20:      si  $x_{n+1,i} > \max$  entonces
21:         $\max = x_{n+1,i}$ 
22:      fin si
23:    fin para
24:     $k = k + 1$ 
25:  fin para
26: hasta que  $(\max - \min) < s$  o  $k > M$ 

```

Algoritmo 1: Evolución diferencial versión rand/1/bin.

2.3. Conceptos de Visión por Computadora

Para poder llevar a cabo la recuperación de la estructura tridimensional a partir de imágenes en dos dimensiones es necesario comprender cómo se forman estas últimas. El capturar un entorno tridimensional en una imagen bidimensional es resultado de un proceso proyectivo, en el cual se pierde una dimensión [21]. En esta sección se tratan aquellos aspectos relacionados con la generación de imágenes, así como la recuperación de información 3D a partir de más de dos vistas (figura 2.7).

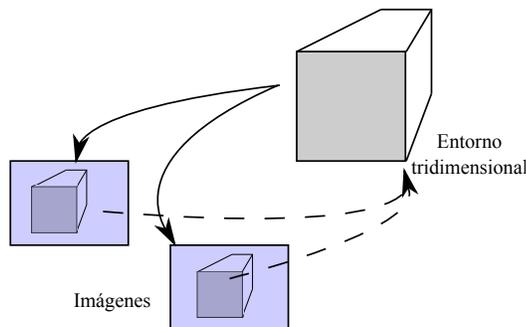


Figura 2.7: Generación de imágenes y recuperación de información 3D.

2.3.1. Modelo de la cámara oscura

Cuando hablamos del modelo de la cámara nos referimos al modelo matemático que explica las relaciones entre los puntos en el espacio tridimensional y su correspondencia en una fotografía o imagen tomada por esa cámara. De forma general, el modelo de la cámara involucra estimar la transformación en perspectiva que aplica la cámara, conocer el aspecto de los píxels (si son cuadrados o rectangulares) y la deformación que presentan las imágenes tomadas debido a la distorsión o las aberraciones de la lente.

El modelo más ampliamente utilizado es el de la *cámara oscura*, el cual asume que todos los puntos de una escena son proyectados sobre un plano llamado *plano imagen*, a través de un rayo que pasa por un punto llamado *centro de proyección*, lo que provoca que la imagen formada sobre el *plano imagen* esté invertida (véase Figura 2.8).

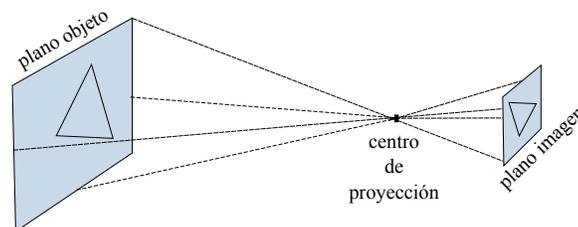


Figura 2.8: Modelo clásico de la cámara oscura.

Para evitar la introducción de signos negativos en las ecuaciones, que reflejen la inversión de la imagen, se utiliza un modelo (que no existe en realidad pero es más intuitivo) donde el plano imagen se sitúa delante del centro de proyección, como se ve en la Figura 2.9.

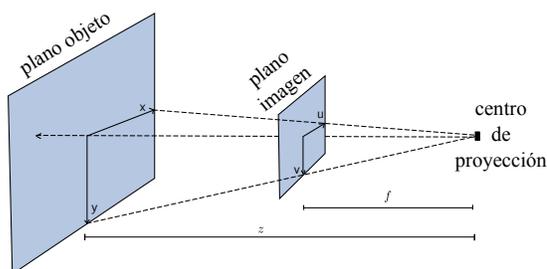


Figura 2.9: Modelo de la cámara oscura.

De manera formal, un punto en el espacio tridimensional representado de forma homogéneas está definido por:

$$\mathbf{P} = \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \quad (2.20)$$

Un punto sobre el *plano imagen* (*punto en la imagen*) se define por:

$$\mathbf{p} = \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \quad (2.21)$$

La relación existente entre un *punto en la imagen* \mathbf{p} y un *punto en el mundo* \mathbf{P} es:

$$\lambda \mathbf{p} = M \mathbf{P}$$

$$\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = M \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \quad (2.22)$$

donde $M \in \mathbb{R}^{(3 \times 4)}$ mantiene la información necesaria para transformar un punto en el mundo a la imagen y λ es un factor de escala.

M , por otro lado, está definida como sigue:

$$M = K[R|\mathbf{t}] \quad (2.23)$$

donde $K \in \mathbb{R}^{3 \times 3}$ mantiene la información de los *parámetros intrínsecos* de la cámara.

- f : *distancia focal*.

- (u_0, v_0) : *punto principal* en la imagen.
- o : *oblicuidad* de los ejes en la imagen.

Así, K se define en su versión básica como:

$$K = \begin{bmatrix} f & 0 & u_0 \\ 0 & f & v_0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.24)$$

y la versión completa la cual considera la oblicuidad o y la posibilidad de tener píxeles no cuadrados:

$$K = \begin{bmatrix} f_x & o & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.25)$$

$R \in \mathbb{R}^{3 \times 3}$ y $\mathbf{t} \in \mathbb{R}^3$ definen la *rotación* y la *translación* del objeto (o de la escena) con respecto a un sistema de coordenadas global.

- R depende de 3 parámetros θ_1, θ_2 y θ_3 que corresponden a los ángulos de rotación de Euler como ya se explicó en la sección 2.1.2 en la página 9.
- \mathbf{t} corresponde al vector de *translación*.

$$\mathbf{t} = \begin{bmatrix} t_1 \\ t_2 \\ t_3 \end{bmatrix} \quad (2.26)$$

De esta forma $\theta_1, \theta_2, \theta_3, t_1, t_2$ y t_3 son conocidos como los *parámetros extrínsecos* de la cámara.

2.3.2. Calibración de la cámara

El proceso de calibración de la cámara consiste en obtener los parámetros intrínsecos y extrínsecos de la misma. Las técnicas de calibración se basan en el uso de un conjunto de puntos formando un patrón conocido, así como en vistas del mismo desde diferentes posiciones (ver figura 2.10). Generalmente, estos patrones suelen estar sobre un entorno plano (un ejemplo de esto son los tableros de ajedrez). Existen muchos trabajos [33, 34, 35, 36, 37, 38] que utilizan patrones de este tipo.

Conociendo el patrón y los puntos que lo conforman apareciendo en las distintas imágenes es posible determinar los parámetros intrínsecos, los extrínsecos e incluso es posible determinar el grado de distorsión introducido por la lente [19].

Técnicamente podemos decir que el proceso de calibración asume que una cámara observa n elementos geométricos como pueden ser puntos o líneas con posición conocida (patrón) en algún sistema coordenado. Esta información es utilizada para calcular la matriz de proyección M asociada a cada imagen, de la cual son extraídos

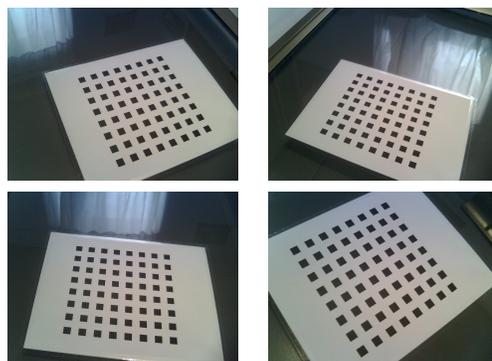


Figura 2.10: Ejemplos de fotografías usadas para la calibración de la cámara.

los parámetros intrínsecos y extrínsecos.

Los investigadores se han esmerado en desarrollar métodos de calibración rápidos para cámaras comunes. Un ejemplo de esto es el método propuesto por Zhang [39], el cual utiliza un patrón plano que puede ser generado usando una impresora láser. También existen métodos que utilizan patrones planos circulares, como el método propuesto por Heikillä del cual la teoría y los detalles de implementación pueden encontrarse en [35]. El campo de calibración de cámaras es sin embargo amplio. Si se desea ver una evaluación de los métodos comunes para calibración, puede consultarse el trabajo de Sun y Cooperstock [40], donde se revisan trabajos como los de Tsai, Heikillä y Zhang y se proveen los detalles prácticos de los mismos. Este trabajo puede servir además como introducción al campo de calibración de cámaras ya que posee una breve revisión de la literatura reciente sobre el tema. Otra fuente de información importante es el libro de Gruen y Huang titulado “Calibration and Orientation of Cameras in Computer Vision“ [34] donde se puede encontrar una vista general de los métodos de calibración de cámaras.

Es importante tomar en cuenta que la precisión de la medición del patrón influye en la precisión de los parámetros resultantes de la calibración. Estos efectos han sido analizados por Lavest et al en [41].

La mayoría de los métodos de calibración se basan en la ecuación (2.22). Un ejemplo de estos métodos clásicos es el propuesto por Tsai [33, 42, 43, 44]. Este método utiliza directamente la ecuación (2.22) para encontrar la matriz de proyección M . La matriz M es una combinación lineal de los parámetros internos y los externos de la cámara en la ecuación (2.23), de ésta es posible obtener la relación entre los puntos en el espacio y en la imagen (ec. 2.27).

$$\begin{aligned}
 u &= \frac{m_{11}x + m_{12}y + m_{13}z + m_{14}}{m_{31}x + m_{32}y + m_{33}z + m_{34}} \\
 v &= \frac{m_{21}x + m_{22}y + m_{23}z + m_{24}}{m_{31}x + m_{32}y + m_{33}z + m_{34}}
 \end{aligned}
 \tag{2.27}$$

Debido a que la matriz de proyección M se conoce hasta un factor de escala, se tienen 11 elementos a determinar. Conociendo por lo menos seis puntos en las imágenes o cuatro si se asume que $z = 0$ además de la ecuación (2.27) es posible obtener los elementos de M .

En la práctica generalmente se tienen más del número mínimo de puntos por lo que el problema se convierte en resolver la ecuación (2.28).

$$Q\mathbf{m} = 0 \quad (2.28)$$

donde la matriz Q se define como sigue:

$$Q = \begin{bmatrix} x_1 & y_1 & z_1 & 1 & 0 & 0 & 0 & 0 & -u_1x_1 & -u_1y_1 & -u_1z_1 & -u_1 \\ 0 & 0 & 0 & 0 & x_1 & y_1 & z_1 & 1 & -v_1x_1 & -v_1y_1 & -v_1z_1 & -v_1 \\ \vdots & & & & \vdots & & & & \vdots & & & \vdots \\ 0 & 0 & 0 & 0 & x_n & y_n & z_n & 1 & -v_nx_n & -v_ny_n & -v_nz_n & -v_n \end{bmatrix} \quad (2.29)$$

La solución a la ecuación (2.28) puede obtenerse por medio de la descomposición en valores singulares $Q = SVD^T$ [45, 46]. La solución es el vector columna de D correspondiente al eigenvalor más pequeño en V .

Una vez que se ha calculado la matriz de proyección M se procede a calcular el valor de la distancia focal en base a los elementos de la matriz M y el punto principal con lo que obtenemos una estimación de los parámetros intrínsecos de la cámara como se muestra en la ecuación (2.30).

$$f = -\sqrt{a - u_0^2 - v_0^2} \quad (2.30)$$

donde a se define como:

$$a = -\frac{m_{12}(m_{11} + m_{31}(-u_0)) + M_{22}(m_{21} + m_{31}(-v_0)) + m_{32}(m_{11}(-u_0) + m_{21}(-v_0))}{m_{31}m_{32}}$$

Para obtener la matriz K del modelo simple de la cámara basta con sustituir los valores de f , u_0 y v_0 en la ecuación (2.24).

El siguiente paso es calcular los parámetros extrínsecos de la cámara, para esto calculamos la matriz de rotación R y el vector de traslación \mathbf{t} como lo muestran las ecuaciones (2.31) y (2.32).

$$\mathbf{t} = \frac{K^{-1}M_{\text{col3}}}{\lambda} \quad (2.31)$$

$$R = [\mathbf{r}_1 \quad \mathbf{r}_2 \quad \mathbf{r}_1] \quad (2.32)$$

donde:

$$\mathbf{r}_1 = \frac{K^{-1}M_{\text{col1}}}{\lambda} \quad (2.33)$$

$$\mathbf{r}_2 = \frac{K^{-1}M_{\text{col}2}}{\lambda} \quad (2.34)$$

$$\mathbf{r}_3 = \frac{K^{-1}M_{\text{col}3}}{\lambda} \quad (2.35)$$

$$\lambda = \|K^{-1}M_{\text{col}1}\| \quad (2.36)$$

Los ángulos de rotación se obtienen a partir de la matriz R con las ecuaciones (2.37), (2.38), y (2.39).

$$\theta_1 = \text{atan2}(R_{32}, -R_{31}) \quad (2.37)$$

$$\theta_2 = \text{acos}(R_{33}) \quad (2.38)$$

$$\theta_3 = \text{atan2}(R_{23}, R_{13}) \quad (2.39)$$

La matriz de rotación R debe ser ortogonal; en caso de no serlo es posible obtener una versión ortogonal R' que se obtiene cambiando la matriz diagonal resultado de la descomposición por valores singulares de la matriz R por la matriz identidad I como se muestra en la ecuación 2.40.

$$R' = UIV^T \quad (2.40)$$

Cuando se lleva a cabo el proceso de obtención de los parámetros intrínsecos y extrínsecos sin utilizar el modelo del patrón el proceso es conocido como auto-calibración. En [37, 38] puede encontrarse información sobre este tipo de métodos y en la sección 3.1 se describe un método [18] basado en evolución diferencial.

2.3.3. Triangulación

El proceso de triangulación consiste en obtener la posición en el espacio (coordenada 3D) asociada a una correspondencia como se muestra en la figura 2.11.

Cuando se conocen los parámetros internos y externos de una cámara, y los parámetros externos de varias tomas de esa misma cámara, el proceso de triangulación (obtención de un punto \mathbf{P}) puede resolverse de la siguiente forma:

A partir de (2.22) se obtienen las siguientes expresiones, donde m_{ij} corresponden a los elementos de la matriz de proyección:

$$\lambda u = xm_{11} + ym_{12} + zm_{13} + m_{14}, \quad (2.41)$$

$$\lambda v = xm_{21} + ym_{22} + zm_{23} + m_{24}, \quad (2.42)$$

$$\lambda = xm_{31} + ym_{32} + zm_{33} + m_{34}, \quad (2.43)$$

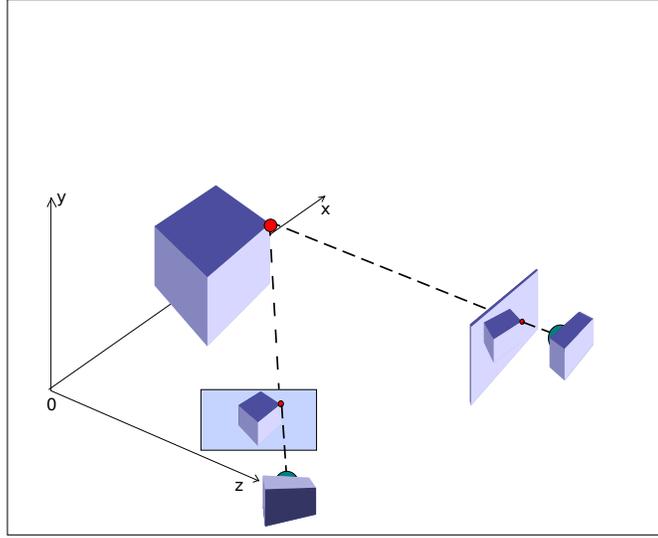


Figura 2.11: Triangulación de una correspondencia.

Al sustituir (2.43) en (2.41) y (2.42) se obtiene:

$$(um_{31} - m_{11})x + (um_{32} - m_{12})y + (um_{33} - m_{13})z = m_{14} - um_{34}, \quad (2.44)$$

$$(vm_{31} - m_{21})x + (vm_{32} - m_{22})y + (vm_{33} - m_{23})z = m_{24} - vm_{34}, \quad (2.45)$$

Expresando (2.44) y (2.45) en forma matricial se obtiene:

$$\begin{bmatrix} (um_{31} - m_{11}) & (um_{32} - m_{12}) & (um_{33} - m_{13}) \\ (vm_{31} - m_{21}) & (vm_{32} - m_{22}) & (vm_{33} - m_{23}) \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} m_{14} - um_{34} \\ m_{24} - vm_{34} \end{bmatrix}, \quad (2.46)$$

Con (2.46) se pueden generar 2 ecuaciones para un punto $\mathbf{p} = [u, v]^T$ dado. Ya que se tienen 2 puntos $\{\mathbf{p}_1\}, \{\mathbf{p}_2\}$ de la correspondencia, se sobre-determina el sistema obteniendo así 4 ecuaciones.

De las 4 ecuaciones generadas se obtiene la siguiente expresión:

$$\mathbf{Ax} = \mathbf{b}. \quad (2.47)$$

donde:

$$\mathbf{x} = \begin{bmatrix} x \\ y \\ z \end{bmatrix},$$

$$\mathbf{A} = \begin{bmatrix} (u_1 m_{31}^1 - m_{11}^1) & (u_1 m_{32}^1 - m_{12}^1) & (u_1 m_{33}^1 - m_{13}^1) \\ (v_1 m_{31}^1 - m_{21}^1) & (v_1 m_{32}^1 - m_{22}^1) & (v_1 m_{33}^1 - m_{23}^1) \\ (u_2 m_{31}^2 - m_{11}^2) & (u_2 m_{32}^2 - m_{12}^2) & (u_2 m_{33}^2 - m_{13}^2) \\ (v_2 m_{31}^2 - m_{21}^2) & (v_2 m_{32}^2 - m_{22}^2) & (v_2 m_{33}^2 - m_{23}^2) \end{bmatrix},$$

$$\mathbf{b} = \begin{bmatrix} m_{14}^1 - u_1 m_{34}^1 \\ m_{24}^1 - v_1 m_{34}^1 \\ m_{14}^2 - u_2 m_{34}^2 \\ m_{24}^2 - v_2 m_{34}^2 \end{bmatrix},$$

donde: $m_{i,j}^k$ es el elemento en la fila i columna j de la matriz $M^k = K[R_k|\mathbf{t}_k]$, $k = 1, 2$ y (u_k, v_k) son los elementos de los puntos $\{\mathbf{p}_k\}$.

