

Trabajo Fin de Grado

¿Cómo debemos posicionar la cámara en el proceso de calibración para reducir el error?

How we should position the camera during the calibration process to reduce the error?

Autor
Javier Tirado Garín

Director
Dr. Javier Civera Sancho

Escuela de Ingeniería y Arquitectura, Universidad de Zaragoza
2019

Agradecimientos

Lo primero que quiero hacer es darle las gracias a todas aquellas personas que me han ayudado directa o indirectamente a llegar hasta aquí y poder terminar el Grado.

Especialmente gracias a mi tutor, el Dr. Javier Civera por su ayuda e interés mostrado durante todo el TFG y por motivarme a querer aprender más.

También gracias a mi familia y amigos, quienes me han podido ayudar incluso más de lo que a lo mejor se pueden pensar.

Especialmente gracias a mi madre, por haberme dado un apoyo incondicional y haber estado aguantándome la mayor parte del tiempo mientras realizaba el TFG.

Sin ellos tengo claro que no habría podido.

Resumen

La calibración de una cámara consiste en la caracterización de cómo la realidad 3D se proyecta en el plano imagen 2D, siendo de gran importancia para sistemas de reconstrucción 3D. El procedimiento habitual de calibración consiste en la captura de varias imágenes de un patrón de geometría conocida, desde distintos puntos de vista. La precisión de la calibración está muy relacionada con dichas vistas; y sin embargo, la literatura acerca de la relación entre ambas es escasa y los pocos métodos existentes son secuenciales: A partir de una calibración predicen cuál es la siguiente vista a tomar para minimizar el error. En la literatura no existen resultados teóricos ni análisis acerca del problema general, no secuencial, de cuáles son las mejores vistas para calibrar una cámara.

El objetivo de este TFG es establecer, mediante análisis teórico y con simulación, una forma general de ubicar los patrones de calibración sin conocimiento previo de otras imágenes. Dicho de otra forma, encontrar unas posiciones y orientaciones de cámara que aseguren un error de calibración bajo. Además, se desarrollará una herramienta interactiva para ayudar a la colocación relativa entre patrón y cámara que asegura dicho error bajo.

Para ello se estudia la relación entre la posición y la orientación de la cámara que capturó las imágenes del patrón con el error o incertidumbre en la estimación de los parámetros del modelo. Nuestros resultados experimentales han mostrado mejoras con respecto al estado del arte en simulación e imágenes reales. En concreto, nuestro método tiene errores de calibración menores o comparables al estado del arte, con un coste significativamente menor.

Índice general

1. INTRODUCCIÓN	11
1.1. Motivación	13
1.2. Objetivos	15
1.3. Herramientas utilizadas	15
1.4. Estructura del trabajo	16
2. FORMACIÓN DE LA IMAGEN	17
2.1. Modelo de las lentes	17
2.2. Modelo pin-hole	19
2.3. Modelo de distorsión	24
2.4. Transformación mundo \leftrightarrow cámara	26
3. CALIBRACIÓN DE UNA CÁMARA CON PATRÓN CONOCIDO	29
3.1. Notación	30
3.2. Obtención de la solución cerrada	31
3.2.1. Cálculo de la matriz de homografía	31
3.2.2. Restricciones en los parámetros intrínsecos	33

3.2.3. Solución cerrada final	34
3.3. Refinamiento de la solución cerrada	36
3.3.1. Interpretación probabilista de la matriz hessiana	41
4. SIMULACION	43
4.1. Entorno de trabajo desarrollado	43
4.2. Importancia de la colocación del patrón	48
4.3. Sensibilidad con el número de imágenes	50
4.4. Sensibilidad con la distancia	52
4.4.1. Justificación teórica	56
4.5. Sensibilidad con la orientación	60
4.6. Comparación con el estado del arte	67
5. IMPLEMENTACIÓN EN IMÁGENES REALES	71
6. CONCLUSIONES	81
Anexos	87
A. Creación de las traslaciones	89
B. Imágenes utilizadas en la evaluación experimental	97

Índice de figuras

1.1.	Productos comerciales que necesitan la calibración de cámaras	12
1.2.	Fotos de un patrón de calibración	13
1.3.	Tipos de ayuda en la colocación de los patrones	14
2.1.	Funcionamiento de una lente fina	18
2.2.	Lentes <i>Double-Gauss</i>	19
2.3.	Modelo pin-hole de formación de la imagen.	20
2.4.	Esquema 3D del modelo pin-hole	20
2.5.	Esquema pixélico del modelo pin-hole	22
2.6.	Punto de fuga	23
2.7.	Modelo de distorsión radial	25
2.8.	Vista 3D del patrón de calibración y la cámara	27
3.1.	Ejemplo de detección de esquinas del patrón de calibración	30
3.2.	Error en la detección de las esquinas	37
3.3.	Importancia de la solución cerrada	39
3.4.	Possible función de aproximación local cuadrática	41

4.1.	Vista 3D del proceso de generación de patrones para una única rotación	45
4.2.	Criterio 1 de visibilidad de los patrones	45
4.3.	Criterio 2 de visibilidad de los patrones	46
4.4.	Círculo de equivalencia angular	47
4.5.	Tablero utilizado en la mayoría de los análisis	47
4.6.	Error en la estimación de 600 calibraciones	49
4.7.	Significado geométrico del error relativo del punto principal	50
4.8.	Errores promedio en función del número de imágenes	51
4.9.	Punto de vista geométrico del análisis de distancia	53
4.10.	Errores en 300 calibraciones con distintas distancias	54
4.11.	Valores promedio del error en los 3 tramos de alejamiento evaluados .	55
4.12.	Valores propmedio del error con otros modelos de cámara	56
4.13.	Valores promedio de la incertidumbre en fc y cc	57
4.14.	Ejemplo de 2 patrones situados lo más cerca posible de la cámara. . .	61
4.15.	Incertidumbre en función de θ y el número de imágenes 1	62
4.16.	Incertidumbre en función de θ y el número de imágenes 2	63
4.17.	Definición gráfica de las 2 componentes del ángulo de visión.	64
4.18.	Determinación del valor de θ óptimo	66
4.19.	Comparación con el estado del arte	69
5.1.	Herramienta visual desarrollada de guiado de los patrones de calibración	72
5.2.	Desviación típica en la detección de las esquinas (σ_{Δ_m}) en función de θ . 74	74
5.3.	Determinación del valor de θ óptimo tras incluir el ruido en la detección	75

5.4. Comparación con imágenes reales 1	78
5.5. Comparación con imágenes reales 2	79
B.1. Fotos realizadas con el método propuesto	98
B.2. Primer conjunto de fotos a comparar	98
B.3. Segundo conjunto de fotos a comparar	99
B.4. Tercer conjunto de fotos a comparar	99

Índice de tablas

4.1. Modelo de cámara utilizado en la mayoría de los análisis realizados	48
4.2. Modelos de cámara que se añaden al análisis de influencia de la distancia	52
4.3. Modelos de cámara usados para validar la expresión 4.1.	67
4.4. Comparación con el estado del arte	70
5.1. Primera comparación con imágenes reales. Modelo: <i>webcam logitech C920</i>	73
5.2. Comparación final	76

Capítulo 1

INTRODUCCIÓN

Este TFG se sitúa dentro del campo de la visión por computador, específicamente, en el proceso de calibración de una cámara. La calibración de una cámara consiste en la obtención de los parámetros intrínsecos que definen la proyección de una escena 3D, para lo cual también se deben obtener los parámetros extrínsecos (rotación y traslación) que la permiten situar en un sistema de coordenadas determinado.

Conocer la calibración de una cámara es imprescindible en todas aquellas aplicaciones en las que se necesite extraer información métrica de imágenes (altura, profundidad, distancias o parámetros geométrico en general). Algunos ejemplos de aplicación a destacar los encontramos dentro del SLAM (acrónimo proveniente del inglés *Simultaneous Localization And Mapping*), elemento clave en usos como la conducción autónoma, la robótica y la reconstrucción 3D de una escena.

Las capacidades del SLAM llevadas a la vida cotidiana han permitido desarrollar varios productos comerciales relevantes. Entre ellos, aplicaciones de realidad aumentada (por ejemplo la desarrollada por IKEA¹, mostrada en la Figura 1.1b, y que sirve para visualizar cómo quedaría un mueble que queramos adquirir en el sitio que elijamos), gafas de realidad virtual (como por ejemplo el modelo *Oculus Rift S*² mostrado en la Figura 1.1c) y sistemas de navegación autónoma que permiten, por nombrar un caso concreto, a un dron seguir a una persona evitando los obstáculos del entorno³ (ver Figura 1.1a). En todas estas aplicaciones es necesario conocer

¹<http://highlights.ikea.com/2017/ikea-place/>

²<https://www.oculus.com/rift-s/>

³<https://www.youtube.com/watch?v=gsfkG1SajHQ>

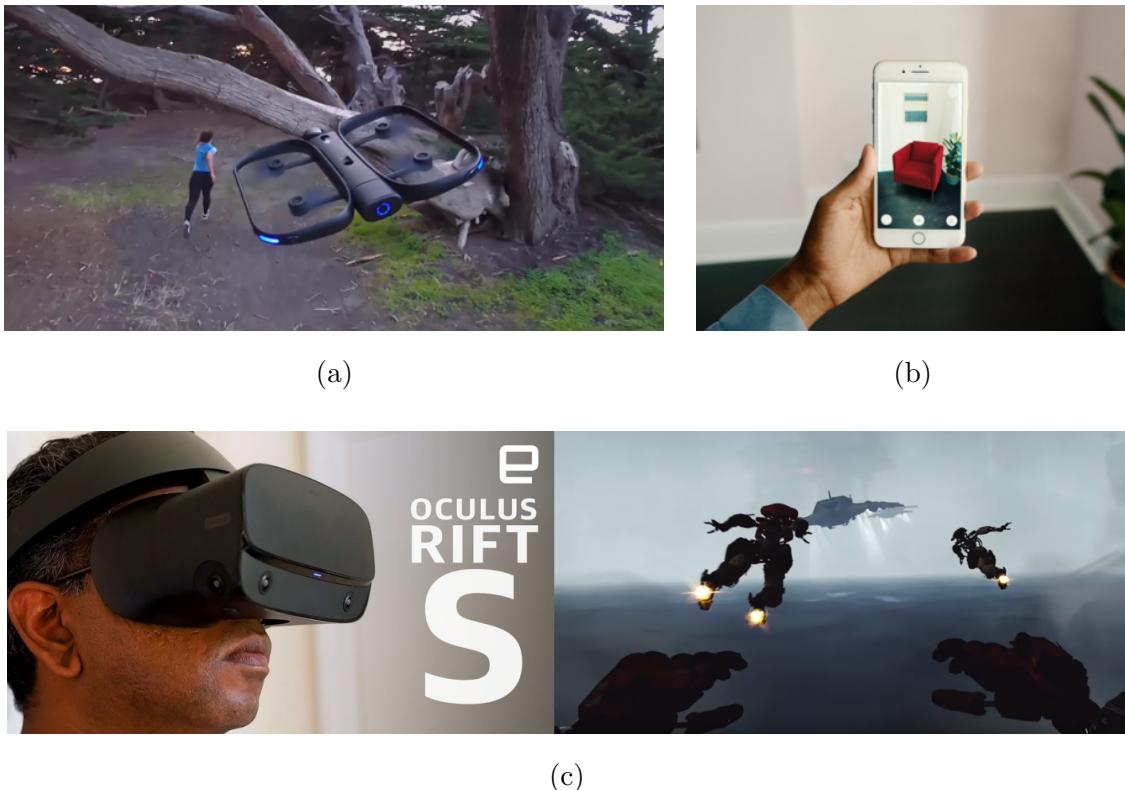


Figura 1.1: Ejemplos de productos comerciales que usan cámaras calibradas. **a:** Dron *Skydio R1*, desarrollado por la empresa Skydio Inc., permite la detección y el seguimiento de personas. **b:** Aplicación de realidad aumentada de IKEA. **c:** Gafas de realidad virtual *Oculus Rift S*, producidas por Oculus VR.

la calibración de la cámara para estimar su movimiento y la reconstrucción de la escena.

El procedimiento de calibración consiste habitualmente en la captura de varias imágenes de un patrón de geometría conocida, realizadas desde distintos puntos de vista como los de la Figura 1.2, siendo muy estrecha la relación entre estas vistas y el error en la estimación de los parámetros de calibración.

En general, las diferentes técnicas de calibración existentes, se pueden agrupar atendiendo a la manera en la que los parámetros son estimados de la siguiente forma [17, 20, 16]:

- Técnicas lineales: Se caracterizan por su sencillez y rapidez. Proporcionan una solución cerrada (sin iteraciones) pero no son capaces de tener en cuenta la distorsión (no lineal) producida por la curvatura de las lentes utilizadas,



Figura 1.2: Diferentes vistas de un patrón de calibración.

resultando en calibraciones que pueden ser poco precisas. El trabajo realizado por Y. Abdel-Aziz [1] es uno de los ejemplos más conocidos de esta técnica.

- Técnicas no lineales: En ellas sí se tiene en cuenta la distorsión gracias a un procedimiento iterativo, lo que permite aumentar la precisión en la estimación de los parámetros. Sin embargo, el algoritmo iterativo necesita una semilla cercana a la solución real para converger. El trabajo realizado por D. C. Brown [7] es un buen ejemplo de esta técnica.
- Técnicas de doble paso: Fueron las últimas en desarrollarse, y son una combinación de las anteriores. En un primer paso obtienen una solución cerrada de los parámetros a estimar mediante una aproximación, para posteriormente realizar una optimización no lineal iterativa. Gracias al primer paso, la convergencia es generalmente mejor que en las técnicas no lineales ya que entre otras ventajas, lo habitual es que se necesiten un menor número de iteraciones. Ejemplos de esta técnica comúnmente utilizados los encontramos en los trabajos de Z. Zhang [27], J. Heikkilä [11] o R. Tsai [26].

Debido a las ventajas mencionadas para la última técnica, es la que más comúnmente se utiliza en aplicaciones de visión por computador [17] y en la que nos centraremos en este TFG.