Así, para obtener \mathbf{x} de (2.47):

$$\mathbf{x} = A^+ \mathbf{b}. \quad (2.48)$$

De esta forma \mathbf{P} es:

$$\mathbf{P} = \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}.$$

La ecuación (2.48) es la estimación de la solución lineal por mínimos cuadrados de la ecuación (2.47). Se debe aplicar una estimación por mínimos cuadrados debido a que tenemos un sistema sobre determinado de ecuaciones, o en otras palabras, la matriz A en (2.47) tiene más renglones que columnas. Sin embargo, el problema puede ser resuelto mediante la descomposición en valores singulares como en la ecuación (2.28).

Existen otros métodos [47, 21] de triangulación los cuales se basan en métodos no lineales.

2.3.4. Modelo de distorsión

En el modelo de la cámara oscura de la ecuación (2.22) una línea 3D es proyectada a la imagen también como una línea, sin embargo con las cámaras reales esta característica decrece conforme la distancia focal y el precio de la lente del dispositivo son menores [21]. El efecto de distorsión proviene de la forma de los lentes. El efecto que la distorsión provoca sobre las imágenes es la curvatura y es más notable en la periferia de la imagen (ver figura 2.12).

El proceso de distorsión actúa sobre los puntos idealmente generados (puntos no distorsionados) con el modelo de la cámara oscura de la ecuación (2.22). El modelo de la distorsión radial de la lente se define en la ecuación (2.49).

$$\begin{pmatrix} u_d \\ v_d \end{pmatrix} = \begin{pmatrix} u \\ v \end{pmatrix} + \begin{pmatrix} u - u_0 \\ v - v_0 \end{pmatrix} \cdot d \quad (2.49)$$

donde (u, v) corresponden a las coordenadas de un punto sin distorsión (obedecen a un modelo de proyección lineal), (u_d, v_d) son las coordenadas del punto (u, v) después de aplicar la distorsión y d corresponde al factor de distorsión definido en la ecuación (2.50).

$$d = k_1 r^2 + k_2 r^4 \quad (2.50)$$

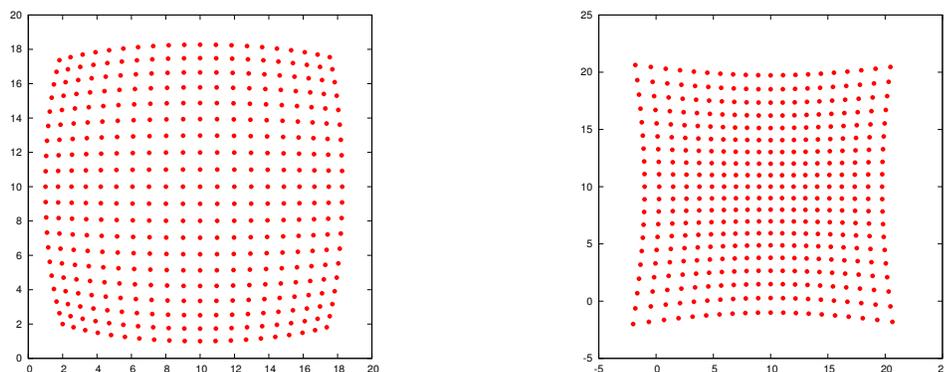


Figura 2.12: Efectos de la distorsión radial.

donde (k_1, k_2) corresponden a los coeficientes de distorsión asociados a la cámara y r es la distancia radial desde el punto principal el cual es definido en la ecuación (2.51).

$$r = \sqrt{x^2 + y^2} \quad (2.51)$$

donde x y y se conocen como las coordenadas normalizadas que se obtienen según las ecuaciones (2.52) y (2.53) respectivamente.

$$x = \frac{u - \left(\frac{\rho}{f_y}\right)(v - v_0) - u_0}{f_x} \quad (2.52)$$

$$y = \frac{v - v_0}{f_y} \quad (2.53)$$

Corrección de la distorsión de la lente

Cuando se conocen los coeficientes de distorsión con que fue generada una imagen es posible corregir o eliminar los efectos de la distorsión radial (ver figura 2.13) de la ecuación (2.49) con el modelo inverso. Es decir, quitar la distorsión consiste en llevar (u_d, v_d) a (u, v) .

El modelo inverso de distorsión se define en la ecuación (2.54)

$$\begin{aligned} u &= \frac{u_d + du_0}{1 + d} \\ v &= \frac{v_d + dv_0}{1 + d} \end{aligned} \quad (2.54)$$

2.4. Reconstrucción

El problema de reconstrucción consiste en obtener la estructura tridimensional del entorno sin tener información tridimensional *a priori* de la misma. La solución

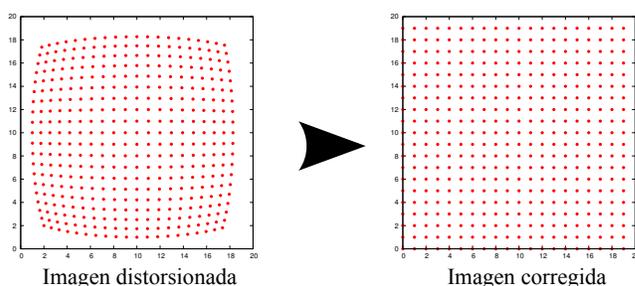


Figura 2.13: Corrección de la distorsión de una imagen.

al problema de reconstrucción es un proceso incremental en el cual lo primero que se tiene que resolver es la calibración de la cámara, esto es, obtener sus parámetros intrínsecos. Luego se procesa una serie de imágenes con correspondencias de puntos de interés o marcas. Estas correspondencias de puntos de interés son utilizadas para poder localizar cámaras o bien para poder triangular nuevos puntos que para ser agregados a la reconstrucción. El proceso se ilustra en la figura 2.14.

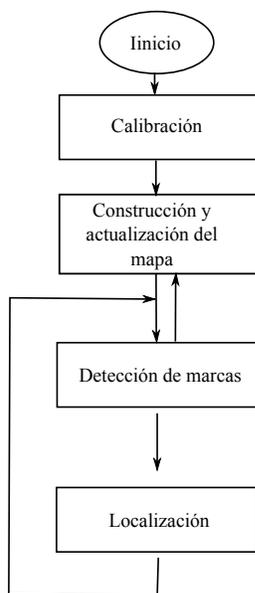


Figura 2.14: Proceso de reconstrucción.

Durante el proceso de reconstrucción se van acumulando errores. Estos errores generan incertidumbre tanto en la estimación de las posiciones de la cámara como en las posiciones de las puntos de la reconstrucción (ver figura 2.15).

La acumulación de errores es la característica que hace al problema de reconstrucción un problema difícil pues la acumulación de errores entorpece la localización de cámaras, así mismo puede llegar a deteriorar gravemente la reconstrucción de la escena. En el proceso reconstrucción los errores provienen de distintas fuentes. Estas fuentes van desde errores de medición de los sensores, errores en la estimación de

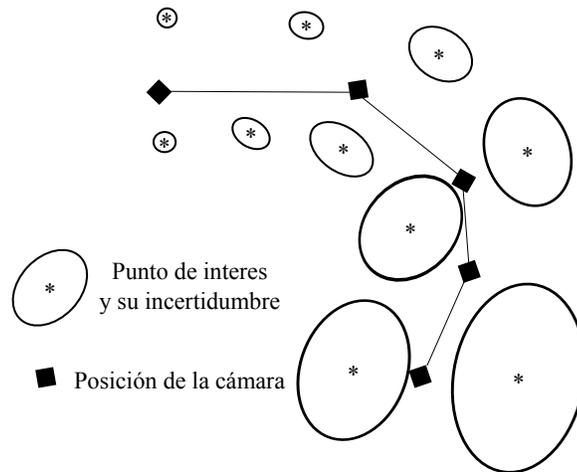


Figura 2.15: Incremento de incertidumbre durante el proceso de reconstrucción.

la posición. Una forma de reducir el efecto de acumulación de errores es regresar a lugares previamente visitados por la cámara. Usualmente este procedimiento se conoce como *cerrado de ciclos* y permite reducir significativamente las incertidumbres mediante la corrección tanto de las posiciones de la cámara como los puntos de la reconstrucción (ver figura 2.16).

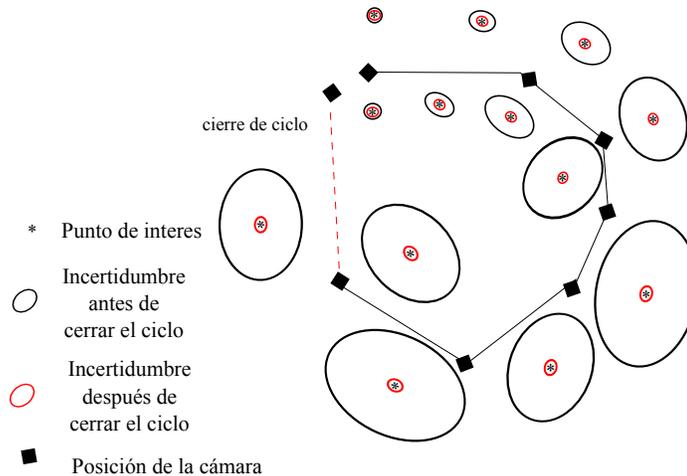


Figura 2.16: Ilustración del “cerrado de ciclo” durante el proceso de reconstrucción.

2.5. Calidad de la reconstrucción

En visión por computadora se necesita medir la calidad de una reconstrucción, es decir la posición de los puntos y las cámaras, así como su orientación. En esta sección se describen las formas disponibles para poder cuantificar la calidad de una reconstrucción, desde el punto de vista de visión por computadora.

Error de reproyección

Dado un punto en una imagen y su punto asociado en la reconstrucción 3D, el error de reproyección es el resultado de comparar el punto en la imagen con la proyección del punto 3D (ver figura 2.17) aplicando el modelo de la cámara oscura de la ecuación (2.22) con los parámetros intrínsecos y extrínsecos de la imagen en la que es visible el punto. El error de reproyección se define en la ecuación (2.55).

De manera formal, un el punto i visto en la imagen j se denota como \mathbf{p}_{ij} y su reproyección por $\lambda \hat{\mathbf{p}}_{ij} = M_j \mathbf{P}_i$, de tal forma que:

$$\|\mathbf{p}_{ij} - \hat{\mathbf{p}}_{ij}\| \quad (2.55)$$

representa el error de reproyección del punto i en la imagen j .

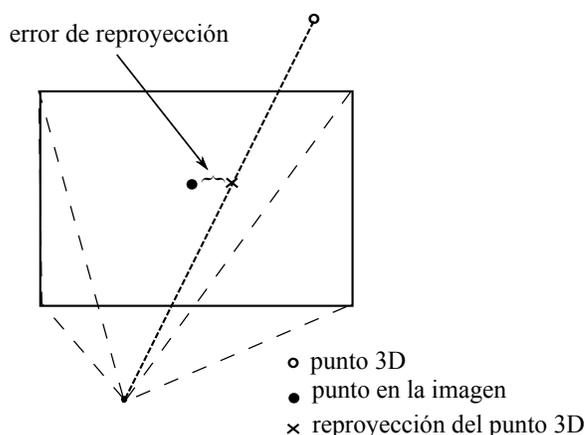


Figura 2.17: Error de reproyección.

Una reconstrucción generalmente tiene más de un punto. Cuando se desea calcular el error de reproyección del sistema puede considerarse sumar el error de reproyección de todos los puntos en la escena. La ec. (2.56) expresa el error de reproyección total del sistema

$$\sum_{i=1}^n \sum_{j=1}^m \|\mathbf{p}_{ij} - \hat{\mathbf{p}}_{ij}\| \quad (2.56)$$

donde m es el número de imágenes y n es el número de puntos visibles en la imagen i .

Cabe mencionar que el error de reproyección puede verse como el residual en donde el modelo puede ser el punto 3D o bien los parámetros de la cámara.

Error cuadrático medio

El error cuadrático medio o RMS por sus siglas en inglés es un indicador que permite conocer qué tan bueno es un modelo calculado a partir de observaciones (ver sección 2.2, en la página 10). Este indicador se basa en los errores que existen entre cada observación y el modo ajustado también llamado residual (ver figura 2.18).

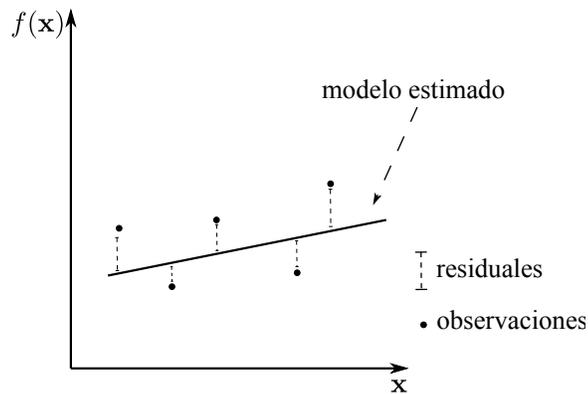


Figura 2.18: Residuales de las observaciones.

Cada punto en la figura 2.18 tiene un residuo $r_i = \mathbf{b} - A\mathbf{x}$ (ver sección 2.2.1) que representa el error cometido por el modelo estimado con la observación i [48]. El tamaño global del error o RMS definido en la ecuación (2.57) representa la distancia a la que se halla un punto típico al modelo con una probabilidad aproximada del 68 % o del 95 % en el caso de 2 RMS (ver figura 2.19).

$$\text{RMS} = \sqrt{\frac{r_1^2 + \dots + r_n^2}{n}} \quad (2.57)$$

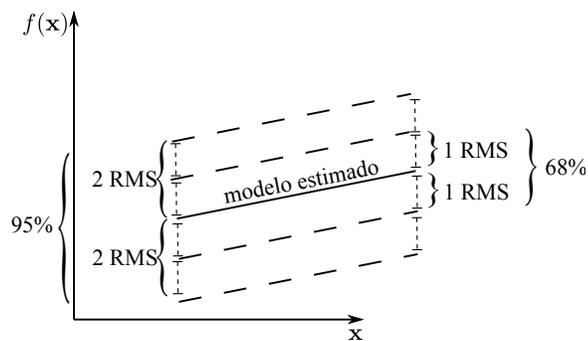


Figura 2.19: Error cuadrático medio.

Incertidumbre

La forma de cuantificar la calidad de una estimación es mediante la incertidumbre del modelo. Supóngase una función $y = f(\mathbf{x})$ donde el vector de parámetros ha sido estimado por algún método de regresión o estimación (ver sección 2.2, en la página 10) junto con el conjunto de observaciones $\{y_1, y_2, \dots, y_n\}$. La relación entre la función $f(\mathbf{x})$ y las observaciones $y_i, i \in [1, n]$ puede definirse como la ecuación (2.58) [49]:

$$\mathbf{y} = f(\mathbf{x}) + \mathbf{e} \quad (2.58)$$

donde \mathbf{e} representa el error cometido por el modelo estimado en cada observación y_i .

La incertidumbre o varianza de cada elemento del vector de parámetros $\sigma_{x_i}^2$ está dada por la ecuación:

$$\sigma_{x_i}^2 = \frac{\sigma_0^2}{(n - m) \cdot \text{Cov}_{ii}} \quad (2.59)$$

donde:

- σ_0^2 es la varianza residual $\sigma_0^2 = [\mathbf{y} - f(\mathbf{x})]^T [\mathbf{y} - f(\mathbf{x})]$.
- Cov_{ii} corresponde al elemento i sobre la diagonal de la matriz de covarianza $\text{Cov} = (J^T J)^{-1}$ con J como la matriz jacobiana. Esta matriz es la derivada de f con respecto a cada elemento \mathbf{x}_i del vector de parámetros \mathbf{x} . Cada renglón de la matriz jacobiana corresponde a cada observación.
- n corresponde al número de observaciones con que fue estimado el vector de parámetros \mathbf{x}
- m corresponde al número de elementos en el vector \mathbf{x}

Este método es útil pues permite propagar la varianza de las observaciones hacia los parámetros permitiendo así poder tener una medida de la calidad de la estimación por parámetro.

2.6. Generación de la imagen que vería una cámara móvil

En la generación de escenas tridimensionales, proponer los ángulos de rotación y el vector de traslación para obtener una vista en específico de una escena puede ser complicado o no tan intuitivo. Una forma de facilitar el proceso de generación de escenas o animaciones es utilizar un esquema en donde suponiendo que el modelo se mantiene fijo, sólo se especifique la posición de la cámara y el punto hacia el que la cámara ve. OpenGL proporciona un procedimiento para poder calcular la matriz de rotación R [50] necesaria para poder utilizar el modelo de la cámara oscura de la ecuación (2.22) de tal forma que se obtenga la imagen que generaría una cámara móvil conociendo los vectores \mathbf{e} , \mathbf{l} y \mathbf{u} que definen la posición de la cámara, el punto al que mira y el punto que es considerado como arriba, respectivamente. La matriz de rotación R se define como:

$$R = \begin{bmatrix} \mathbf{s}^T \\ \mathbf{v}^T \\ -\mathbf{f}'^T T \end{bmatrix} \quad (2.60)$$

donde:

$$\mathbf{s} = \frac{\mathbf{f}' \times \mathbf{u}'}{\|\mathbf{f}' \times \mathbf{u}'\|} \quad (2.61)$$

$$\mathbf{v} = \mathbf{s} \times \mathbf{f}' \quad (2.62)$$

$$\mathbf{f}' = \frac{\mathbf{f}}{\|\mathbf{f}\|} \quad (2.63)$$

$$\mathbf{f} = \mathbf{l} - \mathbf{e} \quad (2.64)$$

$$\mathbf{u}' = \frac{\mathbf{u}}{\|\mathbf{u}\|} \quad (2.65)$$

Capítulo 3

Estado del arte

El problema de localización y reconstrucción simultánea, consiste en obtener el modelo espacial de una escena, mientras se utiliza la reconstrucción parcial hasta el momento para poder llevar a cabo el proceso de localización. Los robots hacen uso de las capacidades propias de sus sensores para resolver este problema [51, 52, 53, 54]. En el caso de una cámara digital que recibe la luz y la convierte en bits, la desventaja recae en la imposibilidad de ver a través de los objetos. Si existiese un sensor capaz de atravesar muros y obstáculos para extraer toda la estructura espacial del entorno sin siquiera tener que ser movido de su punto inicial, el problema de LRS estaría resuelto. Sin embargo, hasta que ese tipo de sensores no exista, los robots necesitan explorar su entorno, es decir: recorrer e ir construyendo el mapa mientras lo hacen.