1.1. Motivación

El trabajo de Z. Zhang [27], facilitó enormemente la calibración de cámaras al necesitar únicamente una superficie plana con un patrón similar al de un tablero de

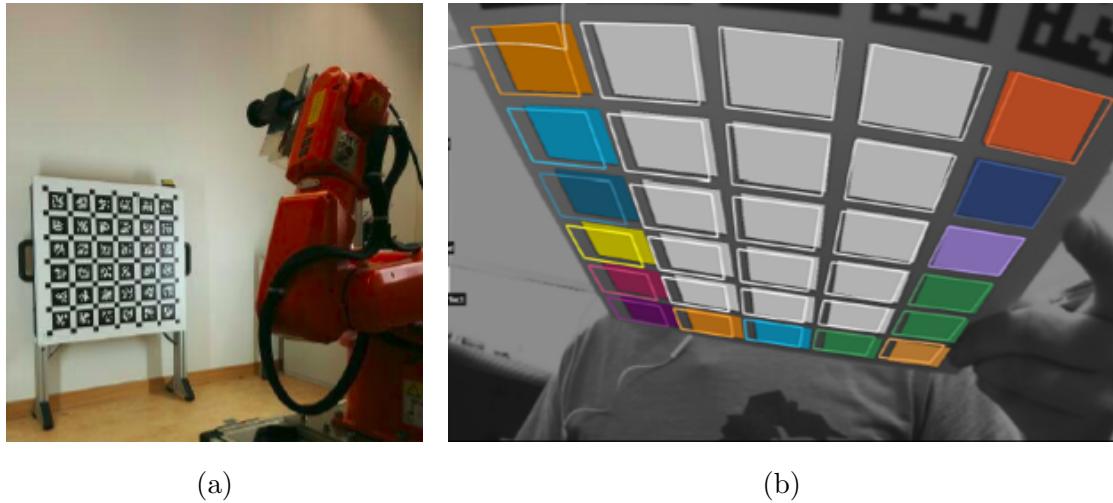


Figura 1.3: Tipos de ayuda en la colocación de los patrones. **a:** Automatizada, por medio de un brazo robótico que controla la posición de la cámara [3]. **b:** Visual, a través de la proyección del tablero en las imágenes capturadas por la cámara [18].

ajedrez, denominada rejilla o patrón de calibración (el que ya se ha mostrado en la Figura 1.2).

La razón por la que la orientación y posición del patrón influyen en la precisión de la calibración, es porque éstas determinan las ubicaciones en la imagen de las esquinas de los cuadrados blancos y negros que lo conforman, siendo éstas las entradas del sistema.

Gracias a este método se pueden obtener resultados precisos sin llevar a cabo grandes inversiones como las necesarias en otras técnicas, como en la ya citada [26] en la que es necesario conocer la traslación de la cámara a calibrar, lo que encarece el equipo necesario. Es por estas razones por las que el trabajo de Z. Zhang [27] es uno de los métodos de calibración más comúnmente utilizados en la actualidad.

En los últimos años, varios trabajos de investigación han realizado mejoras sobre esta técnica básica de calibración, centrándose en cómo optimizar la colocación del patrón/cámara para minimizar la varianza en la estimación de los parámetros.

Algunos de los trabajos que siguen esta tendencia son [18, 15, 3, 19], en los que se establecen criterios que permiten calcular qué posición y orientación del patrón de calibración es la mejor a partir de las capturas realizadas previamente. En la Figura 1.3 se muestra cómo es la implementación de dos de estas técnicas.

Sin embargo, es escasa la literatura acerca de cómo, desde un inicio, sin entradas correspondientes a algunas posiciones y orientaciones, se debería colocar la cámara o el patrón de calibración. En otras palabras, no se ha trabajado sobre un criterio general que permita establecer desde un inicio cuáles son las posiciones óptimas de la cámara/patrón y es en esta línea donde queremos contribuir con este TFG.

1.2. Objetivos

El objetivo que pretendemos alcanzar con este TFG es desarrollar un método nuevo para la calibración de una cámara, así como mostrar su desempeño con respecto al estado del arte. Para ello, más concretamente, se realizarán las siguientes tareas:

- Analizar la influencia de distintas orientaciones y posiciones del patrón de calibración en la precisión del proceso; atendiendo para ello a distintos criterios como la distancia con respecto a la cámara.
- En base al análisis anterior, desarrollar un nuevo procedimiento de calibración que mejore al estado del arte en precisión y tiempo de ejecución en datos simulados.
- Elaborar una herramienta interactiva que permita la implementación del método desarrollado en el punto anterior.
- Con esta última herramienta, llevar a cabo un análisis experimental con imágenes reales para verificar las conclusiones de simulación y realizar estudios adicionales si fuera necesario.

1.3. Herramientas utilizadas

El lenguaje de programación utilizado en este trabajo ha sido MATLAB [13], además se ha hecho uso de la toolbox desarrollada por el investigador J.Yves Bouguet de calibración de una cámara [4].

1.4. Estructura del trabajo

De aquí en adelante, el TFG está organizado de la siguiente forma.

Capítulo 2. Formación de la imagen: En este capítulo se tratan los aspectos fundamentales sobre el proceso que permite a una cámara formar una imagen. Se tratará brevemente cómo es la convergencia de la luz a través de una lente, y se explicará con detalle el modelo matemático detrás de esta formación y que recibe el nombre de modelo pin-hole.

Capítulo 3. Proceso de calibración: En esta parte se explicará con detalle los aspectos más importantes del proceso de calibración, el cual se divide en la obtención de una solución cerrada y en la optimización de esta primera solución inicial. Este capítulo junto con el anterior, sirven para entender y dar contexto al resto del trabajo.

Capítulo 4. Simulación: Se muestran los análisis realizados acerca de la relación entre la distancia y la orientación del patrón de calibración relativo a la cámara, con la precisión de la calibración. Además en este capítulo se presenta una propuesta base de cómo colocar los patrones así como su comparación con el estado del arte.

Capítulo 5. Implementación con imágenes reales: Aquí se pone a prueba el modelo desarrollado en el capítulo anterior, y se ve si en la realidad rinde de la misma manera que en la simulación. Tras ello estudiamos las posibles limitaciones y tras análisis adicionales definimos el método para situar de manera adecuada los patrones de calibración.

Capítulo 6. Conclusiones: En este capítulo se realiza un resumen y análisis crítico sobre el método desarrollado, y se proponen posibles líneas futuras en relación a este trabajo.

Capítulo 2

FORMACIÓN DE LA IMAGEN

En este capítulo se explicarán los aspectos geométricos relacionados con el proceso de formación de una imagen en una cámara y su modelo matemático. Esto será de utilidad para entender y desarrollar el resto del trabajo.

Se comenzará con una breve explicación acerca de cuál es la función que desempeñan las lentes, para luego definir la transformación que se produce al capturar una imagen del mundo 3D, así como los parámetros que la definen y que son los que deberemos estimar en la calibración.

2.1. Modelo de las lentes

La formación de una imagen en el caso de una cámara digital, se consigue por medio de una lente o conjunto de ellas, a través de las cuales los rayos de luz, provenientes del exterior, son proyectados sobre un chip compuesto por elementos fotosensibles denominados fotositos, que codifican de manera digital la cantidad de luz recibida de dichos rayos [23, 6].

En la Figura 2.1a se muestra este proceso de formación para el caso de una lente delgada. En ella se observa cómo tres posibles rayos (en amarillo) provenientes del objeto, al atravesar la lente se proyectan en el plano imagen o chip fotosensible, lo que permite la formación de una imagen invertida del objeto. En esta Figura también se muestra el eje óptico, el cual se obtiene uniendo los centros de curvatura

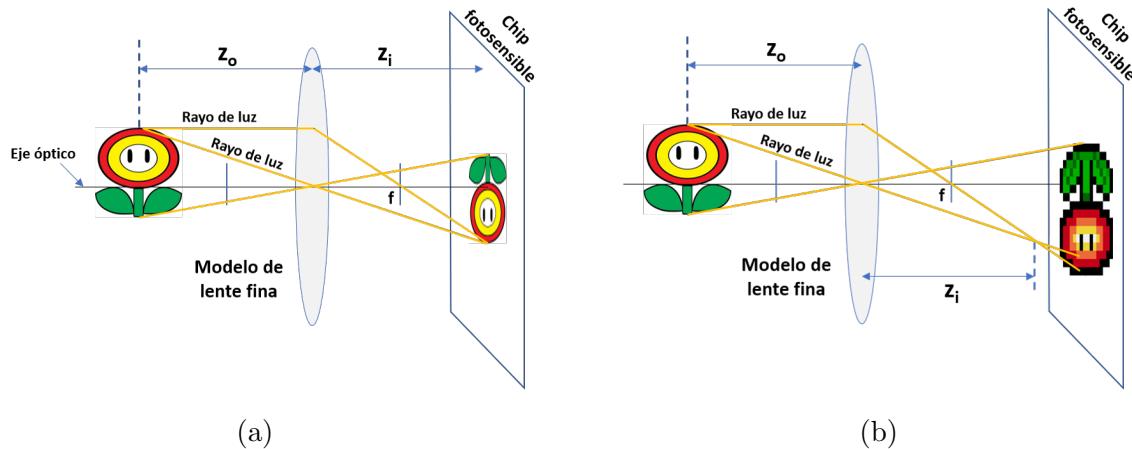


Figura 2.1: Proyección de la luz según el modelo de lente fina. **a:** Objeto enfocado idealmente al converger los rayos en el mismo punto en el que está el sensor. **b:** Objeto con un cierto desenfoque al no ocurrir lo anterior.

de las dos caras de la lente.

Este comportamiento viene determinado por la ley de las lentes:

$$\frac{1}{z_o} + \frac{1}{z_i} = \frac{1}{f}$$

Donde f es la distancia focal, la cual nos da una indicación de cómo es de fuerte la convergencia de los rayos que atraviesan la lente, z_o representa la distancia entre el objeto y la lente y z_i la distancia entre la lente y la imagen enfocada ¹.

A partir de esto, podemos deducir que si los rayos de luz provenientes de un objeto al ser proyectados sobre el chip o plano imagen, presentan una distancia z_i diferente a la distancia a la que se sitúa el plano imagen, esto provocará que el objeto aparezca con un cierto desenfoque, cuya magnitud vendrá dada en función de la profundidad de campo del conjunto de lentes utilizado [6] (ver Figura 2.1b).

En la realidad, lo habitual es que una cámara esté compuesta por un conjunto más sofisticado de lentes que permitan disminuir efectos no deseables como la aberración esférica y cromática, producidas por la geometría de la lente y por las diferencias en la convergencia de rayos de luz de diferente color respectivamente, y que provocan que la distancia focal varíe en función de éstas. Para más información acerca de esto,

¹Con imagen enfocada, nos referimos a que, considerando un objeto 2D como el de la Figura 2.1a, éste aparecerá perfectamente enfocado si el cruce de los rayos provenientes de todos sus puntos tras travesar la lente, se producen en el chip fotosensible. Para objetos 3D el razonamiento es igual.

se recomiendan los trabajos [23, 12]. Un ejemplo de conjunto de lentes de este tipo es el denominado como lentes de diseño *Double-Gauss* mostrado en la Figura 2.2.

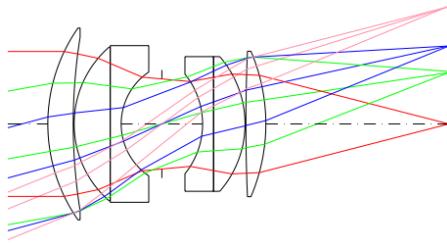


Figura 2.2: Diseño de lentes *Double-Gauss*. Figura extraída de [12].

Por otro lado, en visión por computador es muy común aproximar este comportamiento real de las lentes mediante el modelo pin-hole [23, 6, 27], explicado en la siguiente sección, y que por tanto es el que se utilizará para el desarrollo de este trabajo. Para más información acerca de los distintos modelos de cámara, se pueden consultar las siguientes fuentes [10, 22].

2.2. Modelo pin-hole

En el modelo pin-hole, se presenta un modelo de cámara simplificado, en el cual solo se permite el paso de los rayos a través de una pequeña apertura o agujero (ver Figura 2.3), en vez de a través de una lente o conjunto de ellas como se ha explicado anteriormente. Al proyectarse estos rayos en el plano imagen, al igual que en el caso anterior, forman una imagen invertida en el lado opuesto de la cámara del objeto del que provienen dichos rayos.

Esta simplificación trae consigo varias consecuencias. El hecho de que todos los rayos de luz provenientes de un punto en el espacio tengan que pasar por esta pequeña apertura, que podemos considerar como un punto [6, 23], significa que todos los rayos proyectados en el plano imagen de dicho punto espacial, provienen de una misma dirección. Esto provoca que aspectos comentados anteriormente como el desenfoque, no aparezcan en este modelo.

Además, desde un punto de vista geométrico, es equivalente tratar a la imagen formada del entorno en un plano imagen “virtual” situado delante del plano imagen y a la misma distancia que el plano imagen en el lado opuesto de la cámara. Esto

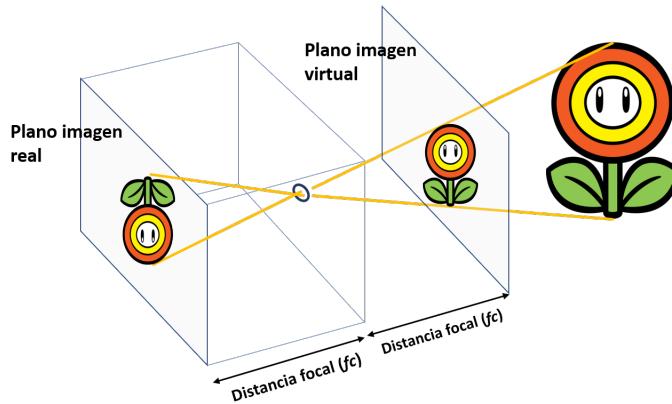
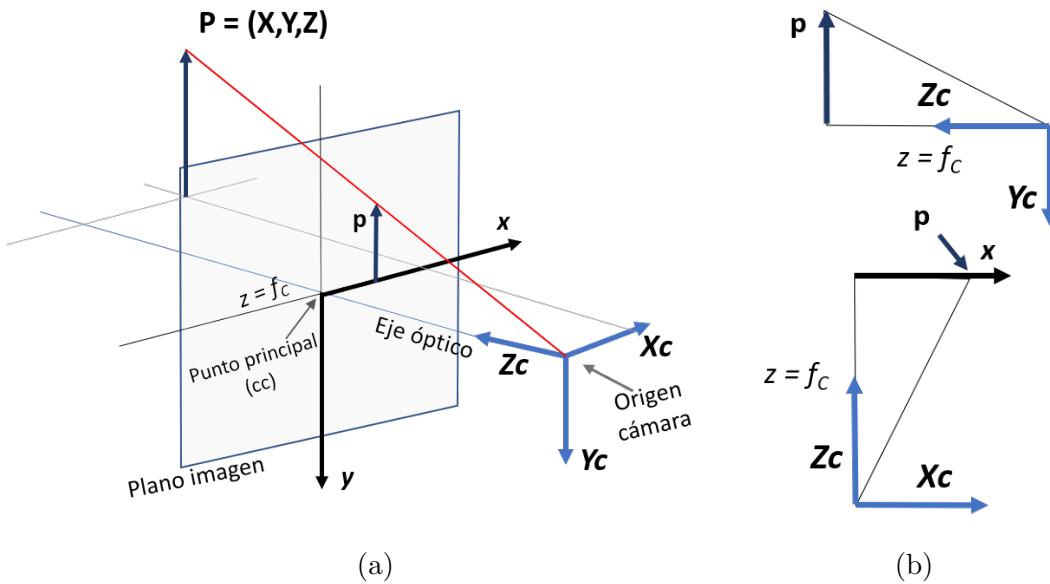


Figura 2.3: Modelo pin-hole de formación de la imagen.