Un aspecto importante a tomar en cuenta en este problema, es que la reconstrucción basada en la medición de los sensores (imágenes en el caso de visión) es propensa a la acumulación de errores. Cuando un robot inicia su recorrido o exploración, cada medición contribuye en la construcción del mapa. Ese mismo mapa es usado para la localización. Si estas mediciones contienen errores, como pueden ser ruido o datos atípicos y éstos no son tratados o tomados en cuenta, la construcción del mapa será, con el paso del tiempo, más y más imprecisa. Esto puede provocar que el robot crea estar ubicado sin en realidad estarlo (véase figura 3.1).

En este capítulo se habla de algunos trabajos que llevan a cabo la reconstrucción de su entorno, así como de los métodos que utilizan para lograrlo: calibración, localización, construcción del mapa y ajustes.

3.1. Calibración

La calibración de la cámara es una etapa necesaria para poder iniciar el procedimiento iterativo de reconstrucción. Los autores de trabajos que realizan reconstrucción [51, 52, 53, 54] eligen utilizar distintos métodos para el proceso de calibración, sin embargo en su mayoría los métodos preferidos utilizan patrones planos bien conocidos para el proceso de calibración [55, 56, 39]. Algunos ejemplos de patrones planos utilizados son: tableros de ajedrez, marcadores de realidad aumentada, hojas cua-

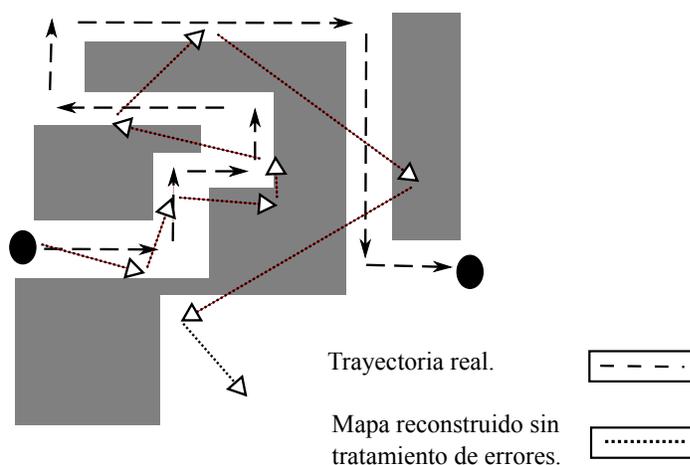


Figura 3.1: Ejemplo de mapa reconstruido con acumulación de errores.

driculadas, elipses, etc. (véase figura 3.2). Sin embargo, esta forma de calibrar una cámara no es la única. También es posible realizar el proceso utilizando objetos tridimensionales como cuboides [57], aunque resulta en cierta forma más sencillo utilizar patrones que se pueden generar utilizando impresoras de papel convencionales.

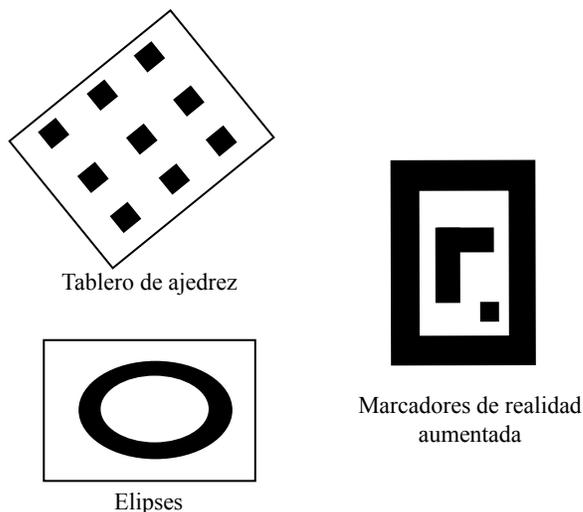


Figura 3.2: Patrones planos comúnmente utilizados para calibración.

Procedimiento de calibración de Zhang

En el 2000, Zhang [39] propuso una metodología para realizar la calibración de una cámara. Este método requiere un patrón plano (tablero de ajedrez, marcadores de realidad aumentada, etc.) para encontrar los parámetros intrínsecos de la cámara. El procedimiento utiliza una solución inicial basada en homografías, la cual es sometida posteriormente a un ajuste no lineal. Este ajuste se encarga no sólo de obtener una

mejor solución al problema de calibración, sino que también permite calcular los coeficientes de distorsión característicos de la lente. Los resultados reportados por Zhang (parámetros intrínsecos de la cámara, coeficientes de distorsión y poses de 5 imágenes), han sido utilizados por diversos trabajos (incluida esta tesis) para realizar comparaciones con los resultados obtenidos (ver sección 5.2 en la página 58).

El procedimiento se muestra en la algoritmo 2.

<p>Entrada: Correspondencias de puntos en las n imágenes y patrón de calibración.</p> <p>Salida: Parámetros intrínsecos y extrínsecos de la cámara y coeficientes de distorsión.</p> <p>Método:</p> <ol style="list-style-type: none"> 1: Obtención de homografías imagen-patrón, para las n imágenes 2: Obtención de valores iniciales de parámetros intrínsecos mediante las homografías del paso 1 3: Obtención de valores iniciales de parámetros extrínsecos mediante las homografías del paso 1 4: Ajuste no lineal por Levenberg-Marquardt para obtener la solución final junto con los coeficientes de distorsión de la lente.

Algoritmo 2: Procedimiento de calibración de Zhang.

Procedimiento de auto-calibración de de la Fraga

Más recientemente (2009) de la Fraga [18] propuso un procedimiento que permite calibrar una cámara a partir de un plano sobre el eje x - y (véase figura 3.3). Las orientaciones de las cámaras son estimadas con respecto a $\theta_1^1 = 0$ y $[u_0, v_0]$ es establecido en el centro de la imagen como punto principal.

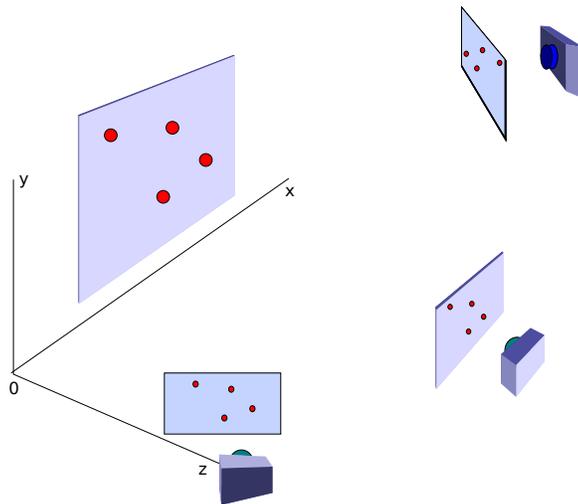


Figura 3.3: Puntos sobre un plano con $z = 0$.

Al estar los puntos sobre el plano $x-y$ un punto en el plano \mathbf{P} de forma homogénea se define como:

$$\mathbf{P} = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

donde la coordenada z es igual a cero.

El procedimiento que se sigue para triangular \mathbf{P} es el mismo de la sección 2.3.3 en la página 22 considerando $z = 0$. Una vez mencionadas las consideraciones para la triangulación el procedimiento para la auto-calibración es el mostrado en la algoritmo 3.

Entrada: Correspondencias de puntos en las n imágenes.

Salida: Parámetros intrínsecos y extrínsecos de la cámara y coeficientes de distorsión.

Método:

- 1: A partir de 3 imágenes se obtiene una estimación del plano y los parámetros de las cámaras usando evolución diferencial.
- 2: Se agregan más vistas y se estiman las poses asociadas a las nuevas vistas, fijando la distancia focal f y el plano reconstruido que se obtuvo en el paso 1 usando evolución diferencial.
- 3: La estimación de las *poses* de las 3 cámaras del paso 1, el plano reconstruido y las estimaciones del paso 2 son usadas como solución inicial para un ajuste por mínimos cuadrados usando el algoritmo de Levenberg-Marquardt.

Algoritmo 3: Procedimiento de auto-calibración de de la Fraga [18].

Por ser éste el procedimiento de auto-calibración en el que se basa esta tesis, a continuación se explican más a detalle cada uno de los pasos del algoritmo:

1. Auto-calibración y estimación del plano:

En este paso se calibran 3 cámaras a partir de *correspondencias* en sus respectivas vistas. En este primer paso no se tiene información sobre los *parámetros internos* y *externos* de las cámaras, ni información sobre el plano a reconstruir. Por tanto, por medio de la evolución diferencial se resuelven simultáneamente ambos aspectos. Así los parámetros a estimar son 18, que formarán el cromosoma para los individuos en la evolución diferencial.

- Parámetros a estimar:

$$\mathbf{w}_1 = [f, \theta_1^2, \theta_1^3, \theta_2^1, \theta_2^2, \theta_2^3, \theta_3^1, \theta_3^2, \theta_3^3, t_1^1, t_1^2, t_1^3, t_2^1, t_2^2, t_2^3, t_3^1, t_3^2, t_3^3]^T$$

Evaluación de un individuo

La evolución diferencial necesita una forma de evaluar la aptitud de cada individuo. A continuación se explica el procedimiento para llevar a cabo dicha evaluación.

Teniendo una solución al problema (3.2) podemos calcular el modelo (2D o 3D) usando la ecuación (2.22): Se conocen:

- Puntos en las imágenes \mathbf{p} .
- Matrices de proyección M (una por cada imagen)

El primer paso es hacer una estimación de los puntos $\mathbf{P} = [x, y, z, 1]^T$. A partir del mismo punto visto en dos (o más) imágenes podemos obtener un sistema sobre determinado para obtener \mathbf{P} como se explica en la sección 2.3.3.

El siguiente paso es reproyectar \mathbf{P} para así obtener $\hat{\mathbf{p}}$.

El objetivo es minimizar la función:

$$g_1 : Q \subset \mathbb{R}^{18} \rightarrow \mathbb{R}$$

$$g_1 = \sum_{i=1}^n \sum_{j=1}^m \|\mathbf{p}_{ij} - \hat{\mathbf{p}}_{ij}(\mathbf{w}_1)\| \quad (3.1)$$

donde g_1 es el error de reproyección, Q es el dominio definido por los límites de búsqueda para cada variable, j es el número de puntos sobre el plano y el operador $\|\cdot\|$ indica la norma euclídea.

Con la función de evaluación definida, la configuración para la evolución diferencial es:

- Límites del espacio de búsqueda:
 - $-90^\circ \leq \theta_1, \theta_2 \leq 90^\circ$
 - $-180^\circ \leq \theta_3 \leq 180^\circ$
 - $-50 \leq t_1, t_2 \leq 50$
 - $100 \leq t_3 \leq 1000$
 - $100 \leq f \leq 2000$
- Tamaño de población= 50 individuos
- Constante diferencial= 0.7
- Probabilidad de recombinación= 0.9

2. Agregar nuevas vistas

Cuando se tiene una estimación del plano en la escena y se desea conocer la pose como fue tomada una imagen (véase figura 3.4), el problema a resolver es similar al problema en el paso 1. Sin embargo, éste es un problema más fácil, pues algunos valores ya se conocen del paso 1:

- Parámetros de la cámara.
- Los puntos de la reconstrucción \mathbf{P} .

Los valores que corresponden a la pose de la imagen son:

$$\mathbf{w}_2 = [\theta_1, \theta_2, \theta_3, t_1, t_2, t_3]^T$$

que representa la orientación y traslación para una nueva imagen.

Estos valores son desconocidos y se obtienen a través de la evolución diferencial de tal forma que se minimice la función:

$$g_2 : Q_2 \subset \mathbb{R}^6 \rightarrow \mathbb{R},$$

$$g_2 = \sum_{i=1}^n \sum_{j=1}^m \|\mathbf{p}_{ij} - \hat{\mathbf{p}}_{ij}(\mathbf{w}_2)\| \quad (3.2)$$

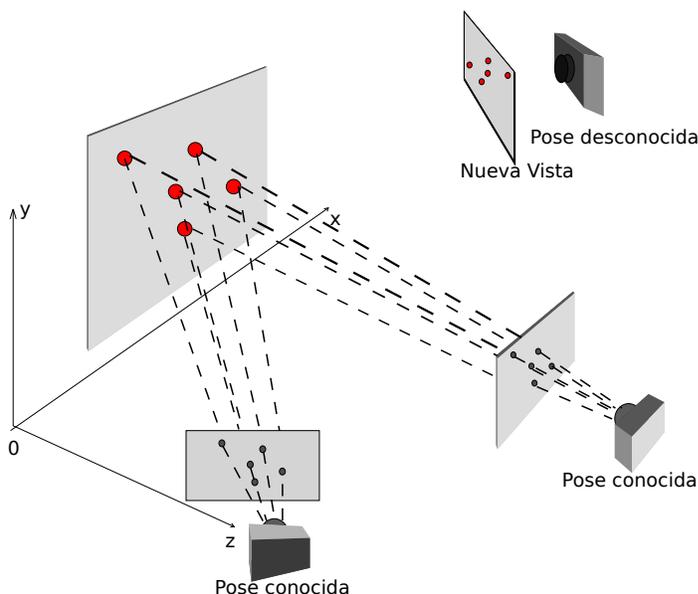


Figura 3.4: Agregando una nueva vista.

3. Ajuste de resultados:

Después de haber agregado varias vistas usando el procedimiento del paso 2, se requiere hacer un ajuste general que corrija simultáneamente las *poses* asociadas a las vistas así como los puntos sobre el plano, para obtener así una mejor aproximación tanto del plano como de las *poses*.

Para llevar a cabo este ajuste se utiliza el algoritmo Levenberg-Marquardt [58]. El error de reproyección de las n imágenes con los m puntos de la reconstrucción.

En la tabla 3.1 se muestran algunos trabajos que utilizan visión monocular para dar solución al problema LRS, junto con un método que utilizan para realizar la calibración de la cámara.

Nombre	Método de calibración usado
Monocular Vision for Mobile Robot Localization and Autonomous Navigation [6]	Lavest et al [56].
Dense 3d mapping with monocular vision [59]	Zhang [39, 55]
A Method for Interactive 3D Reconstruction of Piecewise Planar Objects from Single Images [60]	Propone su propio método
Towards Urban 3D Reconstruction from Video [61]	Ilie y Welch [62]

Tabla 3.1: Trabajos y métodos usados para calibración.

3.2. Localización

El proceso de localización de una cámara consiste en encontrar la pose para una imagen es decir la posición y orientación relativa entre la cámara y la escena como se generó la imagen. Esta tarea se lleva a cabo a partir de las posiciones de los puntos en escena proyectados en la imagen. En otras palabras, localizar la pose para una imagen consiste en saber donde y con que orientación estaba la cámara o el objeto al momento de tomar la fotografía. Este problema puede ser abordado desde diferentes enfoques, los cuales van desde utilizar información espacial y estructural de la escena (modelo 3D o mapa) [63, 64, 65]. También se puede mediante procedimientos que permitan la construcción incremental de un mapa a la vez que se utiliza este mapa para encontrar la pose de más imágenes. En este enfoque las nuevas imágenes con pose encontrada proporcionan información para triangular nuevos puntos que son utilizados para actualizar el mapa [66, 67, 68].

Cuando se se utilizan puntos del mapa o del modelo de la escena, es decir, cuando se desea encontrar la pose entre el modelo y la cámara, el problema de localización de la cámara es conocido como el problema de *perspectiva de n puntos* (PnP) [69, 70, 71, 72]. Sin embargo, cuando se desea encontrar la pose relativa entre dos cámaras en lugar de la pose entre la cámara y el modelo el problema es conocido como el *problema de pose relativa* [73]. Algunos de estos métodos obtienen varias soluciones dependiendo del número de puntos que se utilicen, sin embargo, es común que se proporcionen restricciones adicionales para poder descartar aquellas soluciones que no son válidas. Un ejemplo de estas restricciones puede ser que se debe de elegir la solución que haga que el mayor número de puntos esté al frente de la cámara.

El proceso de localización puede llevarse a cabo tanto en exteriores [74, 75, 76] como en interiores [77, 78, 79]. El procedimiento en ambos tipos de entornos es similar. Sin embargo, los exteriores introducen una serie de problemas adicionales, como son: los cambios de iluminación, cambios de color, sombras, etc. que influyen directamente

en la adquisición de información de los sensores. Estos problemas convierten a la localización en exteriores en un reto mayor a la localización en interiores.

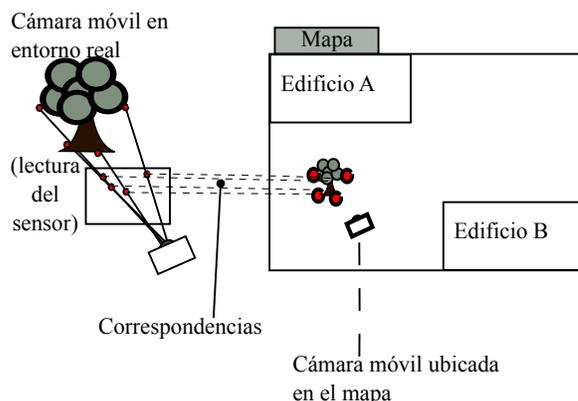


Figura 3.5: Localización de una cámara móvil.

En general el proceso de localización consiste de cuatro pasos [80]: adquisición de datos del sensor, detección de características, detección de correspondencias y cálculo de la posición. La adquisición de datos, se refiere a la lectura de algún sensor: sonar, láser, táctil, imágenes de una cámara, etc. Una vez adquiridos los datos en crudo de los sensores, son procesados para obtener sólo aquellas partes o segmentos de información que son relevantes, necesarias y suficientes para la reconstrucción. En la etapa siguiente (detección de correspondencias) se pretende localizar una misma característica tanto en la lectura del sensor, como en el mapa actual. Varias de estas correspondencias permiten calcular la posición y orientación de la cámara. Este proceso es ilustrado en la figura 3.5.

En este trabajo de tesis se omite el procesamiento de características, por lo que no se profundiza en los trabajos relacionados con este tema. Cuando el problema de obtención de características y correspondencias está parcial o totalmente resuelto, la etapa central de la localización, es decir, la obtención de la pose es solucionada de diferentes formas (véase la tabla 3.2). Los enfoques para abordar esta etapa van desde la utilización de un conjunto de artefactos geométricos que permiten obtener la pose de forma iterativa o bien mediante la solución a sistemas de ecuaciones hasta utilizar la información odométrica de un robot. En el trabajo de Schweighofer y Pinz [70] se presenta una comparación entre distintos métodos tanto iterativos como no iterativos así mismo Grest et al [90] presentan una comparación exclusivamente entre métodos iterativos.

En la tabla 3.3 se muestran algunos trabajos que utilizan visión monocular para dar solución al problema de localización.