Figura 2.4: Modelo pin-hole presentado de manera esquemática. a: Vista general. b: punto p visto de perfil (imagen superior) y desde arriba (imagen inferior).

posibilita trabajar con una imagen no invertida en el modelo. Este plano imagen “virtual”, que por comodidad de aquí en adelante nos referiremos a él como **plano imagen**, se puede observar en las Figuras 2.3 y 2.4a.

A su vez en este modelo, el punto que representa la apertura que se ha comentado, se corresponde con el origen del sistema de coordenadas que define la posición y orientación de la cámara y que, por convención [6, 23, 27], tal y como se aprecia en la Figura 2.4a, el eje Z_C , también denominado como **eje óptico**, es perpendicular al plano sensor o imagen y con un sentido que “apunta” hacia el entorno, y los ejes X_C e Y_C son paralelos al borde horizontal y vertical del plano sensor respectivamente.

En la misma Figura, 2.4a, también aparecen dos de los parámetros que definen este modelo y que, saber lo que son, nos servirá para poder continuar este desarrollo:

- **Punto principal (cc)**: representa el punto de corte del eje óptico (eje Z_C) con el plano sensor.
- **Distancia focal (fc)**: representa la distancia que hay entre el punto principal y el origen de la referencia cámara, medida, por tanto, sobre el eje Z_C .

Ahora, observando ambas figuras de 2.4 y considerando los puntos generales que aparecen en ellas: $P = (X, Y, Z)$ en el entorno y $p = (x, y, z)$ en plano sensor, ambos, vistos desde la referencia cámara, se puede concluir que:

$$x = f \frac{X}{Z} \quad y = f \frac{Y}{Z} \quad (2.1)$$

Por otro lado, como la localización de los puntos en una imagen viene dada en coordenadas medidas en píxeles, debemos transformar las unidades si queremos relacionar los puntos del espacio con los de una imagen. Para ello, tenemos que recordar que el plano imagen de una cámara digital está formado por fotositos, dispuestos en una rejilla de tamaño $W \times H$, correspondiéndose cada uno de ellos con un píxel de la imagen formada [6].

En la Figura 2.5a se muestra una representación del plano imagen con los píxeles o fotositos discretizados, así como su correspondiente sistema de coordenadas (u, v) . Igualmente, por convención es habitual tomar el origen de este sistema de coordenadas tipo píxel en el centro del píxel situado en el borde izquierdo superior del plano imagen [6, 23, 10].

De esta forma, en coordenadas píxel, el punto principal lo podemos expresar como: $cc = (u_o, v_o)$; de igual forma, la ecuación 2.1 también la podemos expresar en coordenadas píxel desde el sistema de referencia (u, v) tal como se muestra a continuación en 2.2:

$$u = \frac{x}{\rho_w} + u_o \quad v = \frac{y}{\rho_h} + v_o \quad (2.2)$$

Donde ρ_w representa el ancho de los píxeles y ρ_h representa la altura de los mismos.

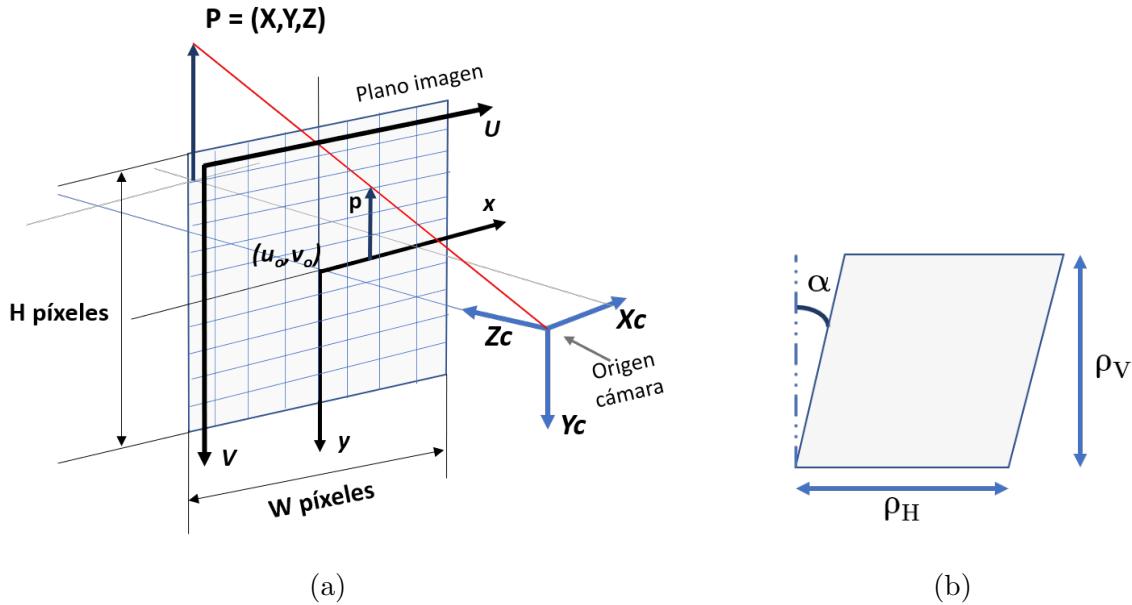


Figura 2.5: Modelo pin-hole con el sistema de coordenadas píxel (u, v) . **a:** Vista general. **b:** posible falta de perpendicularidad de un píxel.

El desarrollo hasta este momento, se puede expresar matricialmente de la forma mostrada en la Ecuación 2.3; en ella, tanto el punto en coordenadas píxel (\tilde{p}) como el punto en coordenadas desde la referencia cámara (\tilde{P}) aparecen representados en coordenadas homogéneas², debido a esto, para obtener los puntos en las coordenadas no homogéneas es necesario dividir por su último componente, tal y como se muestra en la Ecuación 2.4.

$$\tilde{p} = \begin{pmatrix} \frac{f_c}{\rho_w} & 0 & u_o & 0 \\ 0 & \frac{f_c}{\rho_h} & v_o & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \tilde{P} \quad (2.3)$$

$$\tilde{p} = \begin{pmatrix} u \\ v \\ w \end{pmatrix} \rightarrow p = \begin{pmatrix} \frac{u}{w} \\ \frac{v}{w} \end{pmatrix} \quad \tilde{P} = \begin{pmatrix} X \\ Y \\ Z \\ W \end{pmatrix} \rightarrow P = \begin{pmatrix} \frac{X}{W} \\ \frac{Y}{W} \\ \frac{Z}{W} \end{pmatrix} \quad (2.4)$$

La Ecuación 2.3 representa una transformación perspectiva, puesto que, para proyectar en el plano imagen los puntos vistos desde la referencia cámara, estamos

²Para distinguir vectores en coordenadas homogéneas y euclídeas utilizaremos una notación con/sin gorro. \tilde{P} se refiere al punto en coordenadas homogéneas, y P al punto en coordenadas euclídeas.

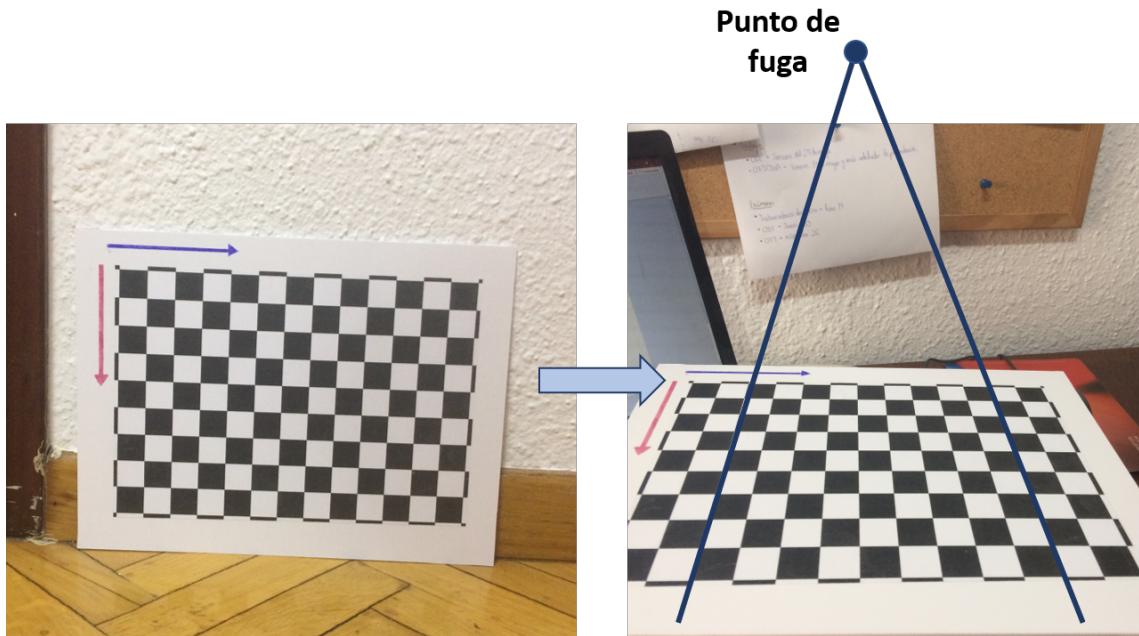


Figura 2.6: Ejemplo de cómo líneas paralelas en 3D se cortan en un punto de fuga (*vanishing point*) tras una transformación perspectiva.

dividiendo las componentes x e y por su componente z [23, 6, 10], algo que también se puede observar en la ecuación 2.1.

Cuando se realiza este tipo de transformación, una de las pocas propiedades que se conserva de los objetos, es la de mantener rectas las líneas que lo son; sin embargo, la orientación, las longitudes, los ángulos y el paralelismo que pueden presentar ciertos objetos, no tienen por qué mantenerse [6, 23].

Un efecto interesante resultado de esto, es el hecho de que líneas que en la realidad son paralelas, al hacerles una foto, si éstas no son paralelas al plano imagen, se cortarán en el punto denominado como punto de fuga o *vanishing point*, tal y como se muestra en la Figura 2.6. En relación a esto, una utilidad de las coordenadas homogéneas es que podemos representar este punto, que en la realidad estaría situado en el infinito, colocando un cero como último elemento del vector.

Por último en cuanto al modelo pin-hole, se puede incluir un parámetro que tenga en cuenta la falta de perpendicularidad entre los bordes de los píxeles que conforman el plano sensor, tal y como se muestra en la Figura 2.5b. Para incluirlo en el modelo, bastaría con completar la Ecuación 2.1 realizando lo mostrado en la Ecuación 2.5 o equivalentemente completando la Ecuación 2.3 como se presenta en

la Ecuación 2.6.

$$x = f \frac{X}{Z} + f \frac{Y}{Z} \tan(\alpha) \quad (2.5)$$

$$\tilde{p} = \begin{pmatrix} \frac{fc}{\rho_w} & \frac{fc}{\rho_w} \tan(\alpha) & u_o & 0 \\ 0 & \frac{fc}{\rho_h} & v_o & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \tilde{P} \quad (2.6)$$

Sin embargo, esta inclinación es despreciada normalmente, debido a que las técnicas habituales de fabricación de sensores permiten que este parámetro sea insignificante [6, 23].

2.3. Modelo de distorsión

Debido a las simplificaciones llevadas a cabo en el modelo pin-hole, un aspecto importante como la distorsión debido a la geometría de las lentes usadas en una cámara y que provocan el desplazamiento de los puntos en el plano imagen con respecto a la ubicación calculada, no está incluido.

Este tipo de distorsión se define por dos componentes, la primera es la distorsión radial, cuya magnitud de desplazamiento está relacionada con la distancia del punto en la imagen al punto principal; y la segunda es la tangencial, cuya magnitud se relaciona con el producto de la coordenada vertical y horizontal de un punto en el plano imagen.

De estas dos componentes, la distorsión radial es generalmente la dominante [6, 28], y se tiende a no considerar a la distorsión tangencial [27, 23]. En este trabajo solo presentaremos los aspectos más importantes relacionados con la distorsión radial, Para más información se recomienda la lectura de [2].

Como hemos introducido antes, la distorsión radial es la causa de que los puntos en la imagen sean desplazados según una dirección radial con centro en el punto principal (*cc*) de la cámara (ver Figura 2.7), por lo que se puede aproximar con precisión con un polinomio como el de la Ecuación 2.7 [4, 23].

$$\delta r = k_1 r^2 + k_2 r^4 + k_3 r^6 + \dots \quad (2.7)$$

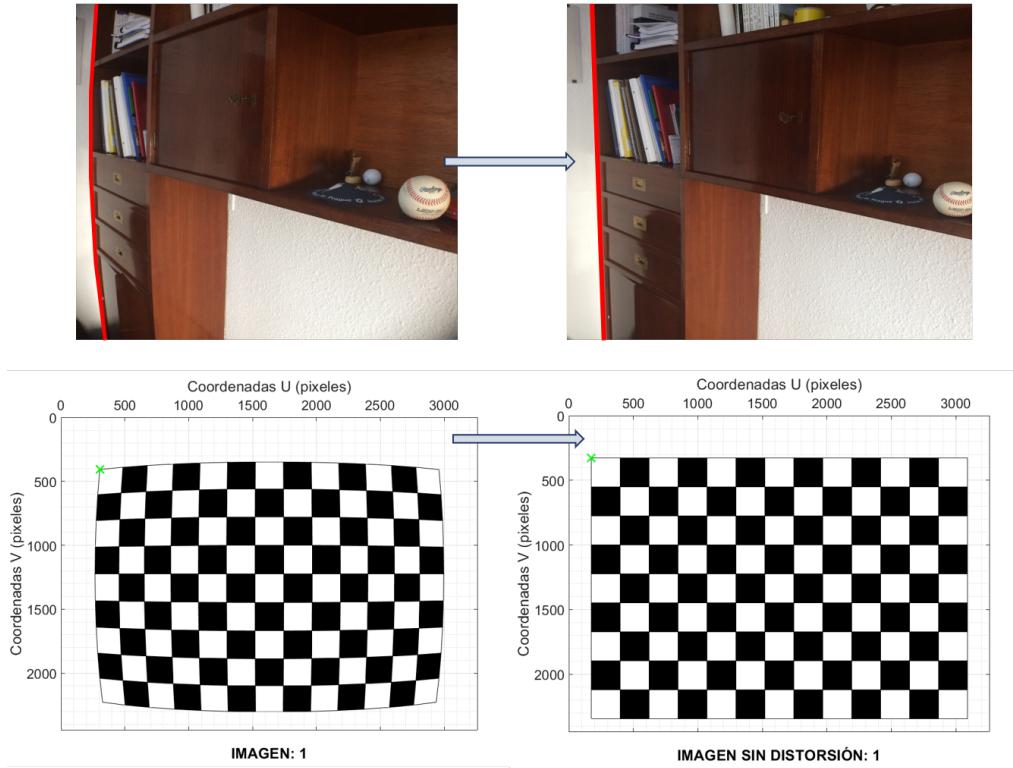


Figura 2.7: Imágenes reales (arriba) y simuladas (debajo) con un modelo de cámara con coeficientes de distorsión $k_1 = -0,146$ y $k_2 = -0,015$ (izquierda) y sin distorsión (derecha). En ellas se observan como, al tener distorsión radial, las líneas se curvan cada vez más conforme nos acercamos a los bordes de la imagen.

Donde δr representa el desplazamiento que sufre el punto en la imagen con respecto a la ubicación calculada por el modelo pin-hole, r representa la distancia radial del punto en la imagen al punto principal y k_i los coeficientes a estimar y que definen la magnitud de este tipo de distorsión en la imagen.

Cuanto mayor sea el grado de este polinomio, mayor será la distorsión radial que se puede modelar. Lo habitual es usar los dos primeros coeficientes [27, 23].

Para introducir la distorsión radial en el modelo que hemos desarrollado, modificamos la Ecuación 2.3 de la siguiente forma:

$$\tilde{p} = \begin{pmatrix} \frac{f_c}{\rho_w} & 0 & u_o \\ 0 & \frac{f_c}{\rho_h} & v_o \\ 0 & 0 & 1 \end{pmatrix} \tilde{x}_d \rightarrow \tilde{x}_d = \begin{pmatrix} \frac{X}{Z}(1 + k_1r^2 + k_2r^4) \\ \frac{Y}{Z}(1 + k_1r^2 + k_2r^4) \\ 1 \end{pmatrix} \quad (2.8)$$

Donde $r^2 = (\frac{X}{Z})^2 + (\frac{Y}{Z})^2$, lo que representa la distancia radial (al cuadrado) desde el origen de la referencia (x, y) situado en el centro del plano sensor de la Figura 2.5a

de manera normalizada; es decir, considerando una distancia focal $fc = 1$ unidad (con la que se esté trabajando, por ejemplo mm).

De esta forma, frente a posibles variaciones en la distancia focal de la cámara, la estimación de los coeficientes de distorsión teóricamente no se ven afectados y es así como está configurado el modelo de cámara en [4], la herramienta principal que se utilizará para la elaboración de este trabajo.

2.4. Transformación mundo \leftrightarrow cámara

Hasta ahora, el modelo que hemos explicado nos permite obtener las coordenadas en píxeles de cualquier punto situado en el entorno que esté referenciado al sistema de coordenadas que define a la cámara.

Sin embargo, en la práctica conocer los puntos del entorno desde la referencia cámara no es siempre posible. Por lo tanto, lo último necesario para completar el modelo es introducir la transformación necesaria de una referencia desde la que sí conozcamos la ubicación de estos puntos (referencia mundo), a la referencia de la cámara.

Esta transformación, compuesta por una rotación y una traslación que se efectúan sobre la referencia mundo para pasar a la referencia cámara, se puede expresar de la siguiente manera:

$$P = (\mathbf{R} \quad \mathbf{t}) \tilde{M} \quad (2.9)$$

Donde R (3x3) es la matriz de rotación, t (3x1) es el vector de traslación, \tilde{M} son las coordenadas (conocidas) de las esquinas del patrón de calibración desde la referencia mundo y P por tanto, son las coordenadas de las esquinas del patrón desde la referencia cámara. De una forma más visual, esto aparece en la Figura 2.8.

Además tal y como se puede apreciar en esta figura, en el proceso de calibración de una cámara, la referencia o sistema de coordenadas mundo, está ubicada en el patrón de calibración. De esta forma, se pueden conocer de antemano la ubicación de todas las esquinas del patrón, ya que la geometría de éste es conocida.

Con esto se termina el capítulo acerca del proceso de formación de una imagen

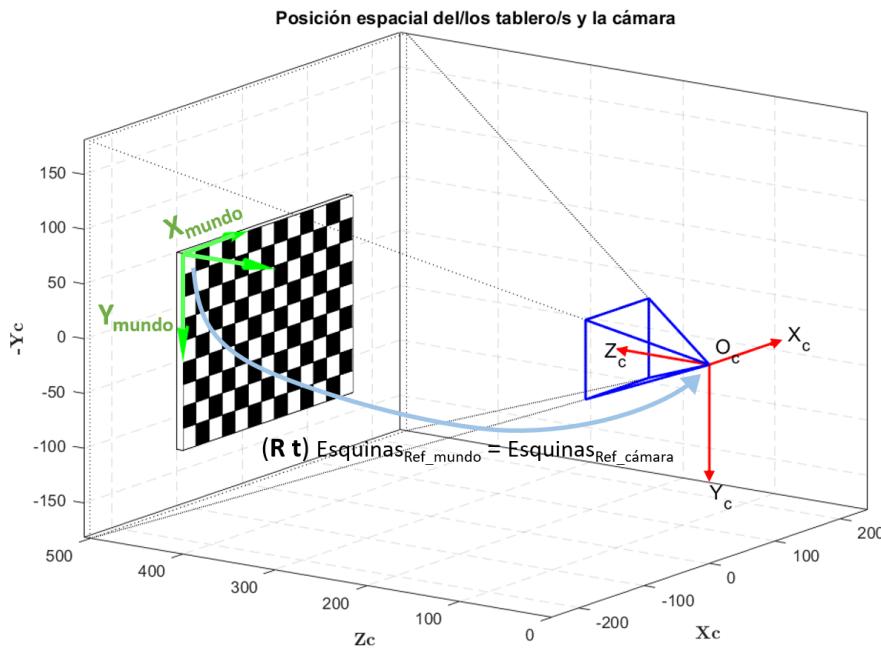


Figura 2.8: Vista 3D del patrón de calibración y la cámara (simplificada en azul con su ángulo de visión)

y cómo se puede modelar matemáticamente. Esta base nos servirá para entender y llevar a cabo el desarrollo del siguiente capítulo que trata los aspectos teóricos acerca del proceso de calibración de la cámara.

Capítulo 3

CALIBRACIÓN DE UNA CÁMARA CON PATRÓN CONOCIDO

En este capítulo se van a presentar los conceptos teóricos necesarios para entender el proceso de calibración de una cámara. Como ya se ha comentado en la introducción, este es el proceso que permite obtener los parámetros intrínsecos que definen a una cámara y que han sido presentados en el capítulo anterior.

El método explicado en este TFG es el desarrollado por el investigador Z. Zhang en [27, 28], en el que se requiere capturar varias imágenes de un patrón de calibración como los que hemos enseñado en los capítulos anteriores (ver Figura 1.2 por ejemplo), localizando, mediante un algoritmo de detección como el de [9], las esquinas en la imagen del patrón (ver el ejemplo de la Figura 3.1).

Con estas localizaciones detectadas, se calculan los parámetros del modelo pinhole, con distorsión incluida, así como su orientación y posición, que mejor estiman la ubicación de las esquinas en la imagen, en base a su ubicación 3D desde la referencia mundo ubicada en el patrón (tal y como se observa en la Figura 2.8), las cuales al conocer la geometría del patrón (número de cuadrados y longitud de éstos), también son conocidas.

Este proceso se divide fundamentalmente en dos fases, la primera de ellas es la

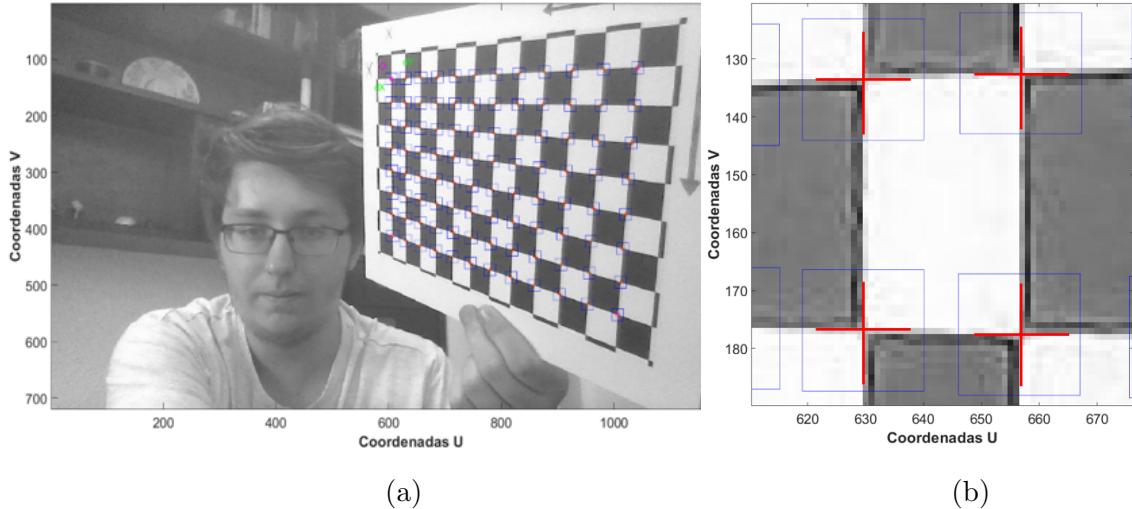


Figura 3.1: Ejemplo de detección de esquinas mediante el algoritmo de detección implementado en [4], el cual está basado en [9]. **a:** Perspectiva de la cámara. **b:** Zoom de **a** con la detección (cruces rojas) aumentada por motivos de visualización.

obtención de una primera estimación de los parámetros mediante una solución cerrada; y la segunda es la optimización no lineal de esta primera estimación mediante un procedimiento iterativo.

Dedicaremos una primera parte para expresar la notación que se utilizará en el cálculo de la solución cerrada, sirviéndonos de lo elaborado en el capítulo anterior. En una segunda parte se explicará cómo obtener esta solución cerrada, basándonos principalmente en los trabajos [27, 28]. Por último se explicará el procedimiento de optimización no lineal.

3.1. Notación

En este capítulo, una esquina del tablero en 3D será representada por $M = (X, Y, Z)^T$ mientras que su proyección 2D en el plano imagen será representada por: $m = (u, v)^T$. Para expresar estos puntos en coordenadas homogéneas se utilizará la misma notación con gorro que hemos usado hasta ahora; es decir, $\tilde{M} = (X, Y, Z, 1)^T$ y $\tilde{m} = (u, v, 1)^T$. De acuerdo a esto y recuperando la Ecuación 2.9 tenemos:

$$s\tilde{m} = A \begin{pmatrix} R & t \end{pmatrix} \tilde{M} \quad (3.1)$$

Donde s es el factor de escala que se utiliza para conseguir un 1 como último elemento del vector \tilde{m} y así tener directamente las coordenadas no homogéneas en sus dos primeros elementos.

Por otro lado, A es la matriz que contiene los parámetros intrínsecos que permiten la proyección de un punto desde el sistema de coordenadas cámara al sistema de coordenadas (u, v) del plano imagen y que ya había sido introducida en la Ecuación 2.3:

$$A = \begin{pmatrix} fc_1 & 0 & u_o \\ 0 & fc_2 & v_o \\ 0 & 0 & 1 \end{pmatrix} \quad (3.2)$$

Con el objetivo de facilitar la lectura, de aquí en adelante nos referiremos a $\frac{fc}{\rho_w}$ como fc_1 y a $\frac{fc}{\rho_h}$ como fc_2 , representando por tanto ambos parámetros la distancia focal medida en píxeles, al igual que u_o y v_o (coordenadas del punto principal) que también se miden en píxeles.

3.2. Obtención de la solución cerrada

El objetivo de esta solución cerrada es de servir como “semilla” para el método de optimización iterativo posterior y así favorecer su convergencia. Por ello, esta solución cerrada no es más que una aproximación de la solución real, que surge de no considerar la distorsión radial que más tarde es introducida en el proceso de optimización.

Para favorecer la comprensión de esta sección así como su organización, se va a dividir en las tres partes diferenciadas: cálculo de la matriz de homografía, restricciones en los parámetros intrínsecos y solución final.