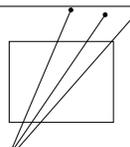
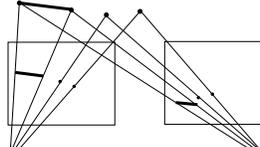
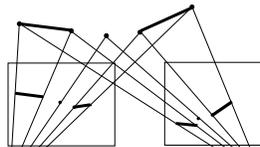
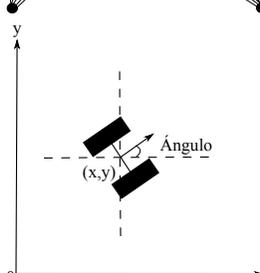
Información usada	Ilustración	Trabajos relacionados
1 imagen, n puntos		[70, 69, 70, 81, 82]
2 imágenes, 2 puntos 3D y 1 línea		[83]
2 imágenes, 2 líneas y 1 punto 3D		[83]
2 imágenes, 5 puntos 3D		[84, 85, 86, 87, 88]
Odometría		[89]

Tabla 3.2: Soluciones al problema de localización.

Nombre	Método usado
Visión Monocular para Localización y Navegación Autónoma de robots [6]	Grunert [82] y Nister [87]
Construcción de mapas densos con visión monocular [59]	Odometría

Tabla 3.3: Trabajos y métodos usados para localización.

3.3. Construcción y actualización del mapa

Con una cámara calibrada y un método elegido para la localización, la reconstrucción del mapa se convierte en un ciclo, el cual consiste de la localización (obtener pose) de una imagen, seguido de la actualización del mapa (véase figura 3.6)

Es este ciclo el que permite la reconstrucción incremental del entorno ya que

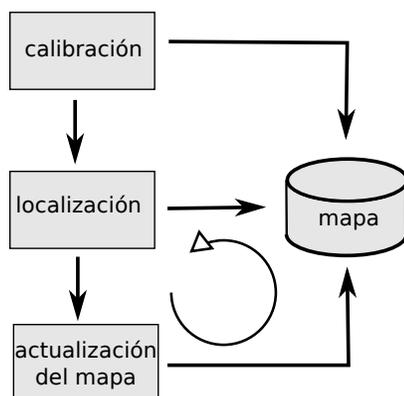


Figura 3.6: Ciclo de reconstrucción.

permite aprovechar la información existente (reconstrucción parcial) para encontrar nueva información (poses de nuevas imágenes) que permite actualizar el modelo de la escena.

En el trabajo de Royer et al. [6] se utilizan dos métodos para encontrar la pose de las imágenes, estos métodos son: el propuesto por Nister [87] y el de Grunert [82]. La forma en que Royer et al. utilizan estos métodos se describe en el algoritmo 4.

Entrada: Secuencia de imágenes, sus correspondencias y cámara calibrada,

Salida: Mapa reconstruido y poses asociadas a las imágenes.

Método:

- 1: Establecer la imagen 1 en el origen y sin rotaciones.
- 2: Obtener la pose de la imagen 1 y 3. Triangular usando 5 correspondencias de puntos y el método de [87].
- 3: Obtener la pose de la segunda imagen usando 3 de los 5 puntos usados en el paso 2.
- 4: Agregar las imágenes $i = 4, 5, 6, \dots, n$ usando 3 puntos del mapa reconstruido y la imagen $i - 1$ utilizando el método Grunert [82]. Triangular nuevos puntos.

Algoritmo 4: Procedimiento de construcción y actualización del mapa en el trabajo de Royer et al. [6].

Debido a la imprecisión inevitable que acompaña a los sensores, la construcción de un mapa en la vida real suele tener errores. Por ello es necesaria una última etapa de ajuste. En la sección 3.3.1 se describe la forma en que algunos trabajos lidian con los errores.

3.3.1. Errores y ajustes

El proceso de reconstrucción incremental viene acompañado de una acumulación de errores producto de los errores que cada imagen procesada puede tener y que van

contribuyendo al error total de la reconstrucción. El error total de la reconstrucción es comúnmente reducido mediante ajustes. Estos ajustes refinan tanto el modelo como las poses de las imágenes para reducir dicho error. En esta sección se describe la forma en que los trabajos mencionados en la tabla 3.3 llevan a cabo esta tarea.

Como ya se mencionó en la tabla 3.3, Wnuk et al. [59] obtienen la información de las poses asociadas a las imágenes a partir de la odometría propia del robot. Cabe mencionar que el uso exclusivo de odometría trae consigo errores. Por otro lado, cuando se trabaja con imágenes de la vida real, es inevitable obtener errores de medición. Lo anterior hace necesario un procedimiento que permita reducir o eliminar los errores que pudieran existir en el mapa construido.

En el trabajo de Wnuk et al. [59] la principal fuente de error está dada por la obtención de las poses a través de la odometría. Sin embargo, para lidiar con los posibles errores, se utiliza la triangulación de puntos utilizando la información de todas las imágenes disponibles para triangular cada punto.

En el trabajo de Royer et al. [6] se realiza un ajuste no lineal usando el método Levenberg-Marquardt [58]. El objetivo es corregir las poses de las imágenes así como las coordenadas 3D de los puntos que conforman en el modelo reconstruido, la función a minimizar es la suma de errores de reproyección para todos los puntos 3D en todas las imágenes en que son visibles. La descripción del procedimiento de ajuste completo se muestra en la algoritmo 5.

Entrada: Secuencia de imágenes, sus correspondencias y cámara calibrada,

Salida: Mapa reconstruido y poses asociadas a las imágenes ajustadas.

Método:

1: Ajuste Jerárquico

- El procedimiento mencionado no se aplica a toda la secuencia completa
- Se divide la secuencia total en 2 partes recursivamente con dos imágenes de traslape para poder unirlos posteriormente
 - Cada tripleta se trata como una secuencia a parte.
Se aplica:
 - Nister
 - Grunert
 - LM sobre las tres imágenes clave (permite que futuros ajustes converjan).

8: Unión de tripletas

- Se usan las dos últimas imágenes de S^1 y las dos primeras imágenes de S^2
Como las vistas son las mismas:
 - Las poses asociadas deben coincidir:
 - Se aplica rotación y traslación a S^2 tal que las poses de S_N^1 coincidan con S_2^2
 - El paso anterior no asegura que S_{N-1}^1 coincida con S_1^2 por lo que se somete el resultado de la unión a LM

13: Ajuste Jerárquico (continuación)

- Debido a que puede haber pequeños errores durante la unión de tripletas, ésta se somete a un ajuste no lineal.
- La reconstrucción es finalizada con un ajuste de la secuencia total con varios cientos de puntos.

Algoritmo 5: Procedimiento de ajuste en el trabajo Royer et al. [6].

Capítulo 4

Propuesta de solución

En este capítulo se presenta el procedimiento propuesto para llevar a cabo la reconstrucción del mapa para entornos en dos y tres dimensiones. En la sección 4.1 se explica el procedimiento de forma general, el cual está basado en la solución a dos subproblemas: auto-calibración de la cámara y localización de cada imagen. Estos subproblemas son resueltos mediante la evolución diferencial [91]. La forma en que se soluciona cada problema es explicada en las secciones 4.2 y 4.3. Algunos otros aspectos importantes para llevar a cabo el proceso de reconstrucción, como el ajuste de la escala, la detección y eliminación de puntos atípicos y el cálculo de la incertidumbre son explicados en las secciones 4.4, 4.5 y 4.6, respectivamente. En la sección 4.7 se describen los aspectos a considerar para poder usar el procedimiento general en la solución al problema de localización y reconstrucción simultánea visual.

4.1. Descripción general

El procedimiento propuesto (ver figura 4.1) realiza la reconstrucción de un modelo en dos o tres dimensiones, a partir de una secuencia de imágenes. Comienza con la auto-calibración de la cámara a partir de tres imágenes. Además de los parámetros intrínsecos de la cámara se obtiene la posición y orientación para cada una de las tres imágenes, lo que permite obtener también una reconstrucción inicial. Los detalles de esta primera etapa se presentan en la sección 4.2.

Una vez que el problema de auto-calibración ha sido resuelto, las imágenes restantes son procesadas una por una utilizando la reconstrucción inicial. Para cada imagen se obtiene una estimación de la pose (orientación y traslación) con la que fue tomada. Conforme se van procesando las imágenes se triangulan aquellos puntos que son visibles en por lo menos dos imágenes y que aún no han sido agregados a la reconstrucción. Este procedimiento es descrito a detalle en la sección 4.3.

Tanto la etapa de auto-calibración como la de localización son resueltas mediante la evolución diferencial, por lo que es necesario ajustar la escala de la reconstrucción de tal forma que siempre se mantenga dentro de los límites de búsqueda establecidos para la heurística. El ajuste de la escala es explicado en la sección 4.4.

Dado que en los problemas de la vida real suele existir puntos atípicos los cuales pueden llegar a entorpecer considerablemente el proceso de reconstrucción, el procedimiento cuenta con un mecanismo basado en las estadísticas del error de reproyección para lidiar con ellos (ver sección 4.5).

Dado que el procedimiento es iterativo y la presencia de ruido es inevitable y la reconstrucción puede gradualmente perder precisión debido a la acumulación de errores. Para evitar el uso indiscriminado de ajustes no lineales que aseguren una buena calidad de la reconstrucción pero que implican un alto costo computacional, la incertidumbre es monitoreada durante todo el proceso (ver sección 4.6); de tal forma que los ajustes no lineales se llevan a cabo sólo cuando se sobrepasa el umbral de incertidumbre. Una vez que el umbral es sobrepasado se realiza un ajuste por medio del método de Levenberg-Marquadt [58] que consiste en minimizar el error de reproyección total del sistema, cuya expresión se muestra en la ecuación (2.56), en la página 28.

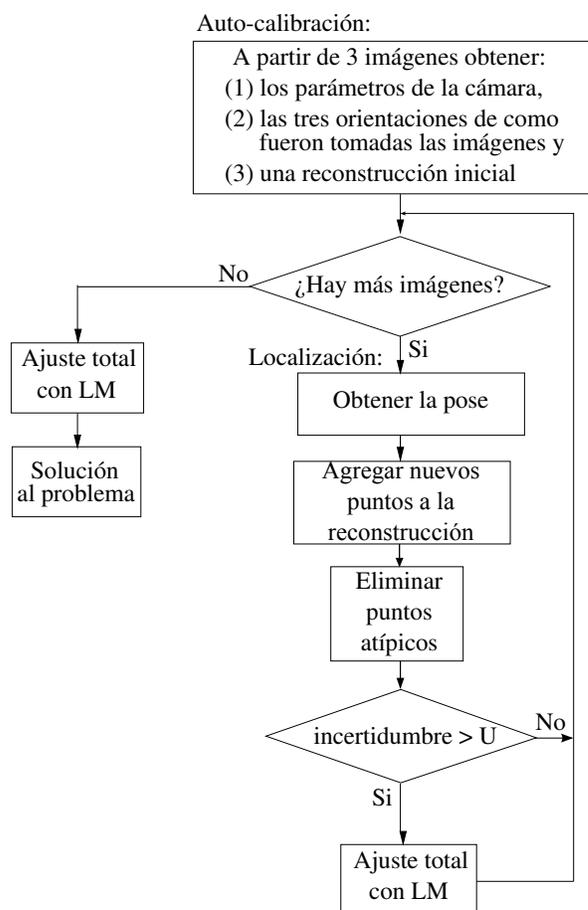


Figura 4.1: Procedimiento general.

4.2. Auto-calibración

El proceso de auto-calibración consiste en obtener los parámetros intrínsecos y extrínsecos de la cámara, además de la estructura de la escena a partir de las características y correspondencias presentes en las imágenes.

En este trabajo se asume que la etapa de extracción de características y sus correspondencias ha sido resuelta.

La estructura de la escena obtenida en esta etapa es usada como solución inicial al problema de reconstrucción.

Se inicia con el modelo básico de la cámara de la ecuación (2.24). El punto principal se sitúa en el centro de las imágenes. El objetivo es utilizar la evolución diferencial [91] como propone de la Fraga en [18] para encontrar el vector solución $\mathbf{w}_1 = \{f, \theta_1^j, \theta_2^j, \theta_3^j, t_1^j, t_2^j, t_3^j\}$ con $j \in \{1, 2, 3\}$ que minimice el error de reproyección total del sistema definido en la ecuación (2.56), en la página 28, para las primeras 3 imágenes. En este trabajo, se propone realizar el proceso de auto-calibración para escenas en dos como en tres dimensiones.

Los límites de búsqueda para la evolución diferencial, así como los parámetros propios de la heurística utilizados en este trabajo se muestran en la tabla 4.1. Estos valores se derivan de los experimentos descritos en el capítulo 6. Es posible que \mathbf{w}_1 esté fuera de los límites de búsqueda. Por esta razón puede ser necesario realizar un ajuste de la escala (ver sección 4.4) de tal forma que la solución deseada se mantenga dentro de los límites de búsqueda.

Una vez que se hayan incorporado más de cinco imágenes, para el caso 2D se puede aplicar el ajuste total con el modelo completo de la cámara obteniendo: f_x, f_y, o, u_0, v_0 además de corregir la distorsión de la lente por medio del método Levenberg-Marquardt.

4.3. Localización

Este proceso consiste en obtener la pose con que fue tomada una imagen mediante la evolución diferencial como propone de la Fraga en [18]. Sin embargo en este trabajo se propone aplicar el mismo procedimiento para escenas 3D.

Dejando fijos la reconstrucción y los parámetros de la cámara para cada imagen se obtiene el vector $\mathbf{w}_2 = \{\theta_1, \theta_2, \theta_3, t_1, t_2, t_3\}$ que minimiza el error de reproyección de todos los puntos de la reconstrucción visibles en la imagen (ver figura 4.2).

Para resolver este problema se utilizan los límites de búsqueda y la configuración de la tabla 4.1 pero con los siguientes cambios:

- $-300 \leq t_1, t_2 \leq 300$
- $10 \leq t_3 \leq 1000$.
- Tamaño de población=30
- $s = 0.1$

Parámetro	Caso 2D	Caso 3D
f	[100, 5000]	[100, 5000]
θ_1	[-180, 180]	[-180, 180]
θ_2	[-90, 90]	[-180, 180]
θ_3	[-180, 180]	[-180, 180]
t_1	[-100, 100]	[-100, 100]
t_2	[-100, 100]	[-100, 100]
t_3	[10, 1000]	[10, 1000]
Tamaño de población	50	50
Constante de la evolución diferencial	0.7	0.7
Probabilidad de recombinación	0.9	0.9
Condición de paros	0.001	0.001

Tabla 4.1: Límites y parámetros para ejecutar la evolución diferencial utilizados para realizar la auto-calibración.

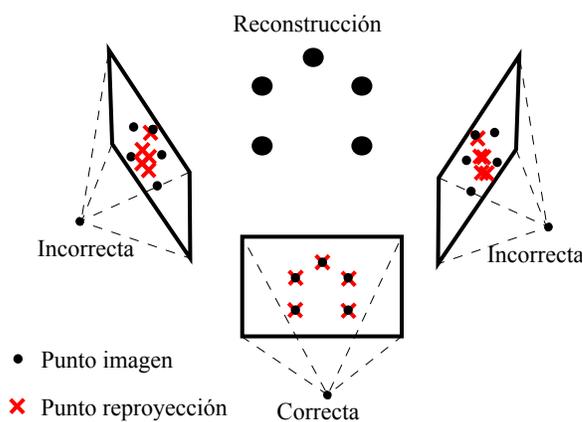


Figura 4.2: Proceso de localización de una imagen.

Cabe mencionar que este problema es más fácil de resolver que el de auto-calibración puesto que se conocen los parámetros intrínsecos de la cámara y la reconstrucción. Al igual que en el caso de la auto-calibración puede ser necesario aplicar un ajuste de la escala (explicado en la sección 4.4) para permitir que la solución deseada esté dentro de los límites de búsqueda para la evolución diferencial.

4.4. Ajuste de la escala

El objetivo del ajuste de la escala es permitir que tanto la reconstrucción como las transformaciones ligadas a la orientación estén dentro de los límites de búsqueda, que se mantienen siempre fijos para la evolución diferencial. Esto se ejemplifica en la figura 4.3. En la subfigura 4.3(a) se muestra un modelo reconstruido de un pentágono donde su pose asociada está fuera de los límites de búsqueda por cuestiones de escala. En la subfigura 4.3(b) se muestra la misma reconstrucción después de aplicar un ajuste de la escala. La pose asociada se encuentra ahora dentro de los límites de búsqueda para la evolución diferencial.

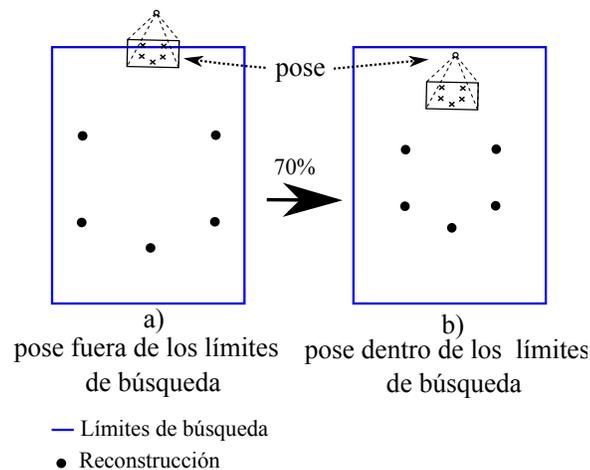


Figura 4.3: Ejemplo de soluciones.

Se considera que una solución está fuera de los límites de búsqueda si en \mathbf{w}_1 ó \mathbf{w}_2 alguno de los elementos del vector de traslación está cerca 3% de los límites establecidos para la evolución diferencial. Si este es el caso, la reconstrucción es reducida a un 70% de su tamaño actual y se busca una nueva solución. El procedimiento se ilustra en la figura 4.4.

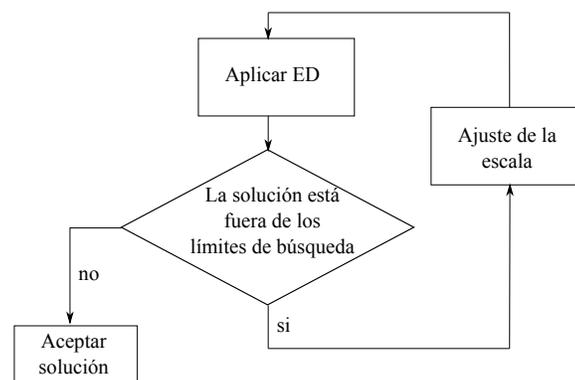


Figura 4.4: Ajuste de la escala.