3.2.1. Cálculo de la matriz de homografía

Como la referencia mundo está ubicada en una de las esquinas del patrón de calibración con los ejes X_{mundo} e Y_{mundo} paralelos a los bordes de dicho patrón (Figura 2.8), la coordenada Z de las esquinas desde esta referencia es nula. Sabiendo

esto e introduciéndolo en la Ecuación 3.1, nos queda:

$$s \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = A \begin{pmatrix} r_1 & r_2 & r_3 & t \end{pmatrix} \begin{pmatrix} X \\ Y \\ 0 \\ 1 \end{pmatrix} = A \begin{pmatrix} r_1 & r_2 & t \end{pmatrix} \begin{pmatrix} X \\ Y \\ 1 \end{pmatrix}$$

Donde se ha representado a la matriz de rotación R por sus columnas r_i . Por otro lado, como $Z = 0$ se cumple siempre, podemos seguir llamando a \tilde{M} de la forma $\tilde{M} = (X, Y, 1)^T$.

De esta forma, la matriz que relaciona las coordenadas de las esquinas 3D con sus respectivas coordenadas píxel y que es propia de cada imagen capturada del patrón de calibración, denominada como matriz de homografía (H), viene dada por la Ecuación 3.3.

$$H_{3 \times 3} = \begin{pmatrix} h_{f1} \\ h_{f2} \\ h_{f3} \end{pmatrix} = A \begin{pmatrix} r_1 & r_2 & t \end{pmatrix} \quad (3.3)$$

Donde se ha representado por sus filas h_{fi} . Debido a esto, el primer paso para obtener los parámetros intrínsecos, es el de estimar los elementos de esta matriz de homografía. Para ello, podemos plantear un sistema de ecuaciones donde las incógnitas sean los propios elementos de la matriz. Esto se puede conseguir relacionando las componentes de m con las de \tilde{M} :

$$m = \begin{pmatrix} u \\ v \end{pmatrix} = \frac{1}{h_{f3}\tilde{M}} \begin{pmatrix} h_{f1}\tilde{M} \\ h_{f2}\tilde{M} \end{pmatrix}$$

A partir de la ecuación anterior:

$$\begin{pmatrix} \tilde{M}^T & 0_{1 \times 3} & -u\tilde{M}^T \\ 0_{1 \times 3} & \tilde{M}^T & -u\tilde{M}^T \end{pmatrix} \begin{pmatrix} h_{f1}^T \\ h_{f2}^T \\ h_{f3}^T \end{pmatrix} = Lx = 0_{2n \times 1} \quad (3.4)$$

En cuanto al número de incógnitas, la matriz de homografía está formada por 9 elementos, pero como está definida según un factor de escala tal y como se observa en la Ecuación 3.1, el número de soluciones que podrían cumplir el sistema 3.4 es infinito. Para solventar esto, se puede adoptar $H(3, 3) = 1$ o lo que es lo mismo $x(9) = 1$ (esto se realiza así por ejemplo en [4]).

De esta forma, el número de esquinas que han de tenerse en cuenta debe ser ≥ 4 puesto que cada una aporta 2 ecuaciones a 3.4. Es decir, si se aportan n esquinas, las dimensiones de L serán de $2n \times 9$.

La solución a este sistema se corresponde con el vector propio asociado al valor propio más pequeño de la matriz $L^T L$ de la Ecuación 3.4. Este cálculo se puede realizar eficientemente mediante la descomposición de $L^T L$ en valores singulares (SVD) y es de esta forma como se realiza en [4]. La demostración a esto se puede encontrar en [8].

Por último en relación a la obtención de la matriz de homografía, decir que debido al ruido que puede estar presente en una imagen por causas como pequeñas diferencias en la ganancia o en el offset de los fotositos [6], la Ecuación 3.1 no siempre se cumple. Es por esto por lo que si $n > 4$ se puede realizar una optimización por mínimos cuadrados minimizando la siguiente suma de errores cuadráticos:

$$\sum_{i=1}^n (m_i - \hat{m}_i)^T (m_i - \hat{m}_i)$$

Donde \hat{m}_i son las coordenadas de las esquinas calculadas mediante la Ecuación 3.1 y m_i son las coordenadas detectadas de las esquinas. Un procedimiento similar de optimización será explicado con más detalle posteriormente.

3.2.2. Restricciones en los parámetros intrínsecos

Teniendo ya calculada la matriz de homografía y expresándola por columnas como $H = (h_1, h_2, h_3)$, si la aplicamos en las ecuaciones 2.1 y 3.3 tenemos:

$$\begin{pmatrix} h_1 & h_2 & h_3 \end{pmatrix} = \lambda A \begin{pmatrix} r_1 & r_2 & t \end{pmatrix}$$

Donde λ al igual que la constante de la Ecuación 3.1 representa un factor de escala. Además, de esta ecuación se desprende que $h_i = \lambda A r_i$ con $i = 1, 2$ y sabiendo que en una matriz de rotación las columnas deben ser ortogonales y por tanto $r_1^T r_2 = 0$:

$$h_1^T (A^{-1})^T A^{-1} h_2 = 0 \quad (3.5)$$

Ahora, atendiendo a la condición de normalidad que también deben tener tanto r_1 como r_2 , podemos igualar sus módulos:

$$h_1^T (A^{-1})^T A^{-1} h_1 = h_2^T (A^{-1})^T A^{-1} h_2 \quad (3.6)$$

Estas dos condiciones, que sirven para relacionar la matriz de parámetros intrínsecos A con las columnas ya calculadas de la homografía, son las que nos permitirán calcular los parámetros intrínsecos tal y como se explicará en la siguiente sub-sección.

Algo importante a señalar sobre esto ahora es la existencia de rotaciones, denominadas como degeneradas, que provocan que las ecuaciones aportadas por dos imágenes o matrices de homografía de dos patrones de calibración con una única rotación sobre su eje Z de diferencia sean combinaciones lineales unas de otras, por lo que en vez de poder extraer 4 ecuaciones (2 por cada imagen) solo podríamos extraer 2.

Por último señalar que, debido a que debemos estimar también los parámetros extrínsecos relacionados con la rotación y la translación del patrón de calibración (6 grados de libertad), al tener la matriz de homografía 8 grados de libertad (recordemos que el noveno se correspondía con la escala), solo podremos estimar dos parámetros intrínsecos con una única homografía, o lo que es lo mismo, con una única imagen del patrón de calibración, por lo que para estimar correctamente los parámetros intrínsecos será necesario tomar varias imágenes de éste.

3.2.3. Solución cerrada final

Como hemos comentado antes, las ecuaciones 3.5 y 3.6, son las que nos permitirán obtener los parámetros intrínsecos. Para facilitar esto nos podemos definir la siguiente matriz:

$$\begin{aligned} \mathbf{B} &= (\mathbf{A}^{-1})^T \mathbf{A}^{-1} = \\ &= \begin{pmatrix} B_{11} & B_{12} & B_{13} \\ B_{12} & B_{22} & B_{23} \\ B_{13} & B_{23} & B_{33} \end{pmatrix} = \begin{pmatrix} \frac{1}{fc_1^2} & 0 & -\frac{u_o f c_2}{fc_1^2 f c_2} \\ 0 & \frac{1}{fc_2^2} & -\frac{v_o}{fc_2^2} \\ -\frac{u_o f c_2}{fc_1^2 f c_2} & -\frac{v_o}{fc_2^2} & \frac{(u_o f c_2)^2}{fc_1^2 f c_2} + \frac{v_o^2}{fc_2^2} + 1 \end{pmatrix} \quad (3.7) \end{aligned}$$

Como se puede ver, esta matriz es simétrica y el término B_{12} es nulo debido a la consideración que ya ha sido argumentada de que $\alpha = 0$ en la ecuación 2.6. Ahora, si reorganizamos los términos de la matriz B de la forma:

$$\mathbf{b} = [B_{11}, B_{22}, B_{13}, B_{23}, B_{33}]^T$$

Y los de h_i (columnas de la matriz de homografía H) de la forma:

$$\mathbf{h}_i = [h_{i1}, h_{i2}, h_{i3}]^T$$

Entonces, generalizando las ecuaciones 3.5 y 3.6 para dos columnas h_i, h_j (con i,j = 1 ó 2), podemos reorganizarlas de la siguiente manera:

$$h_i^T B h_j = v_{ij}^T b \quad (3.8)$$

Donde

$$\mathbf{v}_{ij} = \begin{pmatrix} h_{i1}h_{j1} \\ h_{i2}h_{j2} \\ h_{i3}h_{j1} + h_{i1}h_{j3} \\ h_{i3}h_{j2} + h_{i2}h_{j3} \\ h_{i3}h_{j3} \end{pmatrix}$$

Con lo que, finalmente y de manera compacta, podemos agrupar las ecuaciones 3.5 y 3.6 en el siguiente sistema:

$$\begin{pmatrix} \mathbf{v}_{12}^T \\ (\mathbf{v}_{12} - \mathbf{v}_{22})^T \end{pmatrix} \mathbf{b} = \mathbf{0}_{2n \times 1} \rightarrow \mathbf{Vb} = \mathbf{0}_{2n \times 1} \quad (3.9)$$

Donde \mathbf{V} es una matriz $2n \times 5$. Esto significa que tendremos una solución única del sistema si el número de imágenes capturadas del patrón de calibración es $n \geq 2$ dado que hay que establecer un factor de escala para \mathbf{b} .

Al igual que en el sistema de ecuaciones 3.4 anterior, la solución a este sistema se corresponde con el vector propio asociado al valor propio más pequeño de la matriz $V^T V$.

Con esto, ya tenemos determinados los elementos de la matriz B de la Ecuación 3.7 y solo faltaría despejar de ésta los parámetros intrínsecos para poder obtener la solución cerrada de la matriz A (ecuación 3.2) (despejarlos no aporta mucha información a este TFG por lo que si el lector está interesado puede dirigirse a [27]).

Por último, lo que queda una vez ya hemos calculado los elementos de la matriz A , es calcular los parámetros extrínsecos a partir de ésta. Para ello, a partir de la Ecuación 3.3, se deduce que:

$$\mathbf{t} = \lambda A^{-1} h_3$$

$$\mathbf{r}_1 = \lambda A^{-1} h_1$$

$$\mathbf{r}_2 = \lambda A^{-1} h_2$$

Con $\lambda = 1/\|A^{-1}h_1\| = 1/\|A^{-1}h_2\|$ debido a la normalidad de las columnas de una matriz de rotación. Además, debido a que las columnas de la matriz de rotación son ortogonales:

$$\mathbf{r}_3 = r_1 \times r_2$$

Tras esto, ya hemos completado la estimación de la solución cerrada de los parámetros de la cámara. Como ya hemos dicho en el principio, este desarrollo se corresponde con los trabajos de [27, 28] del investigador Z. Zhang de los que hemos presentado los aspectos fundamentales para dar contexto y entender el resto del trabajo. Para tener más información acerca del proceso se recomienda visitar los artículos citados.

A continuación damos paso al proceso de optimización con el que se mejorará la estimación de esta solución cerrada, teniendo en cuenta aspectos que no han sido tratados en esta sección como la distorsión radial.

3.3. Refinamiento de la solución cerrada

Como hemos dicho anteriormente, la solución cerrada es una aproximación de la solución real al no haberse tenido en cuenta los efectos de la distorsión radial, los cuales aportan un comportamiento no lineal al depender cuadráticamente de la posición en el plano imagen (ver Ecuación 2.8).

Es debido a esto, por lo que para estimar correctamente los parámetros, no basta con recurrir a técnicas algebraicas lineales de resolución como las de la sección anterior; sino que hay que recurrir a técnicas de optimización no lineales como la que se lleva a cabo en [4] y es la que se va a explicar en esta sección.

El método de estimación al cual nos referimos se denomina como “mínimos cuadrados no lineales de Gauss-Newton” (variante del método de optimización de Newton), con el que se minimiza la suma de los errores cuadráticos ponderados (f) entre las coordenadas píxel calculadas (\hat{m}_i) y las coordenadas píxel detectadas (m_i), el cual se denomina como error de reproyección:

$$f(x) = \frac{1}{2} \sum_i^n \Delta m_i(x)^T W_i \Delta m_i(x) \rightarrow \Delta m_i(x) = m_i - \hat{m}_i(x) \quad (3.10)$$

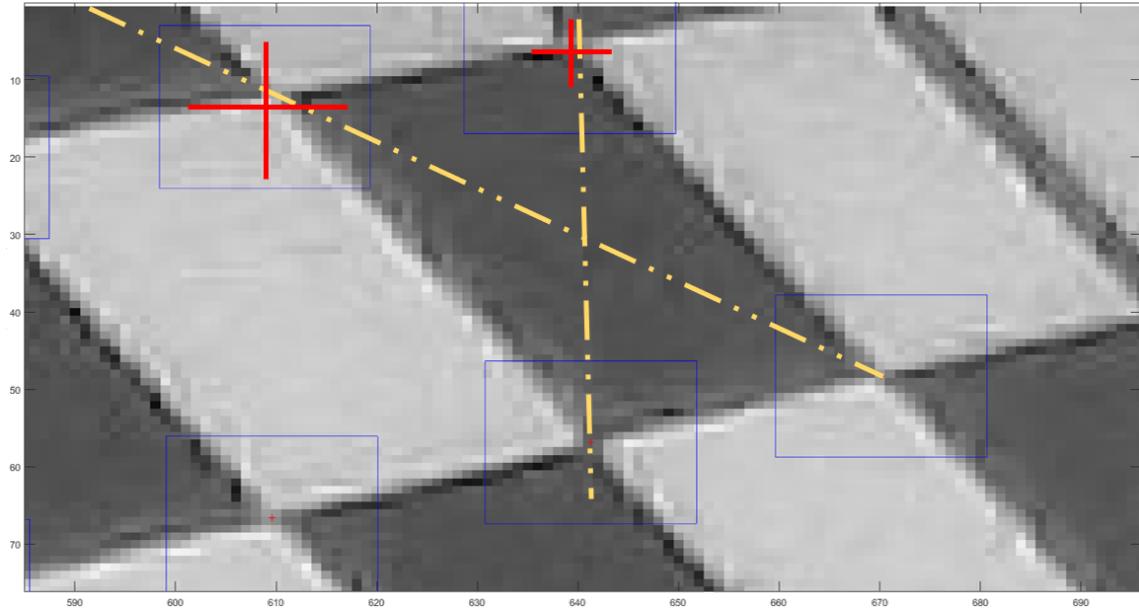


Figura 3.2: Ejemplo de esquinas detectadas con cierto error (cruz roja, aumentada por motivos de visualización). Se puede observar cómo intuitivamente el píxel donde debería situarse la esquina detectada (sobre las líneas amarillas que cruzan la diagonal de los cuadrados) no coincide con la detección.