Para escalar el sistema es necesario multiplicar los vectores de traslación de cada imagen procesada así como las coordenadas de cada punto de la reconstrucción por el factor de reducción, en este caso 0.7.

4.5. Eliminación de puntos atípicos

Este procedimiento consiste en eliminar puntos que pudiesen arruinar la convergencia de la reconstrucción (ver figura 4.5). Consiste en calcular las estadísticas del error de reproyección (ERP) definido en la ecuación (4.1) para cada punto en la reconstrucción.

$$\text{ERP}_i = \frac{\|\sum \mathbf{p}_{ij} - \hat{\mathbf{p}}_{ij}\|}{m} \quad (4.1)$$

donde m es el número de imágenes con pose asignada en las que \mathbf{P}_i es visible y j corresponde al índice de las mismas.

La forma de detectar un punto atípico es la siguiente:

1. Calcular el ERP para cada punto de la reconstrucción.
2. Calcular la media $\overline{\text{ERP}}$ y la desviación estándar σ_{ERP} de los resultados del paso 1.
3. Quitar de la reconstrucción los puntos que tengan un ERP mayor a: $\overline{\text{ERP}} + 2.5 \cdot \sigma_{\text{ERP}}$

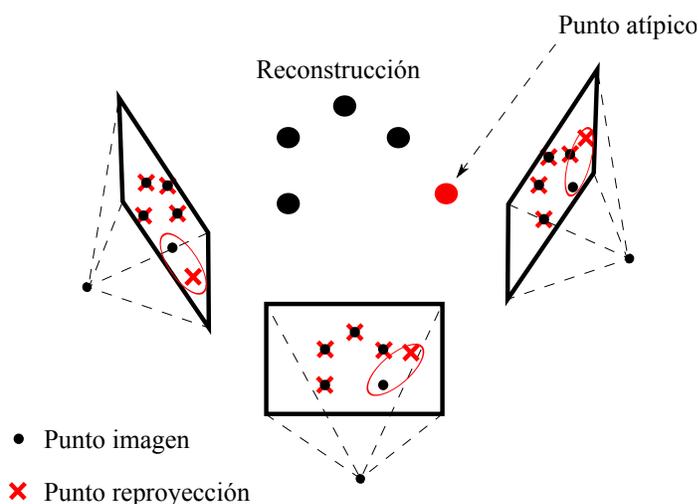


Figura 4.5: Detección de punto atípico.

4.6. Cálculo de la incertidumbre

El objetivo de calcular la incertidumbre del sistema consiste en medir la calidad de la reconstrucción y la información de la pose para las imágenes ya procesadas, esto se logra propagando los errores de reproyección de tal forma que se puede obtener la desviación estándar de cada uno de los elementos que constituyen la pose, así como la de cada una de las coordenadas de los puntos en la reconstrucción.

En este trabajo de tesis utilizamos las ecuaciones de propagación de incertidumbre que Ahn, Rauh y Warnecke utilizaron en [92] en la propagación de la incertidumbre de parámetros de distintas cónicas, sin embargo en nuestro caso las ecuaciones fueron calculadas para obtener la incertidumbre de los parámetros en el modelo de la cámara oscura. Para el caso del cálculo de la incertidumbre de un punto de la reconstrucción se consideró que las poses asociadas a las imágenes eran correctas tal como lo hacen Royer et al. en [6], mientras que en el caso del cálculo de la incertidumbre de una pose se asumió que los puntos de la reconstrucción eran correctos.

Las reproyecciones de cada punto del modelo $\mathbf{P}_i \in \{\mathbf{P}\}$ en una imagen pueden calcularse con la función:

$$\mathbf{F} : \mathbb{R}^6 \rightarrow \mathbb{R}^2$$

$$\mathbf{F}(\mathbf{w}_2, \mathbf{P}_i) = [\hat{u}, \hat{v}]^T = [g_3(\mathbf{w}_2, \mathbf{P}_i), g_4(\mathbf{w}_2, \mathbf{P}_i)]^T \quad (4.2)$$

donde g_3 y g_4 son las funciones base para calcular la incertidumbre a través del siguiente procedimiento

1. Calcular el vector de error \mathbf{e} y σ_0^2 :

$$\mathbf{e} = \begin{bmatrix} u_1 - \hat{u}_1 \\ v_1 - \hat{v}_1 \\ \vdots \\ u_n - \hat{u}_n \\ v_n - \hat{v}_n \end{bmatrix} \quad (4.3)$$

$$\sigma_0^2 = \mathbf{e}^T \cdot \mathbf{e} \quad (4.4)$$

donde σ_0^2 representa la suma de los cuadrados de los errores en el cálculo de la varianza residual.

2. Calcular la matriz jacobiana (detalles en el apéndice A en la página 73):
 - a) Para el caso de necesitar la incertidumbre de un punto en la reconstrucción se calcula como:

$$J_{ij} = \frac{\partial F_i}{\partial P_j}, \quad (4.5)$$

para $j \in \{1, 2, 3\}$ e $i \in \{1, 2, \dots, n\}$, J será de tamaño $2n \times 3$ y n es igual al número de imágenes ya procesadas donde aparece el punto.

- b) Para el caso de calcular la incertidumbre de la orientación y traslación de una imagen:

$$J_{ij} = \frac{\partial F_i}{\partial w_{2j}}, \quad (4.6)$$

para $j = \{1, 2, 3, \dots, 6\}$ e $i \in \{1, 2, \dots, n\}$ por lo que J será de tamaño $2n \times 6$ y n es el número de puntos ya triangulados en la imagen.

3. Calcular la matriz de covarianza:

$$\text{Cov} = (J^T \cdot J)^{-1} \quad (4.7)$$

4. Y finalmente se calcula la incertidumbre:

- a) Para un punto de la reconstrucción:

$$\sigma_{P_j}^2 = \frac{\sigma_0^2}{(2n - 3)} \cdot \text{Cov}_{jj} \text{ para } j \in \{1, 2, 3\} \quad (4.8)$$

- b) Para la orientación y el vector de traslación de una imagen:

$$\sigma_{\mathbf{w}_{2j}}^2 = \frac{\sigma_0^2}{(2n - 6)} \cdot \text{Cov}_{jj} \text{ para } j \in \{1, 2, \dots, 6\} \quad (4.9)$$

Conociendo la incertidumbre de los puntos y de las imágenes se calcula la incertidumbre total del sistema de acuerdo a las ecuaciones (4.10) y (4.11) que corresponden a la desviación estándar promedio para la posición de las cámaras y los puntos para los casos 2D y 3D respectivamente:

$$\frac{\sum_{i=1}^m (\sqrt{\sigma_{x_i}^2} + \sqrt{\sigma_{y_i}^2})}{3n + 2m} + \frac{\sum_{j=1}^n (\sqrt{\sigma_{t_{1j}}^2} + \sqrt{\sigma_{t_{2j}}^2} + \sqrt{\sigma_{t_{3j}}^2})}{3n + 2m} \quad (4.10)$$

$$\frac{\sum_{i=1}^m (\sqrt{\sigma_{x_i}^2} + \sqrt{\sigma_{y_i}^2} + \sqrt{\sigma_{z_i}^2})}{3(n + m)} + \frac{\sum_{j=1}^n (\sqrt{\sigma_{t_{1j}}^2} + \sqrt{\sigma_{t_{2j}}^2} + \sqrt{\sigma_{t_{3j}}^2})}{3(n + m)} \quad (4.11)$$

donde m el número de puntos de la reconstrucción y n es el número de imágenes procesadas.

Como se menciona en la sección 4.4 en la página 49, los puntos de la reconstrucción y los vectores de traslación de cada imagen se encuentran a una escala desconocida con respecto al mundo real. Este factor de escala también afecta la incertidumbre del sistema (ver figura 4.6). Un píxel de incertidumbre en la imagen implica diferentes magnitudes de incertidumbre en la reconstrucción dependiendo de la escala. Por esta razón es importante recalcar que el indicador de incertidumbre debe ser siempre

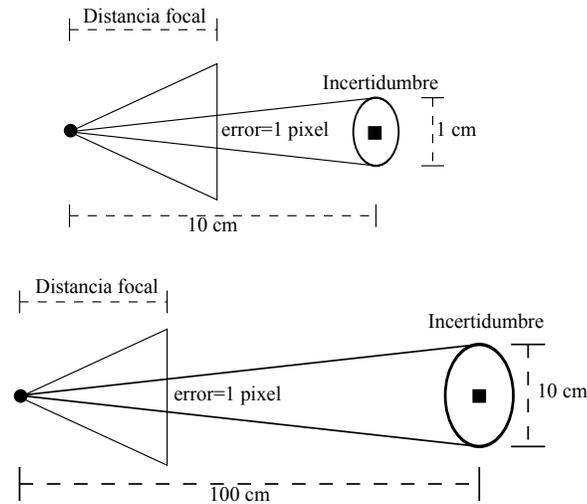


Figura 4.6: Incertidumbre y la escala.

calculado a un factor de escala fijo. Este factor de escala puede ser establecido arbitrariamente pero si se cuenta con alguna distancia de referencia en el mundo real es posible conocer además las posiciones e incertidumbres de los elementos del sistema en unidades reales (milímetros, pulgadas, etc.). Para establecer la escala con una distancia de referencia en el mundo real basta con escalar el sistema con el factor de conversión f_c de la ecuación (4.12)

$$f_c = \frac{\text{distancia}_{\text{real}}}{\text{distancia}_{\text{virtual}}} \quad (4.12)$$

Algunos ejemplos de distancias de referencia son:

- Distancia entre dos puntos del modelo.
- Distancia entre dos cámaras (puede ser obtenida de la odometría de un robot móvil como se explicará en la sección 4.7).
- Distancia de la cámara a un punto del modelo.

4.7. Consideraciones sobre la cámara móvil

El procedimiento mostrado en la figura 4.1 puede ser usado para resolver el problema de localización y reconstrucción del mapa visual. Para poder utilizarlo, deben tomarse en cuenta algunas consideraciones. El modelo de la cámara de la sección 2.3.1 (en la página 17) considera que la cámara se encuentra en el origen (es el modelo el que se mueve) (ver figura 4.7). En el mundo real generalmente se considera que son los objetos los que se mantienen estáticos mientras la cámara es la que se mueve (ver

figura 4.8). Para poder convertir el movimiento de cámara estática a cámara móvil y obtener la posición de la cámara P_c se aplica la ecuación (4.13):

$$\mathbf{P}_c = R^T \cdot (-\mathbf{t}) \quad (4.13)$$

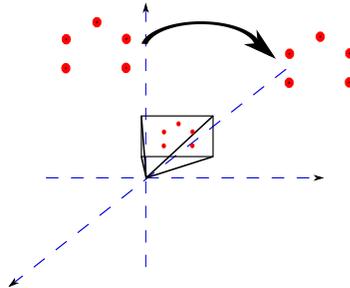


Figura 4.7: Escena con cámara estática.

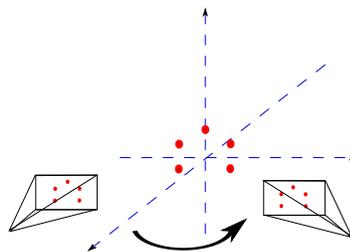


Figura 4.8: Escena con cámara móvil.

Conociendo las posiciones en píxeles de las cámara uno y dos \mathbf{P}_{c1} \mathbf{P}_{c2} y las posiciones obtenidas por la odometría del robot en milímetros \mathbf{P}_{r1} y \mathbf{P}_{r2} , se puede usar $|\mathbf{P}_{r1} - \mathbf{P}_{r2}|$ como $\text{distancia}_{\text{real}}$ y $|\mathbf{P}_{c1} - \mathbf{P}_{c2}|$ como $\text{distancia}_{\text{virtual}}$ para calcular el factor de conversión f_c de acuerdo a la ecuación (4.12).

Capítulo 5

Experimentos y resultados

En este capítulo se presentan los experimentos y los resultados llevados a cabo para poder estimar las condiciones y configuraciones para poder resolver con éxito el problema de auto-calibración y el problema de localización. Se realizaron varios experimentos para validar el funcionamiento de nuestra propuesta. Estos experimentos consistieron en: encontrar el tipo de movimiento, el número de puntos y el valor de la condición de paro mínimos para poder resolver con éxito el problema de auto-calibración y localización (sección 5.1), la reconstrucción de escenas en dos y tres dimensiones (secciones 5.2 y 5.3), así como el análisis de los tiempos y número de evaluaciones requeridos por la evolución diferencial para los problemas de auto-calibración y localización (sección 5.4).

5.1. Configuraciones mínimas

Se realizaron varios experimentos con el fin de encontrar el número de puntos y el valor de la condición de paro de la evolución diferencial, así como las configuraciones de movimiento de la cámara para poder llevar a cabo con éxito los procesos de auto-calibración y localización. Para encontrar el número de puntos, así como el valor de la condición de paro de la evolución diferencial, se generaron 4 patrones planos (A, B, C, D) con 40 puntos sobre ellos (ver figura 5.1) dispuestos aleatoriamente. Es importante señalar que en todos los experimentos realizados consideramos una “solución de la evolución diferencial” a ejecutar cinco veces la instancia del problema y reportar como la solución al problema la mejor solución de las cinco calculadas, esto debido a que la evolución diferencial no puede asegurar al 100 % llegar al óptimo global en una sola ejecución.

A partir de los patrones (A, B, C, D) se generaron imágenes de tamaño 640×480 con el modelo básico de la cámara con $f = 1300.0$ y el punto principal situado en el centro de la imagen (320×240). Se generaron 3 posibles configuraciones de posiciones de la cámara (acercamiento, movimiento lateral, rotación y traslación) como se muestra en la figura 5.2. Estas configuraciones de movimiento se mantuvieron fijas y fueron aplicadas a cada patrón para generar las tres imágenes necesarias para

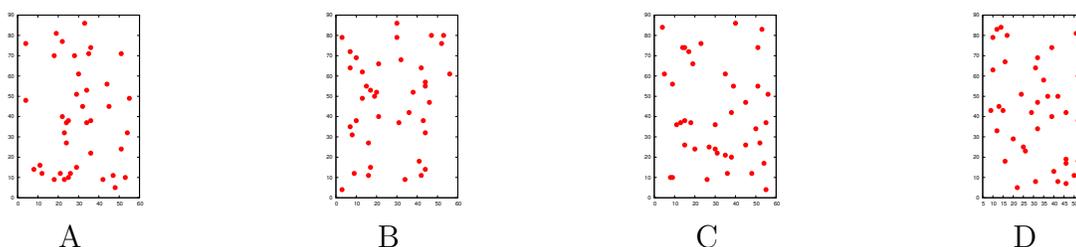


Figura 5.1: Entornos planos utilizados para encontrar configuraciones mínimas.

poder llevar a cabo la etapa de auto-calibración.

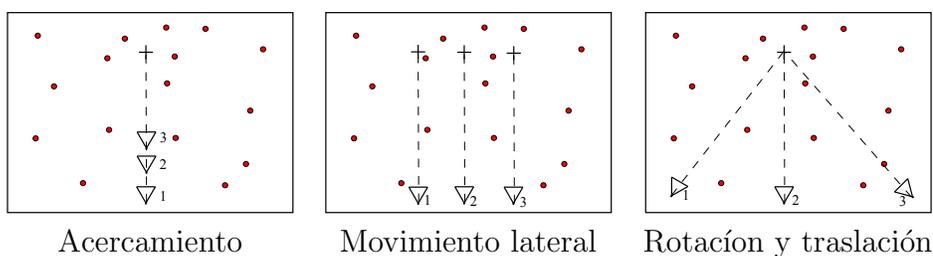


Figura 5.2: Tipos de movimiento.

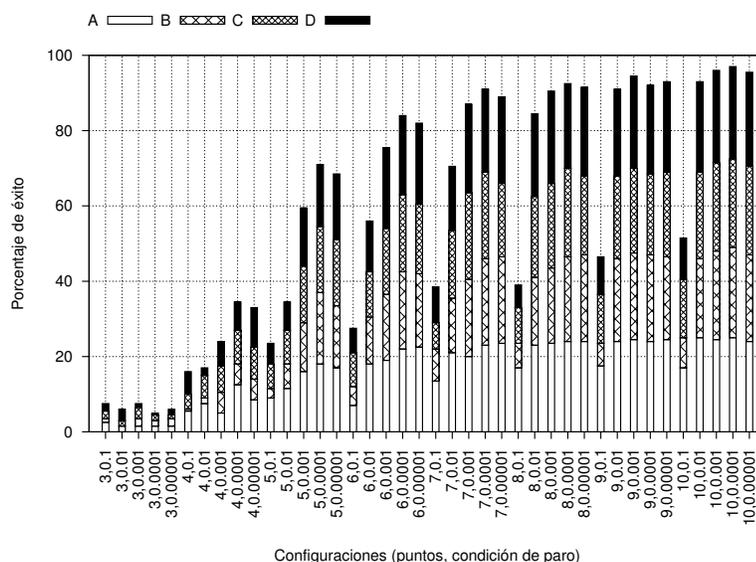


Figura 5.3: Porcentajes de éxito obtenidos con distintas configuraciones para auto-calibración 2D.

El primer experimento tuvo como objetivo encontrar el número de puntos mínimo y el valor de la condición de paro para poder llevar a cabo con éxito las etapas de auto-calibración y localización 2D. Para ambas etapas se utilizaron sólo las imágenes con movimiento de rotación y traslación por ser el tipo de movimiento con mayor

diferencia en las posiciones de las tres imágenes que son necesarias para la etapa de auto-calibración. En el caso de la localización el procesamiento de cada imagen es independiente y por tanto el tipo de movimiento entre cada imagen no afecta el resultado, sin embargo por simplicidad se utilizaron las mismas imágenes. Para los experimentos de localización se utilizaron las posiciones de los puntos en cada imagen, el modelo original y los parámetros intrínsecos de la cámara con que se generaron las tres imágenes. En los experimentos de auto-calibración se utilizaron como entrada sólo las posiciones de los puntos en tres imágenes. Para ambos casos el número de puntos con que se probó fue $n \in \{3, 4, \dots, 10\}$ mientras que la condición de paro de la evolución diferencial fue probada para los valores: 0.1, 0.01, 0.001, 0.0001 y 0.00001. Este conjunto de valores fue elegido por permitirnos probar con condiciones de paro menores y mayores al valor 0.001, que es el recomendado en [93] para encontrar el óptimo global con la evolución diferencial. Cada combinación de número de puntos y condición de paro fue ejecutada 50 veces para cada patrón. Aquellas soluciones obtenidas por la evolución diferencial en las etapas de auto-calibración y localización que diferían en menos del cinco por ciento de la solución real fueron consideradas como exitosas. En el caso de la localización una solución se consideró como exitosa sólo si las poses de las tres imágenes cumplieron con el criterio de diferir en menos del cinco por ciento. Los resultados de este experimento para el caso de auto-calibración y localización se muestran en las figuras 5.3 y 5.4 respectivamente. Analizando las gráficas se obtuvo que el número de puntos mínimo para el caso de de auto-calibración y localización es de 7 y 4, para un porcentaje de éxito arriba del 80 %, respectivamente. Por otro lado la condición de paro de la evolución diferencial necesaria en el caso de auto-calibración es de 0.001 y de 0.1 para localización.

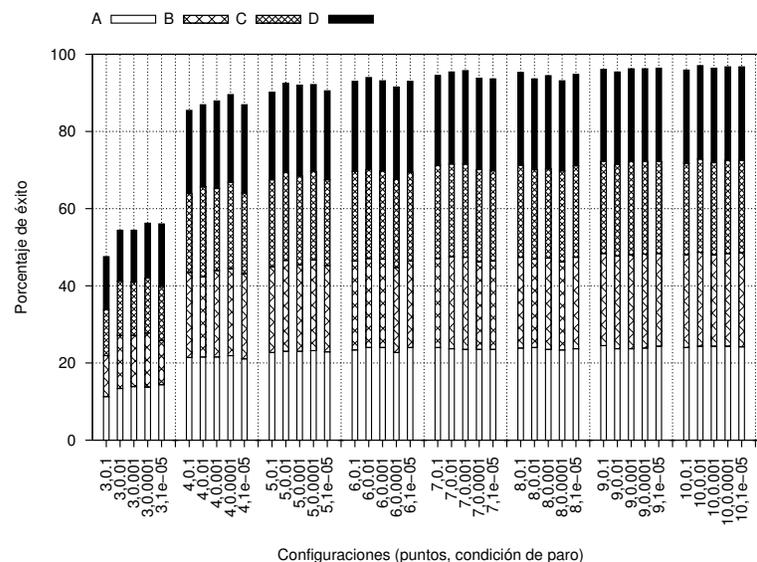


Figura 5.4: Porcentaje de éxito de distintas configuraciones para localización 2D.

El segundo experimento tuvo como objetivo encontrar las configuración de movi-

miento con mayor porcentaje de éxito en el caso de la auto-calibración. Se utilizaron las imágenes de los tres tipos de movimiento generadas para los patrones (A,B,C,D). La prueba se llevó a cabo utilizando 7 puntos para auto-calibrar y la condición de paro de la evolución diferencial de 0.001, que son los valores mínimos resultado del experimento 1. Cada configuración de movimiento fue ejecutada 200 veces, 50 veces para cada patrón. En cada ejecución los 7 puntos fueron elegidos aleatoriamente de entre todos aquellos que fueran visibles en las tres imágenes. Los resultados de este experimento arrojaron un porcentaje de éxito en lograr la auto-calibración de un 88.5 % para el movimiento con rotación y traslación, como se muestra en la figura 5.5. En la gráfica de la figura 5.5 cada indicador muestra el número de soluciones que se pudieron encontrar, con un error menor al 5 %, de cada grupo de imágenes A, B, C y D. Los valores del foco de la cámara además de la orientación y posición en la que se tomó cada imagen fueron tomados como referencia para compararlos con los valores que entrega la evolución diferencial. Si la diferencia de valores entre la solución de la evolución diferencial y la referencia es menor al 5 %, se considera que la evolución diferencial encontró la solución.

De este experimento se concluye que es necesario tomar las imágenes con una cámara con movimiento de rotación y traslación entre las tomas para poder tener una buen porcentaje de éxito en la etapa de auto-calibración.

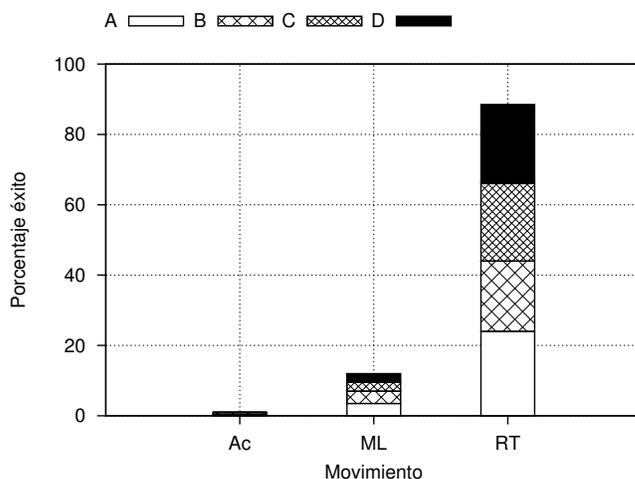


Figura 5.5: Porcentaje de éxito de configuraciones de movimiento para auto-calibración 2D.

5.2. Reconstrucción de entornos planos con datos reales

Para validar el procedimiento propuesto con datos reales se realizó la reconstrucción de un entorno plano usando el conjunto de datos de la rejilla de cuadros utilizado por Zhang en [39] que consiste de 5 imágenes con una resolución de 640×480 píxeles

y que son el resultado de tomar fotografías de un patrón de rejilla de cuadros con 256 puntos de interés (ver figura 5.6). El experimento consistió en aplicar el procedimiento propuesto en la sección 4.1 utilizando distintos subconjuntos de puntos de la rejilla de cuadros con el fin de comparar las soluciones finales y verificar que el procedimiento es consistente y se llega al mismo resultado sin importar los puntos que se seleccionen para la etapa de auto-calibración. Se realizaron 100 ejecuciones con diferente número de puntos desde 3 hasta 10 puntos. En cada ejecución se aplicó el procedimiento propuesto a las 5 imágenes eligiendo una permutación de imágenes al azar de la tabla 5.1. Los puntos tomados para la auto-calibración fueron elegidos también al azar de las primeras tres imágenes de la permutación elegida. El patrón de la rejilla de cuadros es de un tamaño 17 cm. \times 17 cm. y se eligió un umbral de incertidumbre de 1 mm para el proceso de reconstrucción. En la figura 5.7 se muestran los resultados de éxito obtenidos por la evolución diferencial al dar una solución inicial para los diferentes números de puntos utilizados en la etapa de auto-calibración. Fueron tomadas como exitosas las soluciones iniciales que diferían en menos del 5%, 10% y 14% respectivamente de la solución reportada en [39], la elección de estos valores de diferencia es que las soluciones en estos rangos son soluciones muy cercanas y que probablemente podrán ser refinadas por el algoritmo de Levenberg-Marquardt. Todas las soluciones obtenidas son lo suficientemente buenas para que la reconstrucción sea exitosa al aplicar el procedimiento completo de la figura 4.1 en la página 46. La solución final obtenida en las 800 ejecuciones es mostrada y comparada con la reportada en [39] en la tabla 5.2 donde $|\Delta|$ representa el error angular de los vectores formados con la información de las poses de las cinco imágenes.

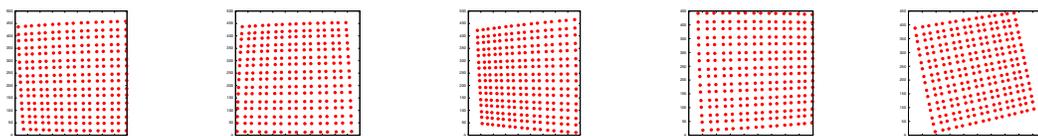


Figura 5.6: Imágenes del conjunto de datos utilizando por Zhang.

5.3. Reconstrucción de entornos tridimensionales

Para validar el funcionamiento de la propuesta en el caso 3D se realizó la reconstrucción de un modelo construido con cuboides así como la del bien conocido dinosaurio de Oxford.

Modelo de cuboides

Se construyó una figura utilizando cuboides a la cual se le tomaron 5 fotografías con un tamaño de imagen de 800×600 píxeles. Los puntos de interés que se consideraron fueron las esquinas formadas por los cuboides. La localización de los puntos

Grupo	Imágenes
1	1, 2, 3, 4, 5
2	1, 2, 4, 3, 5
3	1, 2, 5, 3, 4
4	1, 3, 4, 2, 5
5	1, 3, 5, 2, 4
6	1, 4, 5, 2, 3
7	2, 3, 4, 1, 5
8	2, 3, 5, 1, 4
9	2, 4, 5, 1, 3
10	3, 4, 5, 1, 2

Tabla 5.1: Grupos de imágenes disponibles para reconstrucción de rejilla de cuadros.

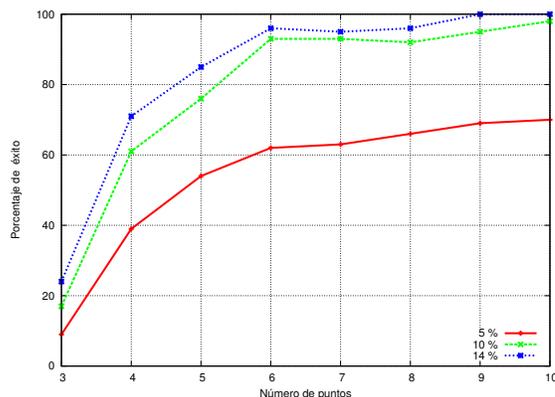


Figura 5.7: Porcentajes de éxito para las soluciones iniciales de reconstrucción para los datos de Zhang.

Métodos	Zhang	Experimento
f_x	832.5	833.501282
f_y	832.53	833.380857
o	0.204494	0.314250
u_0	303.959	312.167771
v_0	206.585	198.484547
k_1	-0.228601	-0.236113
k_2	0.190353	0.170024
$ \Delta $	-	0.45792°
Error RMS	0.335	0.126201

Tabla 5.2: Solución final de los parámetros de la cámara para la rejilla de cuadros.

de interés y la detección de correspondencias se hizo de manera manual, por lo que se asume que los datos de entrada contienen ruido.

Antes de realizar la reconstrucción de la figura la cámara fue calibrada mediante el ampliamente aceptado método de Zhang [39] con el fin de poder comparar resultados. La calibración se llevo a cabo utilizando 3 imágenes de una rejilla de cuadros con 256 puntos de interés como la utilizada por Zhang. Las imágenes usadas en la calibración y los resultados obtenidos con el método de Zhang se presentan en la figura 5.8 y en la tabla 5.3



Figura 5.8: Imágenes usadas para calibrar la cámara con el método de Zhang.

Parámetro	Valor
f_x	982.289003
f_y	976.500102
o	7.059925
u_0	290.200895
v_0	209.447807
k_1	-0.083405
k_2	-0.101362

Tabla 5.3: Parámetros intrínsecos de la cámara usada para la generación de las imágenes de la figura de cuboides.

En la reconstrucción de la figura de cuboides se utilizaron 15 puntos para la etapa de auto-calibración y la localización de las dos imágenes restantes se llevo a cabo utilizando todos los puntos disponibles al momento de localizar la imagen. Para poder establecer la escala (ver sección 4.7 en la página 53) se utilizó la distancia entre la posición de la cámara de la imagen 1 y la imagen 2 de 10 cm que fue medida manualmente al momento de tomar las imágenes. El umbral de incertidumbre utilizado fue elegido arbitrariamente a 1 mm.

Después de aplicar el procedimiento completo a las imágenes de la figura de cuboides se obtuvieron 45 puntos de la reconstrucción. Los parámetros de la cámara, el error RMS y el indicador de incertidumbre del sistema obtenida para la reconstrucción se muestra en la tabla 5.4. Los ángulos de rotación y el vector de traslación en milímetros obtenidos para las 5 imágenes se muestran en la tabla 5.5.

Las fotografías tomadas así como su reproducción a partir de la solución se muestra en la figura 5.6. Imágenes de las vistas superior, frontal y lateral del modelo reconstruido se muestran en la tabla 5.7.

La solución final de la reconstrucción del modelo de cuboides fue utilizada para conocer el número de puntos mínimo necesario en la etapa de auto-calibración en el

f	974.39
o	0.0
u_0	400
v_0	300
Error RMS	1.66
Incertidumbre	0.22 mm

Tabla 5.4: Parámetros de la cámara y error RMS para el modelo construido.

Imagen	1	2	3	4	5
θ_1	0.00	0.99	2.15	-0.08	1.67
θ_2	-0.43	-0.62	-1.78	2.23	1.16
θ_3	1.63	0.87	0.78	-1.71	-0.64
$t1$	62.87	49.65	-55.85	-48.30	37.60
$t2$	4.69	13.98	12.82	-0.33	-12.07
$t3$	186.82	144.95	152.79	210.84	252.02

Tabla 5.5: Ángulos de rotación y vectores de traslación para las imágenes del modelo construido.

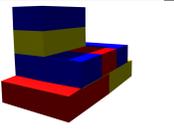
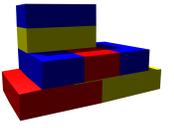
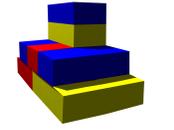
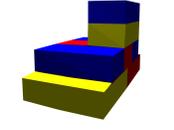
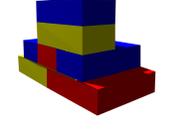
#	Fotografía	Reproducción
1		
2		
3		
4		
5		

Tabla 5.6: Fotografías y su reproducción.

caso 3D. Se ejecuto la etapa de auto-calibración utilizando diferente número de puntos

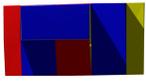
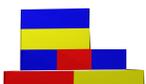
Vista	Imagen
Superior	
Frontal	
Lateral	

Tabla 5.7: Vistas del modelo reconstruido.

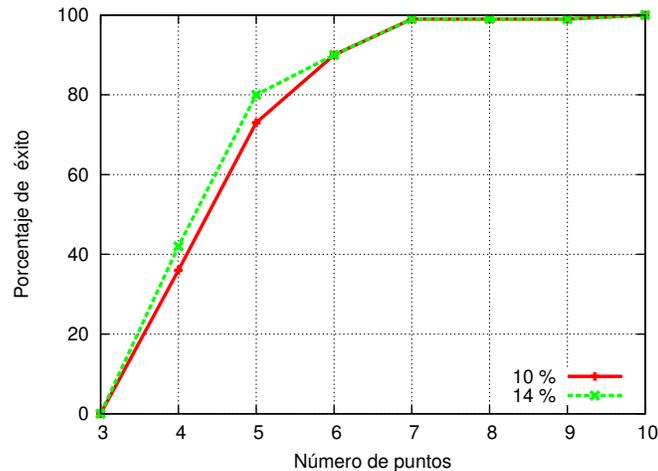


Figura 5.9: Porcentajes de éxito para el número de puntos en la auto-calibración 3D.

que fueron desde tres hasta diez con el fin de compararlos con la solución final. La condición de paro que se eligió fue la misma que para el caso de de auto-calibración 2D de 0.001. Para cada número de puntos se ejecuto la etapa de auto-calibración 100 veces y se tomó como soluciones exitosas a aquellas que obtenidas mediante la evolución diferencial diferían en menos del 10% y 14% de la solución real. Los resultados de este experimento se muestran en la figura 5.9 de donde se puede apreciar que a partir de 6 puntos el problema de auto-calibración 3D tiene un porcentaje del 90%, por lo tanto 6 es el número elegido como mínimo elegido para está etapa.

Del mismo modo que para la auto-calibración 3D, se utilizó la solución final de la figura de cuboides para encontrar el número de puntos mínimo necesario para la etapa de localización 3D. Utilizando como solución inicial la distancia focal y los puntos de la reconstrucción final se ejecutó la etapa de localización 3D utilizando diferente número de puntos, desde tres hasta diez con el fin de comparar la pose

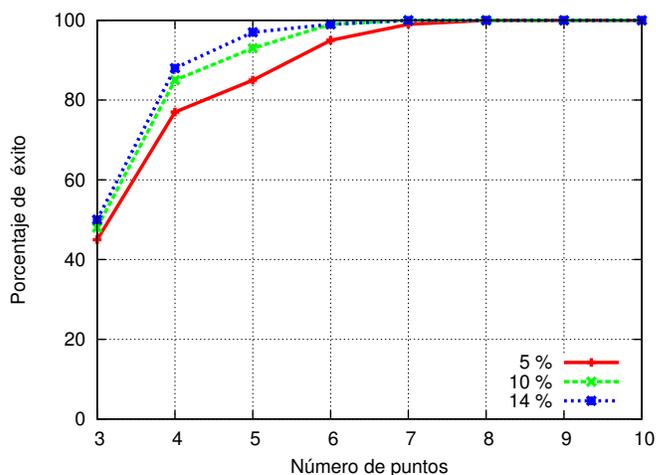


Figura 5.10: Porcentajes de éxito para el número de puntos en la localización 3D.

obtenida por la evolución diferencial con la pose ya conocida. La condición de paro que se eligió fue la misma que para el caso de localización 2D de 0.1. Se ejecutó la etapa de localización 3D 100 veces para cada número de puntos tomando como soluciones exitosas aquellas que diferían en menos del 5 %, 10 % y 14 % de la solución real. Los resultados de este experimento se presentan en la figura 5.10 y muestran que a partir de 6 puntos se obtiene un 95 % de tener soluciones con diferencia menor al 5 % de la solución real. Por lo que 6 es el número elegido para esta etapa.

Dinosaurio de Oxford

Se utilizaron las 36 imágenes del conjunto del dinosaurio de Oxford [94] (ver figura 5.11) que tienen un tamaño de 720×576 . En este conjunto de datos las posiciones de los puntos en cada imagen contienen ruido y algunas de las mediciones son datos atípicos. Se utilizaron 15 puntos para la etapa de auto-calibración y se asumió que la altura del dinosaurio era de 25 cm para establecer la escala. Se utilizó un umbral de incertidumbre de 0.5 mm. Después de aplicar el procedimiento completo se obtuvieron 4633 puntos de la reconstrucción. Los parámetros de la cámara y el error RMS obtenido son mostrados en la tabla 5.8. La figura 5.12 muestra el resultado de la reconstrucción del juguete desde diferentes ángulos.

f	2788.92
o	0.0
u_0	360
v_0	288
Error RMS	0.22
Incertidumbre	0.43 mm

Tabla 5.8: Parámetros de la cámara y error RMS para el modelo construido.