Donde i representa la imagen i -ésima del patrón de calibración, por lo que el resto de variables de la ecuación están referidas a dicha imagen; x representa el vector que contiene todos los parámetros a estimar y que por tanto lo podemos representar como $x = (fc_i, u_o, v_o, k_i, R_i, t_i)$ y $\Delta m_i(x)$ es el error de reprojeción.

Algo importante a señalar en este punto, es que la detección de las esquinas del patrón de calibración no es perfecta (ver Figura 3.2). Es por esto que m_i es una estimación de la ubicación de las esquinas sujeta a un determinado error en la detección. Este error es común considerarlo como un error Gaussiano de media nula y desviación típica σ ($\mathcal{N}(0, \sigma^2)$) [27, 6, 25].

Por otro lado, W_i se corresponde con la matriz de pesos, la cual es una matriz semi-definida positiva que permite dar más importancia a determinados errores de reprojeción. En la práctica, esta matriz suele considerarse como diagonal y con todos los elementos iguales, representando así que los errores en la detección son iguales e independientes para todas las esquinas [27, 25, 18].

Además, con el objetivo de que tenga un significado estadístico útil, esta matriz se suele elegir como aquella que aproxime a la inversa de la matriz de covarianza

asociada a la detección de las esquinas (m_i). De esta forma, la Ecuación 3.10 se corresponde con el negativo del logaritmo de la estimación de máxima verosimilitud¹.

Sabiendo todo esto, para representar la ecuación 3.10 de manera más compacta, podemos formar un único vector en el que aparezcan todas las coordenadas píxel asociadas a cada esquina de todas las imágenes que hemos utilizado del patrón. De esta forma definimos $m = (m_1^T, \dots, m_n^T)^T$. En cuanto a las matrices de pesos de cada imagen W_i , las podemos agrupar en una matriz diagonal por bloques de la forma $W = \text{diag}(W_1, \dots, W_n)$ resultando en la siguiente ecuación equivalente:

$$f(x) = \frac{1}{2} \Delta m(x)^T W \Delta m(x) \rightarrow \Delta m(x) = m - \hat{m}(x) \quad (3.11)$$

Antes de empezar con las operaciones algebraicas que hay que llevar a cabo en este método, es importante destacar la importancia de la solución cerrada. Ya que este método al igual que otros de optimización no lineal como Levenberg-Marquardt o Newton, necesitan una estimación inicial de los parámetros a estimar, estando la convergencia de estos métodos fuertemente relacionada con lo buena o mala que es dicha estimación previa [25].

Para visualizar esto, se recomienda ver la Figura 3.3, en la que se representa una hipotética función $f(x)$ como la de 3.10 en su eje Z , que viene dada en función de 2 variables X e Y (para poder visualizar en 3D utilizamos solo 2), en la que se observa cómo para dos soluciones cerradas con una diferencia próxima en la estimación de sus 2 variables, podemos llegar a acabar en un mínimo local en vez de uno global de $f(x)$ tras la optimización o proceso iterativo.

Tal y como hemos dicho antes y en relación a la Ecuación 3.11, x recoge a todos los parámetros de los que depende la función $f(x)$ a minimizar, y son los que queremos estimar. Además, como ya hemos introducido antes, los valores que pueden tener este conjunto de parámetros están asociados con un espacio multi-variable no lineal que da como resultado a las coordenadas pixel \hat{m} .

Sin embargo, para realizar la optimización, tenemos que asumir que localmente es posible linealizar este espacio; es decir, ante pequeños cambios en el vector x ,

¹La estimación de máxima verosimilitud de una variable distribuida normalmente establece la media y la varianza más verosímiles dada una muestra de dicha variable. Dicha media y varianza se corresponden con aquellas que consiguen minimizar la expresión (particularizada a nuestro problema) 3.10, siempre y cuando W_i sea la inversa de la matriz de covarianza [25].

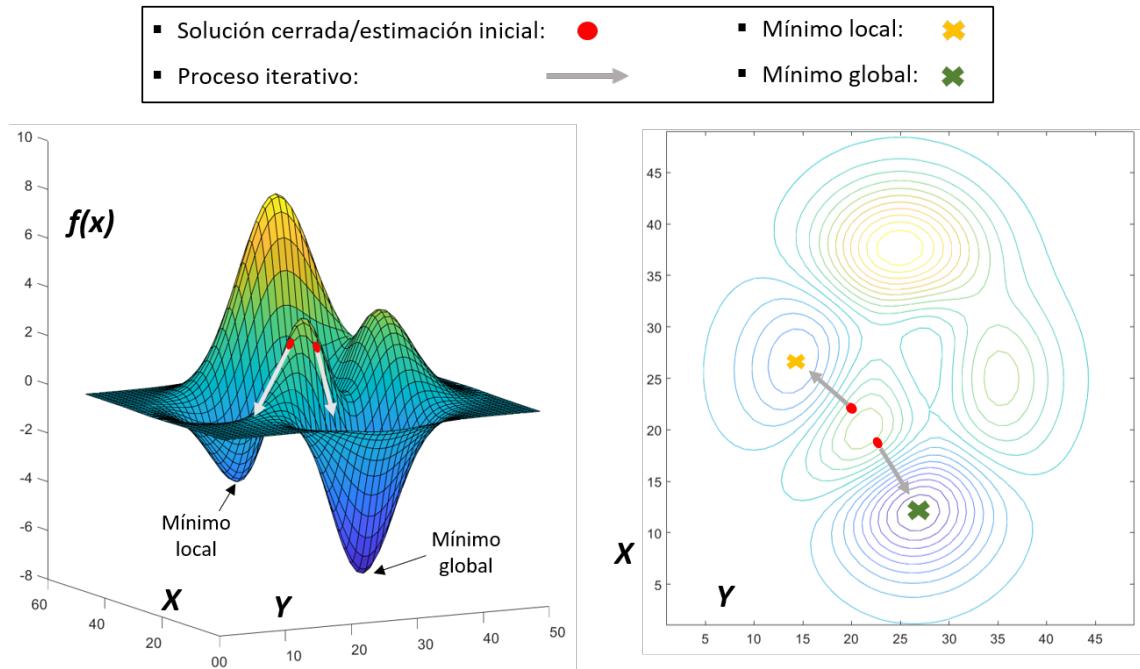


Figura 3.3: Muestra de cómo diferencias en la estimación inicial, pueden llevar a acabar en un mínimo local de $f(x)$ en vez de en uno global.

denominados como δx , la transformación producida en m será localmente lineal. De esta forma, asociándolo a la Ecuación 3.11, donde $f(x)$ es una función cuadrática con m , significa que podremos aproximar $f(x + \delta x)$ con una serie de Taylor de segundo orden²:

$$f(x + \delta x) \approx f(x) + \frac{df}{dx}\delta x + \frac{1}{2}\delta x^T \frac{d^2 f}{dx^2} \delta x \quad (3.12)$$

Donde x representa el valor actual de los parámetros en cada iteración. De esta forma si conseguimos minimizar este modelo local cuadrático, conseguiremos minimizar la Ecuación 3.11 [25].

Además en esta ecuación, aparecen los conceptos de gradiente (g) y matriz hessiana (H) de $f(x)$, muy importantes en este desarrollo ya que, como veremos a continuación, nos permitirán calcular la variación de los parámetros necesaria en cada iteración o δx , con el objetivo de acercar $f(x)$ a su convergencia (alcanzando el mínimo global). Estos términos son:

$$g = \frac{df}{dx}(x) \quad H = \frac{d^2 f}{dx^2}(x)$$

²Recordemos que, una función $f(x)$ aproximada por una serie de Taylor de segundo orden y en el entorno de un punto x_{actual} viene dada por $f(x) = f(x_{actual}) + \frac{df}{dx}(x - x_{actual}) + \frac{1}{2} \frac{d^2 f}{dx^2}(x - x_{actual})^2$. Para escribirlo en su forma incremental, basta con sustituir en esta ecuación el valor $x = x_{actual} + \delta x$ quedándonos finalmente la ecuación 3.12

Ahora, recordando que $f(x) = \frac{1}{2}\Delta m(x)^T W \Delta m(x)$, donde $\Delta m(x) = m - \hat{m}(x)$, y sabiendo por tanto que $\frac{d\Delta m}{dx} = \frac{d\hat{m}}{dx} = J$, matriz jacobiana de las esquinas calculadas; podemos expresar el gradiente y la matriz hessiana igualmente de esta forma:

$$g = \frac{df}{dx}(x) = \Delta m^T W J \quad H = \frac{d^2 f}{dx^2}(x) \approx J^T W J$$

Donde se ha considerado la componente de la matriz hessiana $\frac{d^2 m}{dx^2} \approx 0$ al ser una aproximación propia del método de optimización de Gauss-Newton, algo que es muy común en la optimización por mínimos cuadrados, dado que con las asunciones realizadas $m(x)$ es localmente aproximadamente lineal [25].

Con estas herramientas, ya es posible formular el δx a realizar para alcanzar el mínimo de la función localmente cuadrática $f(x + \delta x)$, consiguiéndolo mediante la diferenciación de 3.12 con respecto a x e igualándola a un vector de ceros:

$$\begin{aligned} \frac{df}{dx}(x + \delta x) &\approx g + H\delta x = 0 \rightarrow \\ \rightarrow \delta x &= -H^{-1}g \rightarrow \delta x = -(J^T W J)^{-1} J^T W \Delta m \end{aligned} \tag{3.13}$$

Algo a destacar de esta expresión es que el δx al venir dado por $-H^{-1}g$, y el gradiente de la función a minimizar, $g(x)$, representar la dirección en la que se produce la variación más rápida de $f(x)$, significa que si H , evaluada en x , es definida positiva³ entonces al aplicar el δx calculado estaremos reduciendo $f(x)$ hasta alcanzar el mínimo global.

Sabiendo esto, podemos reafirmar de nuevo la importancia de la solución cerrada, dado que la condición de que H sea definida positiva se cumplirá si está solución cerrada nos da un valor de x que se encuentra dentro del espacio de $f(x)$ en el que la aproximación de segundo orden en series de Taylor 3.12 es aproximadamente válida [25].

Dicho desde un punto de vista geométrico, si H es definida positiva en el entorno del espacio en el que la aproximación 3.12 es válida, $f(x)$ localmente vendrá dada por una función cuadrática con un único mínimo [25] como la de la Figura 3.4.

³Si H evaluada en un cierto valor de los parámetros (x) es definida positiva, significa que para cualquier vector g , $g^T H^{-1} g > 0$, lo que implica que $H^{-1}g$ no transforma la dirección de g un ángulo superior a 90°, siendo por tanto δx una variación de los parámetros que reduce a $f(x)$.

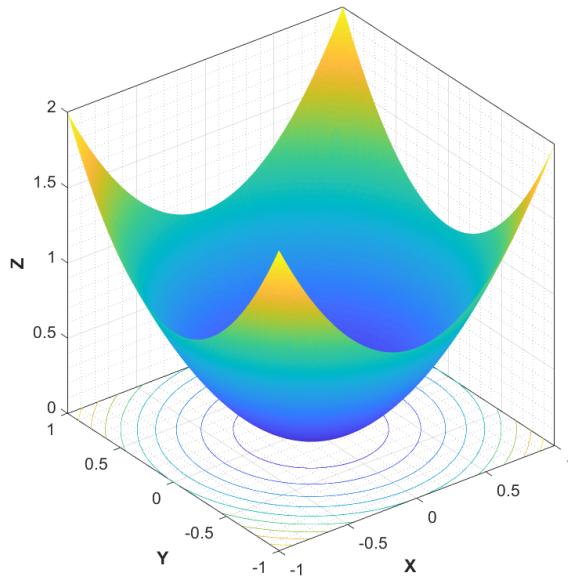


Figura 3.4: Ejemplo de función cuadrática ($Z = X^2 + Y^2$) que cumple la aproximación 3.12.

3.3.1. Interpretación probabilista de la matriz hessiana

Un aspecto que será de gran utilidad en la evaluación de las simulaciones realizadas en el siguiente capítulo, será la *incertidumbre* asociada a ellas.

En nuestro problema, al estar utilizando el método de optimización de Gauss-Newton, la incertidumbre podrá ser calculada a partir de la matriz hessiana de $f(x)$, que recordemos que viene dada por $H = \frac{d^2f}{dx^2} \approx J^T W J$, donde J es la matriz jacobiana asociada al cálculo de las esquinas y que será presentada con más detalle en el siguiente capítulo.

Como hemos introducido ya antes, al considerar un ruido gaussiano en la detección de las esquinas, la expresión 3.11 representa el negativo del logaritmo de la estimación de máxima verosimilitud ($\ell(x)$):

$$f(x) = -\ell(x) = -\sum_i^n \log(g(m_i|x))$$

Donde se ha utilizado una notación similar a la de [14], en la que g representa la función que nos permite obtener las coordenadas de las esquinas que vienen dadas por las ecs. 2.8 y 2.9.

Con esto, es posible definir la matriz de información de Fisher *observada* ($I(x)$)

[14] como:

$$I(x) = -\frac{d^2}{dx^2}\ell(x) = H(x)$$

Que, como vemos coincide con la matriz hessiana. Ahora, sabiendo que la inversa de la matriz de información de Fisher, se corresponde con la matriz de covarianza de la estimación de los parámetros (x) [14], y que esta matriz converge asintóticamente a $I(x)$ [25], podemos aproximar la varianza en la estimación de los parámetros por:

$$Var(x) = diag(H^{-1}) \approx diag((J^T W J)^{-1})$$

A partir de esto, es posible definir la incertidumbre en la estimación de los parámetros como $\pm 3 Var(x)^{\frac{1}{2}} = \pm 3 \sigma_i$, por tanto, esta incertidumbre añadida a la estimación de cada parámetro x_i representa una región en la que el verdadero valor se situará el 99,7% de las veces.

De esta forma es tal y como está implementada la obtención de las incertidumbres asociadas a los parámetros en [4], cuyos resultados serán utilizados de cara a la simulación y que gracias a este contexto podemos saber de dónde viene dicho cálculo.