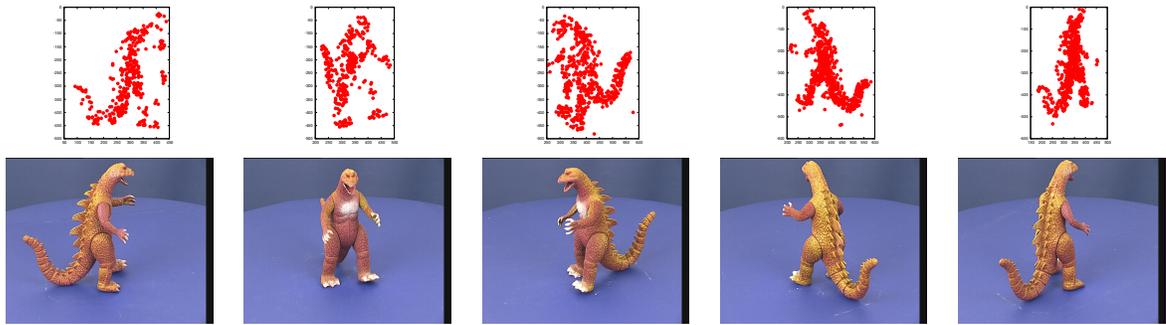


Figura 5.11: Imágenes ejemplo del conjunto de datos del dinosaurio de Oxford (fotografía y puntos de interés).

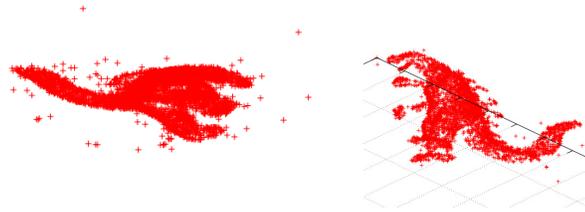


Figura 5.12: Vistas del modelo reconstruido.

5.4. Número de evaluaciones y tiempos de ejecución

Se realizaron pruebas para poder conocer el tiempo y el número de evaluaciones promedio que son necesarios para poder llevar a cabo los procesos de auto-calibración y de localización en el caso 2D y 3D. La prueba consistió en ejecutar varias veces los procedimientos utilizando la condición de paro mínima de 0.001 para el caso de auto-calibración y 0.1 para el caso de localización. Los tiempos reportados en esta sección son el resultado de la ejecución de nuestra implementación para C++ en una computadora con las características de la tabla 5.9. Todas las ejecuciones hechas para el cálculo de tiempos y número de evaluaciones en esta sección fueron calculados para casos libres de ajuste de la escala.

CPU	AMD Athlon X2 1.7 GHz
Memoria RAM	2 GB
Sistema operativo	Ubuntu 10.04 32 bits

Tabla 5.9: Características de la computadora.

Auto-calibración y localización 2D

El conjunto de imágenes de la rejilla de cuadros utilizado en el experimento de la sección 5.2 fue reconstruido 1000 veces. Se utilizaron 7 puntos y la condición de paro de 0.001 para etapa de auto-calibración y 4 puntos con la condición de paro 0.1 para la etapa de localización de una imagen. Estos valores corresponden a las configuraciones mínimas obtenidas en la sección 5.1 para los casos 2D. Debido a que para ambas etapas se hicieron 5 ejecuciones y se eligió la mejor, cada ejecución consiste de:

- 5 instancias de la solución al problema de auto-calibración.
- 5 instancias de la solución al problema de localización.

En total de las 1000 ejecuciones se obtuvieron 5000 instancias del problema de auto-calibración y 5000 instancias del problema de localizar la pose de una imagen. En la tabla 5.11 se muestran la media y la desviación estándar del número de evaluaciones de la función objetivo hechas por la evolución diferencial, así como del tiempo requerido para ambas etapas.

Etapa	Número evaluaciones	Tiempo (segundos)
Auto-calibración (media)	186967.22	1.943600
Auto-calibración (d.e.)	70667.46	0.738913
Localización (media)	6361.15	0.013742
Localización (d.e.)	1186.66	0.005584

Tabla 5.10: Tiempos y número de evaluaciones para las etapas de auto-calibración 2D y localización 2D.

Auto-calibración y localización 3D

Una vez que se conocen el número de puntos mínimo para el caso 3D de las etapas de auto-calibración y localización se realizaron pruebas para medir el número evaluaciones y el tiempo requeridos para las ya mencionadas etapas. El mismo conjunto de imágenes de la figura con cuboides fue reconstruido 1000 veces utilizando 6 puntos y la condición de paro de 0.001 para etapa de auto-calibración y 6 puntos con la condición de paro 0.1 para la etapa de localización, valores que corresponden a las configuraciones mínimas obtenidas para el caso 3D. Por cada ejecución se considero la etapa de auto-calibración y la etapa de localización de una imagen. Al igual que como se hizo las pruebas de tiempo y número de evaluaciones del caso 2D se ejecutaron 5 instancias de cada etapa y se eligió la de mejor de las soluciones entregadas por la evolución diferencial por lo tanto de las 1000 ejecuciones se obtuvieron 5000 instancias del problema de auto-calibración y 5000 instancias del problema de localizar la pose de una imagen. En la tabla 5.11 se muestran la media y la desviación estándar del número de evaluaciones de la función objetivo hechas por la evolución diferencial, así como del tiempo requerido para ambas etapas.

Etapa	Número evaluaciones	Tiempo (segundos)
Auto-calibración (media)	268436.65	24.039040
Auto-calibración (d.e.)	77801.43	7.134147
Localización (media)	6028.5	0.020951
Localización (d.e.)	2857.2	0.0054319

Tabla 5.11: Tiempos y número de evaluaciones para las etapas para el caso 3D de auto-calibración y localización.

Capítulo 6

Conclusiones y trabajo a futuro

El procedimiento propuesto para la reconstrucción de escenas en dos y tres dimensiones consiste en obtener primero una estimación inicial de los parámetros intrínsecos y extrínsecos de la cámara asociados a un conjunto de tres imágenes de entrada. Este problema es no lineal y es resuelto mediante la evolución diferencial. La estimación inicial permite obtener una reconstrucción parcial. La reconstrucción final se va obteniendo de manera iterativa conforme se procesan más imágenes. Durante el proceso iterativo la calidad de la reconstrucción es monitoreada; cuando se sobrepasa un umbral de incertidumbre el modelo es ajustado con el método Levenberg-Marquardt con el fin de calcular el mínimo global de todos los datos y reducir así la acumulación de errores para facilitar así la convergencia de la reconstrucción.

El procedimiento funciona de manera aceptable tanto en la reconstrucción de escenas planas como tridimensionales de la vida real. Se realizaron experimentos para conocer el número de puntos y el valor de la condición de paro para la evolución diferencial necesarios para poder llevar a cabo con éxito la resolución de los problemas de auto-calibración y localización. De los experimentos se obtuvo que la evolución diferencial funciona bien para el caso 2D y 3D con la condición de paro 0.001 para auto-calibrar y 0.1 para el problema de localización. La reconstrucción inicial puede obtenerse a partir de siete puntos en el caso 2D y seis puntos en el caso 3D, mientras que el problema de localización puede resolverse con más de cuatro puntos en el caso 2D y con más de 6 puntos en el caso 3D. Los resultados obtenidos en este trabajo de tesis para el caso de localización concuerdan con los datos por Davies en [69].

En las pruebas del procedimiento con datos de la vida real, se demostró que es posible realizar la reconstrucción de escenas con ruido y puntos atípicos. Las pruebas que se hicieron para para datos reales fueron:

1. Reconstrucción de una rejilla de cuadros a partir de las cinco imágenes de Zhang [39] disponibles en su página web [95]. El experimento consistió en utilizar distintos puntos en la obtención de la solución inicial para comprobar la convergencia del algoritmo. En este experimento se obtuvo una solución con menor error de reproyección que la reportada en [39] sin usar los datos del modelo de donde se generaron los datos. Es por ello que nuestro método también puede clasificarse

como un método de auto-calibración de la cámara [21].

2. Reconstrucción de una figura hecha con cuboides, a partir de cinco imágenes tomadas alrededor de un juguete, con mediciones en presencia de ruido. En la reconstrucción final se obtuvieron los 45 puntos del modelo que conforman la figura.
3. Reconstrucción del dinosaurio de Oxford. Este conjunto de puntos de prueba cuenta con más de 4500 puntos del modelo de un dinosaurio de juguete. La reconstrucción se llevó a cabo a partir de 36 imágenes tomadas alrededor del juguete en donde las mediciones se encuentran en presencia de ruido y puntos atípicos.

En todas las pruebas, el procedimiento funcionó de manera correcta. Sin embargo, después de todas las pruebas hechas, se concluye que para el caso 3D no es posible obtener el modelo completo de la cámara. Aún así el modelo básico de la cámara es suficiente para obtener reconstrucciones de buena calidad, puesto que es una buena aproximación al modelo completo.

Analizando las etapas de auto-calibración y de localización se puede decir que funcionan bien con las configuraciones mínimas derivadas de los experimentos de la sección 5.1 (en la pág. 55). Sin embargo, utilizar un valor de condición de paro menor o más puntos de las configuraciones mínimas, mejora la calidad de las estimaciones, lo que se refleja en una reducción de la frecuencia con que la que se tiene que ajustar la reconstrucción para no sobrepasar el umbral de incertidumbre. En general se recomienda utilizar las configuraciones mínimas de la sección 5.1 pudiendo aumentar ligeramente el número de puntos para mejorar el desempeño.

Trabajo a futuro

El procedimiento propuesto ha mostrado ser capaz de obtener resultados buenos en la reconstrucción de escenas reales. Sin embargo, es posible hacer algunas modificaciones y extensiones que pudiesen agregar capacidades al procedimiento así como permitir su integración en un robot móvil. Estas extensiones y modificaciones se enumeran a continuación:

1. Agregar un módulo que lleve a cabo las etapas de detección de características y correspondencias, las cuales fueron asumidas como resueltas en este trabajo de tesis.
2. Extender el procedimiento para que pueda segmentar el la escena de tal forma que el procedimiento pueda ser empleado en la reconstrucción de escenas de gran escala como calles o ciudades.
3. Integrar el procedimiento en un robot móvil real para que pueda ser utilizado en la toma de decisiones y navegación de entornos desconocidos.

4. Probar el procedimiento con lentes de ángulo amplio que permita tener más información de la escena con menos movimientos de la cámara.
5. Extender el procedimiento para poder recuperarse de lo que en la jerga de robótica se conoce como secuestro (cambio brusco de posición de la cámara móvil que haga al procedimiento perder de vista el segmento de reconstrucción que se tenía hasta el momento).
6. Mejorar la visualización del mapa o de la reconstrucción tridimensional, lo cual incluye extender el procedimiento para permitirle triangular el mapa y poder agregar texturas a la reconstrucción.
7. Estudiar a más detalle el caso de reconstrucción 3D para poder obtener el modelo completo de la cámara así como los coeficientes de distorsión en este tipo de entornos.

Apéndice A

Cálculo de derivadas parciales para el cálculo de la incertidumbre

Como se menciona en la sección 4.6, para poder calcular la incertidumbre de un punto o de los parámetros extrínsecos de una cámara, es necesario calcular las derivadas parciales del modelo de la cámara oscura expresado en forma de función. En esta sección se presentan las derivadas parciales de los coordenadas (u, v) de un punto en la imagen con respecto a $\theta_1, \theta_2, \theta_3, t_1, t_2$ y t_3 necesarios para conocer la incertidumbre de la pose de una cámara o con respecto a x, y y z necesarios en el cálculo de la incertidumbre de un punto tridimensional.

Derivadas parciales para el cálculo de la incertidumbre de la pose de una cámara

$$\begin{aligned}\frac{\partial u}{\partial \theta_1} &= \frac{-f y A - f x B}{E} - \frac{C (-f x (-A) - f y B - \sin \theta_2 f z \cos \theta_3 - f t_1)}{E^2} \\ \frac{\partial u}{\partial \theta_2} &= \frac{I}{E} - \frac{F (-f x (-A) - f y B - \sin \theta_2 f z \cos \theta_3 - f t_1)}{E^2} \\ \frac{\partial u}{\partial \theta_3} &= \frac{-f y G - f x H + \sin \theta_2 f z \sin \theta_3}{E} \\ \frac{\partial u}{\partial t_1} &= -\frac{f}{E} \\ \frac{\partial u}{\partial t_2} &= 0 \\ \frac{\partial u}{\partial t_3} &= \frac{-f x (-A) - f y B - \sin \theta_2 f z \cos \theta_3 - f t_1}{E^2}\end{aligned}$$

$$\begin{aligned} \frac{\partial v}{\partial \theta_1} &= \frac{-f x(-G) - f y H}{E} - \frac{C(-f y(-G) - f x(-H) - \sin \theta_2 f z \sin \theta_3 - f t_2)}{E^2} \\ \frac{\partial v}{\partial \theta_2} &= \frac{J}{E} - \frac{F(-f y(-G) - f x(-H) - \sin \theta_2 f z \sin \theta_3 - f t_2)}{E^2} \\ \frac{\partial v}{\partial \theta_3} &= \frac{-f x(-A) - f y B - \sin \theta_2 f z \cos \theta_3}{E} \\ \frac{\partial v}{\partial t_1} &= 0 \\ \frac{\partial v}{\partial t_2} &= -\frac{f}{E} \\ \frac{\partial v}{\partial t_3} &= \frac{-f y(-G) - f x(-H) - \sin \theta_2 f z \sin \theta_3 - f t_2}{E^2} \end{aligned}$$

donde:

$$\begin{aligned} A &= (\sin \theta_1 \sin \theta_3 - \cos \theta_1 \cos \theta_2 \cos \theta_3) \\ B &= (-\cos \theta_1 \sin \theta_3 - \sin \theta_1 \cos \theta_2 \cos \theta_3) \\ C &= (\cos \theta_1 \sin \theta_2 y + \sin \theta_1 \sin \theta_2 x) \\ D &= (\cos \theta_1 \cos \theta_2 \cos \theta_3 - \sin \theta_1 \sin \theta_3) \\ E &= (\cos \theta_2 z + \sin \theta_1 \sin \theta_2 y - \cos \theta_1 \sin \theta_2 x - t_3) \\ F &= (-\sin \theta_2 z + \sin \theta_1 \cos \theta_2 y - \cos \theta_1 \cos \theta_2 x) \\ G &= (\sin \theta_1 \cos \theta_2 \sin \theta_3 - \cos \theta_1 \cos \theta_3) \\ H &= (-\cos \theta_1 \cos \theta_2 \sin \theta_3 - \sin \theta_1 \cos \theta_3) \\ I &= -\cos \theta_2 f z \cos \theta_3 - \sin \theta_1 \sin \theta_2 f y \cos \theta_3 + \cos \theta_1 \sin \theta_2 f x \cos \theta_3 \\ J &= -\cos \theta_2 f z \sin \theta_3 - \sin \theta_1 \sin \theta_2 f y \sin \theta_3 + \cos \theta_1 \sin \theta_2 f x \sin \theta_3 \end{aligned}$$

Derivadas parciales para un punto

Para simplificar el cálculo de las derivadas parciales implicadas en la incertidumbre de un punto las expresiones se escribieron en términos de los elementos de la matriz de proyección M .

$$\begin{aligned}\frac{\partial u}{\partial x} &= \frac{m_{11}}{O} - \frac{m_{31} P}{O^2} \\ \frac{\partial u}{\partial y} &= \frac{m_{12}}{O} - \frac{m_{32} P}{O^2} \\ \frac{\partial u}{\partial z} &= \frac{m_{13}}{O} - \frac{m_{33} P}{O^2} \\ \frac{\partial v}{\partial x} &= \frac{m_{21}}{O} - \frac{m_{31} Q}{O^2} \\ \frac{\partial v}{\partial y} &= \frac{m_{22}}{O} - \frac{m_{32} Q}{O^2} \\ \frac{\partial v}{\partial z} &= \frac{m_{23}}{O} - \frac{m_{33} Q}{O^2}\end{aligned}$$

donde:

$$O = m_{33} z + m_{32} y + m_{31} x + m_{34}$$

$$P = m_{13} z + m_{12} y + m_{11} x + m_{14}$$

$$Q = m_{23} z + m_{22} y + m_{21} x + m_{24}$$

El cálculo de las derivadas parciales puede llevarse a cabo mediante el uso de herramientas de cálculo simbólico como Maxima [96]. En el algoritmo 6 se presenta el código en Máxima que calcula las derivadas parciales utilizadas en el cálculo de la incertidumbre de la sección 4.6 en la página 51.

```

ca:cos(theta1);
sa:sin(theta);
cb:cos(theta2);
sb:sin(theta2);
cg:cos(theta3);
sg:sin(theta3);
Rza:matrix([ca,-sa,0],[sa,ca,0],[0,0,1]);
Ryb:matrix([cb,0,sb],[0,1,0],[-sb,0,cb]);
Rzg:matrix([cg,-sg,0],[sg,cg,0],[0,0,1]);
R:Rzg.Ryb.Rza;
Q:matrix([R[1,1],R[1,2],R[1,3],t1],
[R[2,1],R[2,2],R[2,3],t2],
[R[3,1],R[3,2],R[3,3],-t3]);
K:matrix([-f,0,0],[0,-f,0],[0,0,1]);
M:K.Q;
lambda:M[3,1]*x+M[3,2]*y+M[3,3]*z+M[3,4];
u:(M[1,1]*x+M[1,2]*y+M[1,3]*z+M[1,4])/lambda;
v:(M[2,1]*x+M[2,2]*y+M[2,3]*z+M[2,4])/lambda;
/*Cálculo de derivadas parciales*/
dua:diff(u,theta1);
dub:diff(u,theta2);
dug:diff(u,theta3);
dut1:diff(u,t1);
dut2:diff(u,t2);
dut3:diff(u,t3);
dva:diff(v,theta1);
dvb:diff(v,theta2);
dvg:diff(v,theta3);
dvt1:diff(v,t1);
dvt2:diff(v,t2);
dvt3:diff(v,t3);
dux:diff(u,x);
duy:diff(u,y);
duz:diff(u,z);
dvx:diff(v,x);
dvy:diff(v,y);
dvz:diff(v,z);

```

Algoritmo 6: Código en Maxima para el cálculo de las derivadas parciales.

Bibliografía

- [1] H.P. Bowditch and W.F. Southard. A comparison of sight and touch. *The Journal of Physiology*, 3(3-4):232–245, January 1882.
- [2] Y. Bannai, M. Ishizawa, H. Shigeno, and K. Okada. A communication model of scents mediated by sense-descriptive adjectives. In *Proceedings of the International Conference on Artificial Reality and Telexistence*, volume 4282 of *Lecture Notes in Computer Science*, pages 1322–1332, Hangzhou, CHN, November 29 – December 1 2006. Springer.
- [3] R. Szeliski. *Computer Vision: Algorithms and Applications*. Texts in Computer Science. Springer, Washington, USA, 1 st edition, 2010.
- [4] S. Thrun, Y. Liu, D. Koller, A.Y. Ng, Z. Ghahramani, and Durrant H.W. Simultaneous localization and mapping with sparse extended information filters. *The International Journal of Robotics Research*, 23(7-8):693–716, July–August 2004.
- [5] M.M. Nevado, J.G. Garcia-Bermejo, and E.Z. Casanova. Obtaining 3d models of indoor environments with a mobile robot by estimating local surface directions. *Robotics and Autonomous Systems*, 48(2-3):131–143, August 2004.
- [6] E. Royer, M. Lhuillier, M. Dhome, and J.M. Lavest. Monocular vision for mobile robot localization and autonomous navigation. *International Journal of Computer Vision*, 74(3):237–260, August 2007.
- [7] M. Tomono. Robust 3d slam with a stereo camera based on an edge-point icp algorithm. In *Proceedings of the International Conference on Robotics and Automation*, pages 4306–4311, Kobe, JPN, May 12-17 2009. IEEE.
- [8] L.S. Davis. A survey of edge detection techniques. *Computer graphics and image processing*, 4(3):248–270, September 1975.
- [9] F. Mokhtarian and F. Mohanna. Performance evaluation of corner detectors using consistency and accuracy measures. *Computer Vision and Image Understanding*, 102(1):81–94, April 2006.
- [10] A. Ericsson and J. Karlsson. Measures for benchmarking of automatic correspondence algorithms. *Journal of Mathematical Imaging and Vision*, 28(3):225–241, July 2007.

- [11] A. Kundu, K.M. Krishna, and C.V. Jawahar. Realtime motion segmentation based multibody visual slam. In *Proceedings of the Indian Conference on Computer Vision, Graphics and Image Processing*, pages 251–258, Chennai, IND, December 12 – 15 2010. ACM.
- [12] J.P. Tardif, Y. Pavlidis, and K. Daniilidis. Monocular visual odometry in urban environments using an omnidirectional camera. In *Proceedings of the International Conference on Intelligent Robots and Systems*, pages 2531–2538, Nice, FRA, Sept 22–26 2008. IEEE.
- [13] A. Eudes, M. Lhuillier, S.N. Collette, and M. Dhome. Fast odometry integration in local bundle adjustment-based visual slam. In *Proceedings of the International Conference on Pattern Recognition*, pages 290–293, Istanbul, TUR, August 23–26 2010. IEEE.
- [14] H. Strasdat, J.M. Montiel, and A.J. Davison. Scale drift-aware large scale monocular slam. In *Proceedings of the Conference on Robotics: Science and Systems*, page 5, Zaragoza, ESP, June 27–30 2010. IEEE.
- [15] H. Strasdat, J.M. Montiel, and A.J. Davison. Real-time monocular SLAM: Why filter? In *Proceedings of the International Conference on Robotics and Automation*, pages 2657–2664, Alaska, USA, May 3–8 2010. IEEE.
- [16] J. Fields, G. Salgian, S. Samarasekera, and R. Kumar. Monocular structure from motion for near to long ranges. In *Proceedings of the International Conference on Computer Vision*, pages 1702–1709, Kyoto, JPN, September 27 – October 4 2009. IEEE.
- [17] D.L. Cardon, W.S. Fife, J.K. Archibald, and D.J. Lee. Fast 3D reconstruction for small autonomous robots. In *Proceedings of the Conference of the IEEE Industrial Electronics Society*, page 6, North Carolina, USA, November 6–10 2005. IEEE.
- [18] L.G. de la Fraga. Self-calibration from planes using differential evolution. In *Proceedings of the Iberoamerican Congress on Pattern Recognition*, volume 5856 of *Lecture Notes in Computer Science*, pages 724–731, Jalisco, MEX, November 15–18 2009. Springer.
- [19] Peter Corke. *Robotics, Vision and Control - Fundamental Algorithms in MATLAB®*, volume 73 of *Springer Tracts in Advanced Robotics*. Springer, 2011.
- [20] John A. Vince. *Rotation Transforms for Computer Graphics*. Springer, 2011.
- [21] R. Hartley and A. Zisserman. *Multiple view geometry*, volume 6. Cambridge university press, New York, NY, 1st edition, 2000.
- [22] A. Björck. *Numerical Methods for Least Squares Problems*. Miscellaneous Bks. Society for Industrial and Applied Mathematics, 1996.

- [23] D.D. Joshi. *Linear Estimation And Design Of Experiments*. John Wiley & Sons, 1987.
- [24] V. Cherkassky and F.M. Mulier. *Learning from Data: Concepts, Theory, and Methods*. The Wiley bicentennial-knowledge for generations. John Wiley & Sons, 2007.
- [25] I. Rechenberg. *Evolutionsstrategie Optimierung technischer systeme nach prinzipien der biologischen evolution*. Friedrich Frommann Verlag, Stuttgart-Bad Cannstatt, 1973.
- [26] Hans-Paul Paul Schwefel. *Evolution and Optimum Seeking: The Sixth Generation*. John Wiley & Sons, Inc., New York, NY, USA, 1993.
- [27] John H. Holland. Outline for a logical theory of adaptive systems. *J. ACM*, 9(3):297–314, 1962.
- [28] David E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1st edition, 1989.
- [29] Heinz Mühlenbein and Dirk Schlierkamp-Voosen. Predictive models for the breeder genetic algorithm – i. continuous parameter optimization. *EVOLUTIONARY COMPUTATION*, 1:25–49, 1993.
- [30] Ralf Salomon. Reevaluating genetic algorithm performance under coordinate rotation of benchmark functions - a survey of some theoretical and practical aspects of genetic algorithms. *BioSystems*, 39:263–278, 1995.
- [31] R. Storn and K. Price. Differential Evolution- A Simple and Efficient Adaptive Scheme for Global Optimization over Continuous Spaces. Technical report, 1995.
- [32] K. Price, R.M. Storn, and J.A. Lampinen. *Differential Evolution: A Practical Approach to Global Optimization*. Natural Computing Series. Springer, 2005.
- [33] Olivier Faugeras. *Three-dimensional computer vision: a geometric viewpoint*. MIT Press, Cambridge, MA, USA, 1993.
- [34] A. Gruen and Thomas S. Huang, editors. *Calibration and Orientation of Cameras in Computer Vision*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2001.
- [35] Janne Heikkilä. Geometric camera calibration using circular control points. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22:1066–1077, 2000.
- [36] Sunil Kumar Kopparapu and Peter I. Corke. The effect of measurement noise on intrinsic camera calibration parameters. In *ICRA*, pages 1281–1286, 1999.

- [37] Manolis I.A. Lourakis and Rachid Deriche, Dr. Camera Self-Calibration Using the Kruppa Equations and the SVD of the Fundamental Matrix: The Case of Varying Intrinsic Parameters. Rapport de recherche RR-3911, INRIA, 2000.
- [38] Quang-Tuan Luong and Olivier Faugeras. Self-calibration of a stereo rig from unknown camera motions and point correspondences. Rapport de recherche RR-2014, INRIA, 1993.
- [39] Z. Zhang. A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(11):1330–1334, March 2000.
- [40] Wei Sun and R. Cooperstock. An empirical evaluation of factors influencing camera calibration accuracy using three publicly available techniques. *Mach. Vision Appl.*, 17(1):51–67, March 2006.
- [41] Jean-Marc Lavest, Marc Viala, and Michel Dhome. Do we really need an accurate calibration pattern to achieve a reliable camera calibration? In *Proceedings of the 5th European Conference on Computer Vision-Volume I - Volume I*, ECCV '98, pages 158–174, London, UK, UK, 1998. Springer-Verlag.
- [42] Tzay Y. Young, editor. *Handbook of pattern recognition and image processing (vol. 2): computer vision*. Academic Press, Inc., Orlando, FL, USA, 1994.
- [43] Reinhard Klette, Karsten Schlüns, and Andreas Koschan. *Computer vision - three-dimensional data from images*. Springer, 1998.
- [44] Emanuele Trucco and Alessandro Verri. *Introductory Techniques for 3-D Computer Vision*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 1998.
- [45] Gene H. Golub and Charles F. Van Loan. *Matrix computations (3rd ed.)*. Johns Hopkins University Press, Baltimore, MD, USA, 1996.
- [46] William H. Press, Brian P. Flannery, Saul A. Teukolsky, and William T. Vetterling. *Numerical Recipes in C: The Art of Scientific Computing, Second Edition*. Cambridge University Press, 2 edition, October 1992.
- [47] Václav Hlaváč and Radim Sára, editors. *Computer Analysis of Images and Patterns, 6th International Conference, CAIP'95, Prague, Czech Republic, September 6-8, 1995, Proceedings*, volume 970 of *Lecture Notes in Computer Science*. Springer, 1995.
- [48] D. Freedman, A. Coduras, and T. Cuffí. *Estadística*. Textos universitarios. Antoni Bosch, 1993.
- [49] H.W. Sorenson. *Parameter estimation: principles and problems*. Control and systems theory. M. Dekker, 1980.

-
- [50] S. Guha. *Computer Graphics Through OpenGL: From Theory to Experiments*. Chapman & Hall/CRC computer graphics, geometric modeling, and animation series. CRC Press, 2010.
- [51] W.D. Rencken. Concurrent localisation and map building for mobile robots using ultrasonic sensors. In *Proceedings of the International Conference on Intelligent Robots and Systems*, volume 3, pages 2192–2197, Tokyo, JPN, July 26 – 30 1993. IEEE.
- [52] J.A. Castellanos and J.D. Tardos. *Mobile robot localization and map building: A multisensor fusion approach*. Springer, Massachusetts, USA, 1 st edition, 1999.
- [53] M. Deans and M. Herbert. Experimental comparison of techniques for localization and mapping using a bearing-only sensor. In *Proceedings of the International Symposium on Experimental Robotics*, volume 271 of *Lecture Notes in Control and Information Sciences*, pages 395–404, Hawaii, USA, December 11–13 2000. Springer.
- [54] J. Guivant, E. Nebot, and S. Baiker. Autonomous navigation and map building using laser range sensors in outdoor applications. *Journal of robotic systems*, 17(10):565–583, October 2000.
- [55] J. Heikkila and O. Silven. A four-step camera calibration procedure with implicit image correction. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pages 1106–1112, San Juan, PUR, June 17–19 1997. IEEE.
- [56] J.M. Lavest, M. Viala, and M. Dhome. Do we need an accurate calibration pattern to archive a reliable camera calibration? In *Proceedings of the European Conference on Computer Vision*, volume 1406 of *Lecture Notes In Computer Science*, pages 158–174, Freiburg, GER, June 2–6 1998. Springer.
- [57] M. Wilczkowiak, E. Boyer, and P. Sturm. Camera calibration and 3d reconstruction from single images using parallelepipeds. In *Proceedings of the International Conference on Computer Vision*, pages 142–148, Vancouver, CAN, July 7–14 2001. IEEE.
- [58] J. Moré. The levenberg-marquardt algorithm: Implementation and theory. In *Proceedings of the Biennial Conference*, volume 630 of *Lecture Notes in Mathematics*, pages 105–116, Dundee, SCO, June 28–July 1 1978. Springer.
- [59] Wnuk, F. Dang, and Z. Dodds. Dense 3d mapping with monocular vision. In *Proceedings of the International Conference on Autonomous Robots and Agents*, pages 212–217, Palmerston North, NZL, December 13-15 2004. Citeseer.
- [60] Peter F. Sturm and Stephen J. Maybank. A method for interactive 3d reconstruction of piecewise planar objects from single images. In *In Proc. BMVC*, pages 265–274, 1999.

- [61] A. Akbarzadeh, J.-M. Frahm, P. Mordohai, B. Clipp, C. Engels, D. Gallup, P. Merrell, M. Phelps, S. Sinha, B. Talton, L. Wang, Q. Yang, H. Stewenius, R. Yang, G. Welch, H. Towles, D. Nister, and M. Pollefeys. Towards urban 3d reconstruction from video. In *3D Data Processing, Visualization, and Transmission, Third International Symposium on*, pages 1–8, June 2006.
- [62] Adrian Ilie and Greg Welch. Ensuring color consistency across multiple cameras. In *Proceedings of the Tenth IEEE International Conference on Computer Vision - Volume 2, ICCV '05*, pages 1268–1275, Washington, DC, USA, 2005. IEEE Computer Society.
- [63] D. Kim and R. Nevatia. Symbolic navigation with a generic map. *Autonomous Robots*, 6(1):69–88, June 1999.
- [64] J. Pan, D.J. Pack, A. Kosaka, and A.C. Kak. Fuzzy-nav: a vision-based robot navigation architecture using fuzzy inference for uncertainty reasoning. In *Proceedings of the World Congress on Neural Networks*, pages 602–607, Washington, USA, July 17–21 1995. IEEE.
- [65] S. Thrun. Probabilistic algorithms in robotics. *AI Magazine*, 21(4):93, March 2000.
- [66] S. Thrun. Learning metric-topological maps for indoor mobile robot navigation. *Artificial Intelligence*, 99(1):21–71, March 1998.
- [67] Z. Zhang and O. Faugeras. A 3D world model builder with a mobile robot. *The International journal of robotics research*, 11(4):269–285, December 1992.
- [68] Y. Yagi, S. Kawato, and S. Tsuji. Real-time omnidirectional image sensor (CO-PIS) for vision-guided navigation. *IEEE Transactions on Robotics and Automation*, 10(1):11–22, May 1994.
- [69] E.R. Davies. *Machine Vision: Theory, Algorithms, Practicalities*. Signal Processing and Its Applications. Elsevier Science, 2004.
- [70] G. Schweighofer and A. Pinz. Globally optimal $\mathcal{O}(n)$ solution to the pnp problem for general camera models. In *Proceedings of the British Machine Vision Conference*, pages 55.1–55.10. BMVA Press, 2008. doi:10.5244/C.22.55.
- [71] Friedrich Fraundorfer, Petri Tanskanen, and Marc Pollefeys. A minimal case solution to the calibrated relative pose problem for the case of two known orientation angles. In *Proceedings of the 11th European conference on Computer vision: Part IV, ECCV'10*, pages 269–282, Berlin, Heidelberg, 2010. Springer-Verlag.
- [72] Dawei Leng and Weidong Sun. Finding all the solutions of pnp problem. In *Imaging Systems and Techniques, 2009. IST '09. IEEE International Workshop on*, pages 348–352, May 2009.

-
- [73] K. Daniilidis and R. Klette. *Imaging Beyond the Pinhole Camera*. Computational Imaging and Vision. Springer, 2007.
- [74] V. Graefe and K.D. Kuhnert. Vision-based autonomous road vehicles. In Ichiro Masaki, editor, *Proceedings of the International Conference on Industrial Electronics, Control, Instrumentation, and Automation*, pages 1–29, California, USA, November 9–13 1992. Springer.
- [75] D.A. Pomerleau. Efficient training of artificial neural networks for autonomous navigation. *Neural Computation*, 3(1):88–97, July 1991.
- [76] E.D. Dickmanns. Computer vision and highway automation. *Vehicle System Dynamics*, 31(5-6):325–343, March 1999.
- [77] N.J. Nilsson. Shakey the robot. Technical Report 323, Artificial Intelligence Center. Computer Science and Technology Division, DTIC Document, April 1984.
- [78] R. Chatila and J. Laumond. Position referencing and consistent world modeling for mobile robots. In *Proceedings of the International Conference on Robotics and Automation*, pages 138–145, Missouri, USA, March 25–28 1985. IEEE.
- [79] A. Kosaka and A.C. Kak. Fast vision-guided mobile robot navigation using model-based reasoning and prediction of uncertainties. *Graphical Models and Image Processing*, 56(3):271–329, November 1992.
- [80] G.N. DeSouza and A.C. Kak. Vision for mobile robot navigation: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(2):237–267, April 2002.
- [81] M. A. Fischler and R. C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis an automated cartography. *Communications of the ACM*, 24(6):381–395, June 1981.
- [82] J.A. Grunert. Das pothenotische problem in erweiterter gestalt iebst bler seine aiiwendun- gen in der geodasie, grunerts archiv fur math- einatik und physik. *Grunerts Archiv für Mathematik und Physik*, 1(1):238–248, 1841.
- [83] S. Ramalingam, S. Bouaziz, and P. Sturm. Pose estimation using both points and lines for geo-localization. In *Proceedings of the International Conference on Robotics and Automation*, pages 4716–4723, Shanghai, CHN, may 9–13 2011.
- [84] D. Batra, B. Nabbe, and M. Hebert. An alternative formulation for five point relative pose problem. In *Proceedings of the Workshop on Motion and Video Computing*, page 21, Texas, USA, February 23-24 2007. IEEE.
- [85] Z. Kukelova, M. Bujnak, and T. Pajdla. Polynomial eigenvalue solutions to the 5-pt and 6-pt relative pose problems. In *Proceedings of the British Machine Vision Conference*, pages 56.1–56.10, Leeds,UK, September 1–4 2008. BMVA Press.

- [86] H. Li and R. Hartley. Five-point motion estimation made easy. *Proceedings of the International Conference on Pattern Recognition*, 1:630–633, August 20–24 2006.
- [87] D. Nister. An efficient solution to the five-point relative pose problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(6):756–770, June 2004.
- [88] J. Philip. A non-iterative algorithm for determining all essential matrices corresponding to five point pairs. *The Photogrammetric Record*, 15(88):589–599, April 1996.
- [89] D. Nister, O. Naroditsky, and J. Bergen. Visual odometry. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, volume 1, pages I–652, Washington, USA, June 27 – July 2 2004. IEEE.
- [90] Daniel Grest, Thomas Petersen, and Volker Krüger. A comparison of iterative 2d-3d pose estimation methods for real-time applications. In *Proceedings of the 16th Scandinavian Conference on Image Analysis*, SCIA '09, pages 706–715, Berlin, Heidelberg, 2009. Springer-Verlag.
- [91] Rainer Storn and Kenneth Price. Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*, 11:341–359, December 1997. 10.1023/A:1008202821328.
- [92] S.J. Ahn, W. Rauh, and H.J. Warnecke. Least-squares orthogonal distances fitting of circle, sphere, ellipse, hyperbola, and parabola. *Pattern Recognition*, 34(12):2283–2303, January 2001.
- [93] Efrén Mezura-Montes, Mariana Edith Miranda-Varela, and Rubí del Carmen Gómez-Ramón. Differential evolution in constrained numerical optimization: An empirical study. *Inf. Sci.*, 180(22):4223–4262, November 2010.
- [94] Visual geometry group webpage. <http://www.robots.ox.ac.uk/~vgg/data.html>. Accessed: 12/08/2012.
- [95] A flexible new technique for camera calibration webpage. <http://research.microsoft.com/en-us/um/people/zhang/calib/>. Accessed: 12/08/2012.
- [96] Maxima webpage. <http://maxima.sourceforge.net/>. Accessed: 12/08/2012.