Como conclusión, iterar el δx de 3.13, es lo que se conoce como el método de Gauss-Newton, del que hemos presentado sus aspectos fundamentales en esta sección y damos por terminado su desarrollo. Gracias a él, como ya hemos dicho, es posible mejorar la solución cerrada obtenida previamente, algo que es lógico ya que es aquí cuando el modelo de distorsión radial ha sido introducido.

De esta forma, damos por terminada la explicación de los aspectos fundamentales detrás de uno de los procesos de calibración más utilizados actualmente; [18, 3, 19, 15] son solo algunos de los ejemplos de investigaciones recientes que utilizan las mismas bases de este método.

Capítulo 4

SIMULACION

En este capítulo se van a mostrar las simulaciones realizadas con el objetivo de alcanzar el objetivo propuesto, es decir, encontrar un procedimiento general de selección de posiciones y orientaciones del patrón de calibración de tal forma que podamos superar la precisión de una calibración estándar así como la de los métodos ya establecidos en el estado del arte.

En una primera parte introduciremos el entorno de trabajo desarrollado, el cual nos permitirá obtener resultados de manera visual y analítica. En una segunda parte se analizarán los resultados. Finalmente, se comparará el método desarrollado con los del estado del arte.

4.1. Entorno de trabajo desarrollado

El software utilizado para la creación del entorno de trabajo ha sido *MATLAB* [13], junto a la *toolbox*¹ de calibración de una cámara desarrollada por el investigador Jean-Yves Bouguet [4], en la cual se implementan las ecuaciones desarrolladas en los capítulos 2 y 3. Dicha *toolbox* contiene los algoritmos para una calibración estándar de la cámara. En este TFG hemos realizado numerosas modificaciones sobre dicha *toolbox* para alcanzar nuestros objetivos.

¹Toolbox es la palabra utilizada para referirse a los algoritmos destinados a un campo o rama científica en concreto, los cuales se ponen a disposición de terceros ya sea gratuitamente o mediante su compra.

Con el entorno de trabajo que hemos desarrollado somos capaces de crear de manera automática patrones de calibración sintéticos o virtuales (creando hipotéticas coordenadas de esquinas), hasta más de 1 millón de ellos, y llevar a cabo más de 1000 calibraciones con un tiempo de computación razonable.

La generación de los patrones siguen el procedimiento mostrado en el Algoritmo 1. Como se puede observar en él, se generan múltiples traslaciones para una rotación dada del patrón, siempre que se cumpla la condición de que es visible en la cámara. Para observar distintos parámetros que se han definido en él visualmente, se recomienda ver la Figura 4.1.

Algoritmo 1: Generación de patrones de calibración

```

1: patrongeom {geometría del patrón de calibración}
2: cam {parámetros intrínsecos de la cámara a simular}
3: distxy {distancia entre 2 tableros contiguos con mismo alejamiento}
4: distz {diferencia de alejamiento entre tableros}
5: Δang {incremento de giro angular en un eje (grados sexagesimales)}
6: for girox = 0 to 359 con paso = Δang do
7:   for giroy = -89 to 90 con paso = Δang do
8:     for giroz = 0 to 359 con paso = Δang do
9:        $R_{\text{patron}} = \text{Rot}_x(giro_x)\text{Rot}_y(giro_y)\text{Rot}_z(giro_z)$ 
10:      Patron_visible = Test_visibilidad(cam, patrongeom, Rpatron)
11:      if Patron_visible then
12:        Tpatron = Crear_traslacion(cam, patrongeom, Rpatron, distxy, distz)
13:      end if
14:    end for
15:  end for
16: end for

```

En cuanto al test de visibilidad, se impuso en primer lugar que la normal del patrón expresada desde la referencia cámara, debía tener una componente negativa en el eje Z_c ; de esta forma, se consiguen eliminar situaciones como las de la Figura 4.2.

Para completar esta condición y evitar casos como los de la Figura 4.3 que surgen debido a la transformación perspectiva realizada por la cámara, se impuso también

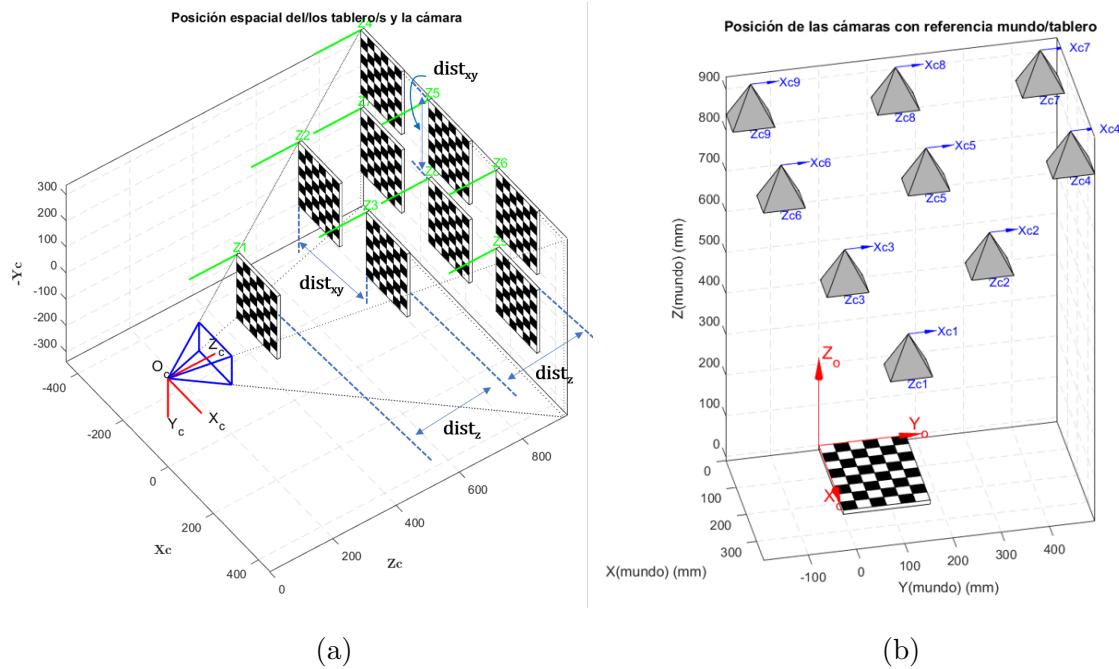


Figura 4.1: Vista 3D del proceso de generación de patrones para una única rotación. **a:** si consideramos que cambia la posición de los tableros. **b:** equivalencia a si consideramos un cambio de posición en la cámara.

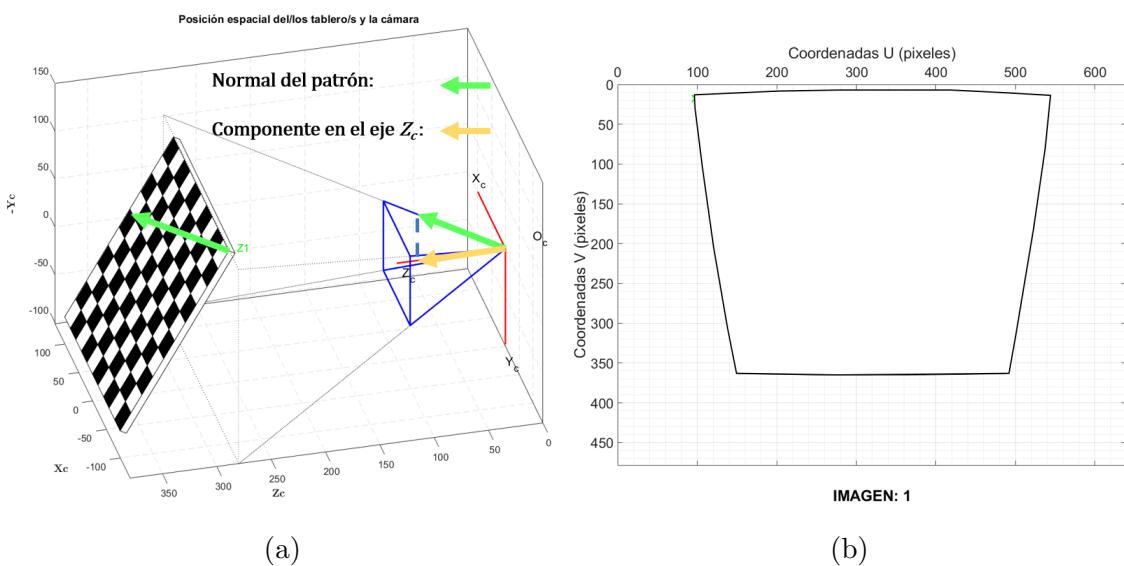


Figura 4.2: Ejemplo de patrón no visible debido a que su normal tiene componente positiva en el eje Z_c de la cámara. **a:** vista 3D. **b:** Imagen que capturaría la cámara.

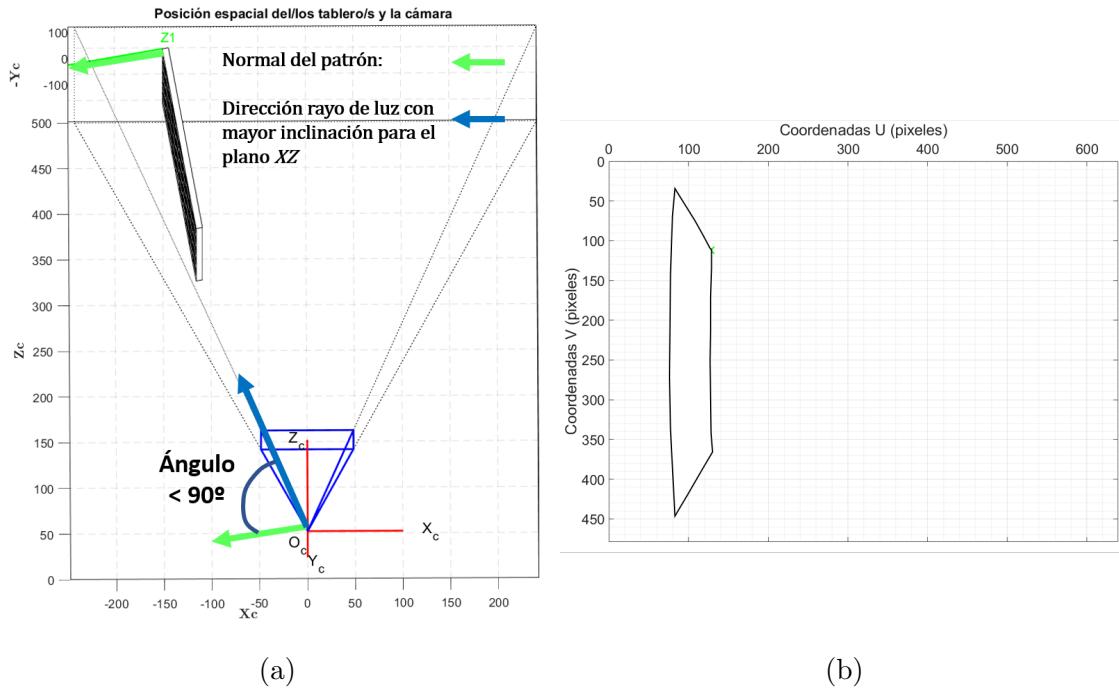


Figura 4.3: Ejemplo de patrón no visible debido a que su normal forma un ángulo inferior a 90° con la dirección que marcan los rayos de luz entrantes de mayor inclinación en plano XZ de la cámara. **a:** vista 3D. **b:** Imagen que capturaría la cámara.

la siguiente condición: Si la normal del patrón tenía proyección sobre los planos X_cZ_c e Y_cZ_c , dicha proyección debe formar un ángulo mayor a 90° con la dirección de los rayos que entran con una mayor inclinación por dichos planos y que coinciden con los límites del ángulo de visión de la cámara. Si no se cumple esta condición, no se crean traslaciones que ubiquen a la proyección del patrón en la mitad del plano X_cZ_c o Y_cZ_c asociado al rayo de luz comentado.

En cuanto al criterio de creación de las rotaciones del Algoritmo 1, se diseñó para que se pudiesen crear todas las rotaciones posibles con el mínimo coste computacional; es decir, evitando repetir rotaciones equivalentes que se pueden producir por diferentes combinaciones de giros en los ejes.

En nuestro caso, al utilizar la secuencia de rotaciones $Rot_x(\alpha)Rot_y(\beta)Rot_z(\gamma)$, no solo existe el conjunto de rotaciones equivalentes que son fruto de sumar al ángulo de giro un múltiplo entero de 360° y que viene dado por $Rot_x(\alpha + 360k)Rot_y(\beta + 360k)Rot_z(\gamma + 360k)$, sino que también existe el conjunto de rotaciones equivalentes $Rot_x(\alpha+180)Rot_y(180-\beta)Rot_z(\gamma+180)$. Sabiendo que la equivalencia de un ángulo

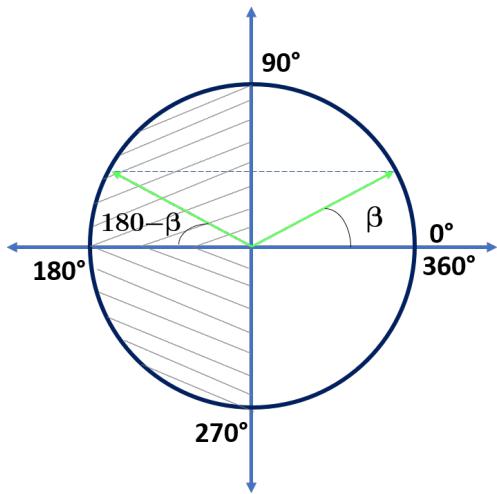


Figura 4.4: Círculo que muestra la zona, sombreada, en la que se encuentran los ángulos equivalentes a un ángulo $\beta \in (-90^\circ, 90^\circ]$



Figura 4.5: Tablero utilizado en la mayoría de los análisis

$\beta \in (-90^\circ, 90^\circ]$ asociado a Rot_y , se encuentra en la zona sombreada del círculo de la Figura 4.4, podemos establecer un giro cualquiera en los otros dos ejes y obtener una matriz de rotación única, siempre y cuando β no pertenezca a dicha zona sombreada.

Por último, para la función asociada a la creación de las traslaciones, se adoptó el criterio de calcular el alejamiento mínimo para que el patrón sea totalmente visible en la cámara, y partir de esta localización ir aumentando el alejamiento según $dist_z$ a la vez que se sitúan los patrones que pudiesen entrar en dicho ángulo de visión en base a $dist_{xy}$.

Este cálculo se realizó en base a la geometría del patrón y las características de la cámara, centrándose principalmente en los límites establecidos por el campo de visión. Debido a su extensión y mayor complejidad, si el lector está interesado en la función, puede dirigirse al anexo A donde se encuentra el código con comentarios detallados.

	Parámetros	Unidades
<i>Resolución</i>	Horiz. (nx)	640
	Vert. (ny)	480
<i>Distancia focal</i>	fc1	660
	fc2	660
<i>Punto principal</i>	cc1 (u_o)	319.5
	cc2 (v_o)	239.5
<i>Coef. de distorsión</i>	kc1	mm^{-2}
	kc2	mm^{-4}

Tabla 4.1: Modelo de cámara utilizado en la mayoría de los análisis realizados

4.2. Importancia de la colocación del patrón

Para mostrar la importancia que tiene la colocación de un patrón en relación al error de la calibración, se realizó un análisis consistente en 600 calibraciones formadas por 30 imágenes cada una, con una elección aleatoria de los tableros creados mediante el Algoritmo 1.

Inspirados por el modelo de cámara utilizado en la documentación de [4] y con el objetivo de simular con un modelo de cámara real, se utilizaron unos parámetros muy similares a los obtenidos por J. Yves-Bouquet en esa calibración de ejemplo, los cuales aparecen recogidos en la tabla 4.1.

En cuanto al patrón de calibración se utilizó de nuevo el de la documentación de J. Yves-Bouquet (ver Figura 4.5), es decir, un tablero de 11×12 cuadrados en los ejes X_{mundo} e Y_{mundo} respectivamente, y de un tamaño cada uno de $dX = dY = 30mm$. El error en la detección de sus esquinas tal y como se ha introducido en la sección 3.3 se consideró gaussiano con media nula y con desviación estandar de 1 píxel (aceptable para la gran mayoría de cámaras [27]).

Los resultados aparecen en la Figura 4.6, en la que se muestra el error en la estimación de los parámetros de la cámara conforme se van añadiendo imágenes a la calibración. Cada línea representa una única calibración, por lo que al representar todas en una misma gráfica, se puede observar entre qué valores máximos y mínimos puede estar el error así como su frecuencia.

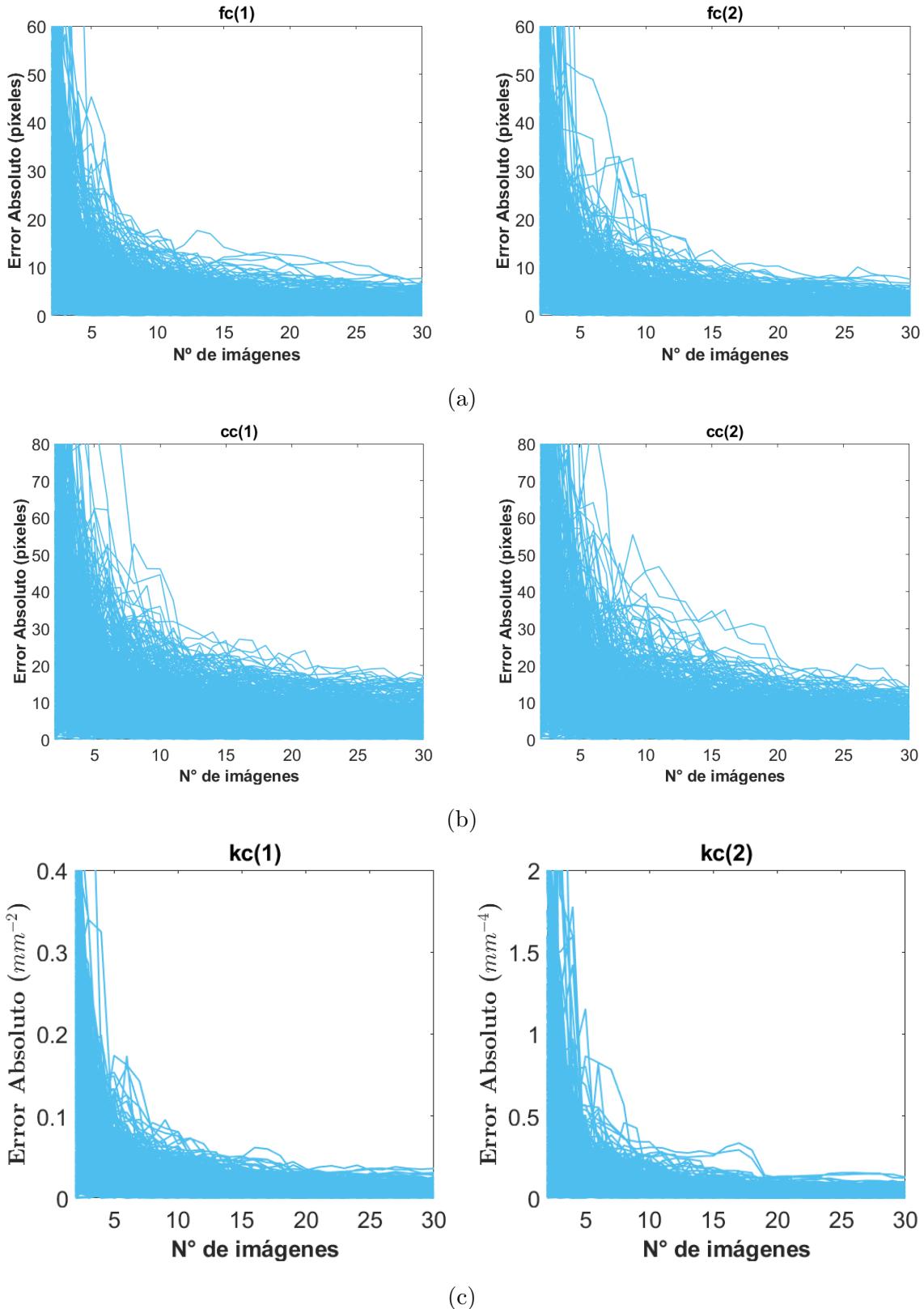


Figura 4.6: Error en la estimación de 600 calibraciones (cada una representa una línea). Nótese la gran variación del error, debida a las distintas posiciones de cámara de las calibraciones. **a:** Distancia focal (f_{c_i}). **b:** Punto principal (cc_i). **c:** Distorsión (kc_i).

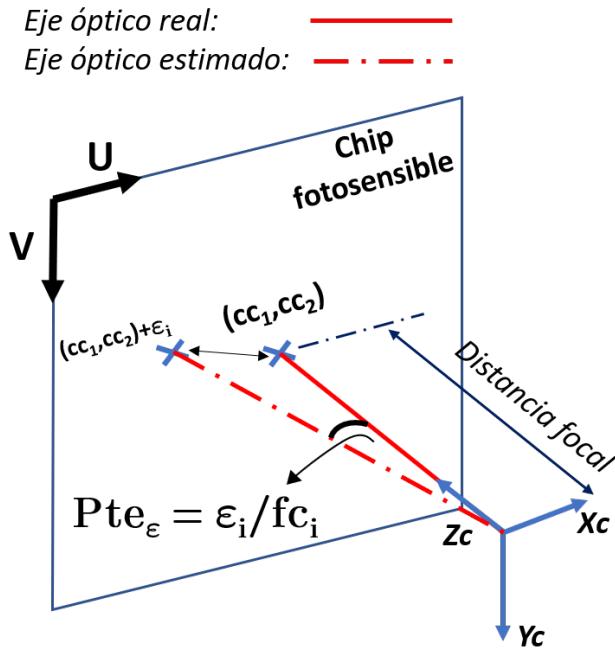


Figura 4.7: Significado geométrico del error $\left| \frac{cc_{real} - cc_{estim}}{fc} \right|$

Como conclusión, los errores de distintas calibraciones tienen un rango de variación muy amplio. Esto justifica la intención de establecer un procedimiento de orientación y traslación de los tableros que garantice errores bajos.

4.3. Sensibilidad con el número de imágenes

Este análisis se plantea como una extensión del anterior, en el que se muestra el promedio de los errores relativos en la estimación de los parámetros de las 600 calibraciones anteriores en función del número de imágenes utilizado.

En todos los parámetros se ha utilizado la definición normal de error relativo: $error_{rel} = \left| \frac{param_{real} - param_{estimado}}{param_{real}} \right|$ excepto en el caso del punto principal (cc_i) debido a que, tal y como señalan los trabajos [27, 24], su error no tiene una importancia significativa; mientras que si se opta por utilizar en el denominador la distancia focal, esto le aporta un significado geométrico al representar la pendiente entre el eje óptico calculado y el eje óptico real (ver Figura 4.7). Los resultados se presentan en la Figura 4.8.

Al observar estos resultados, se concluye que conforme el número de imágenes

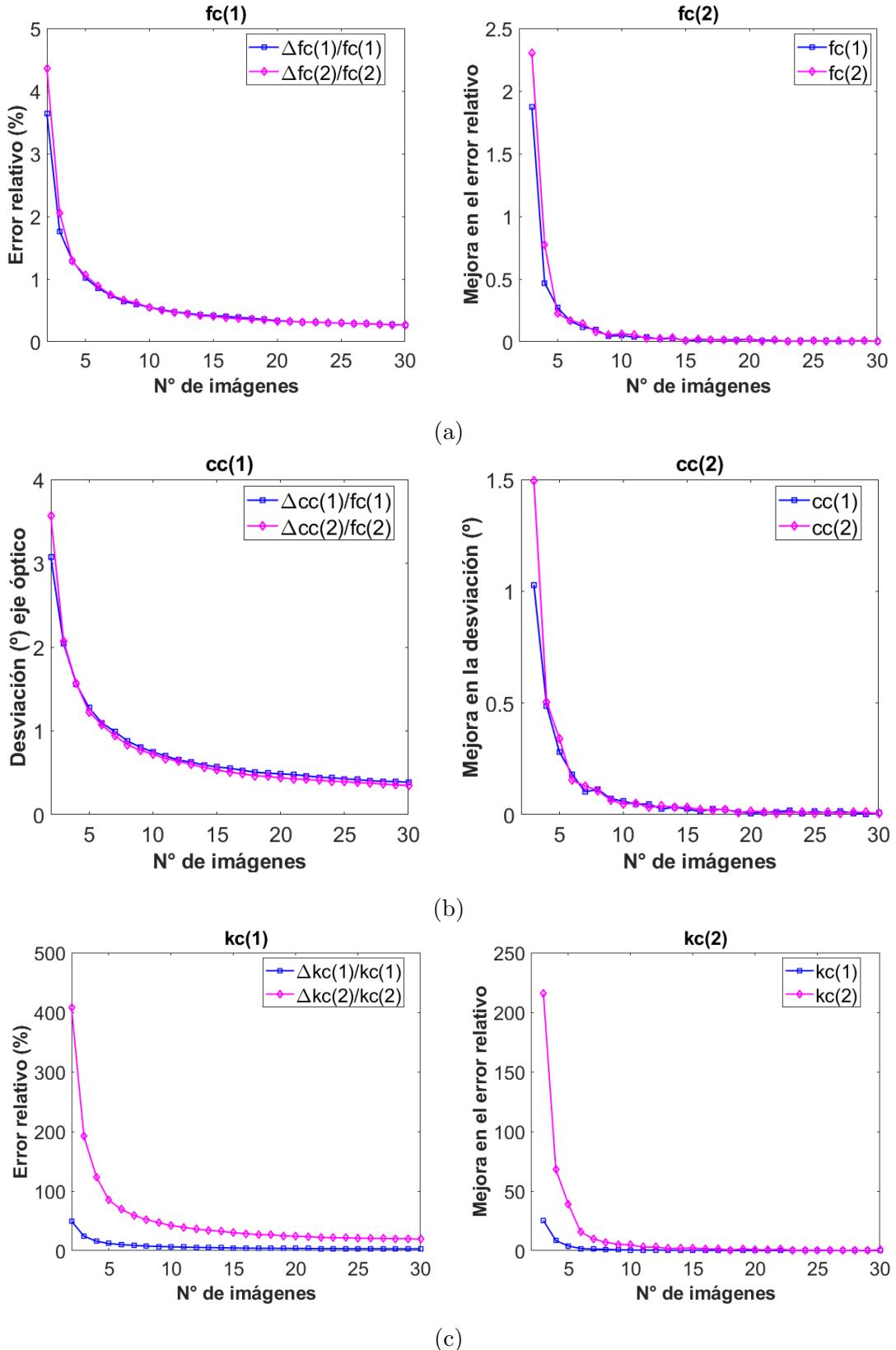


Figura 4.8: Errores relativos promedio de los parámetros conforme se añaden imágenes (izq.) y mejora en el error relativo por adición de una imagen (dcha.) **a:** Distancia focal (fc_i). **b:** Punto principal (cc_i). **c:** Distorsión (kc_i).

Parámetros	a)	b)	Unidades
<i>Resolución</i>	Horiz. (nx)	3264	320
	Vert. (ny)	2448	240
<i>Distancia focal</i>	fc1	2244	300
	fc2	2444	300
<i>Punto principal</i>	cc1 (u_o)	1631.5	159.5
	cc2 (v_o)	1223.5	120.5
<i>Coef. de distorsión</i>	kc1	-0.146	mm^{-2}
	kc2	-0.015	mm^{-4}

Tabla 4.2: Modelos de cámara que se añaden al análisis de influencia de la distancia

aumenta, la precisión en la estimación de los parámetros también lo hace. Sin embargo, hay que tener en cuenta que al añadir más imágenes, el proceso de calibración se hace más largo y eso puede llegar a ser un inconveniente en aquellas situaciones en las que se requiera calibrar un número considerable de cámaras o en las que no se disponga de mucho tiempo. Por estas razones hay que intentar conseguir unos resultados precisos con un número de imágenes aceptable.

Por otro lado, tal y como se observa en las gráficas, en general a partir de haber utilizado 20 imágenes, las mejoras en el error se aproximan a un valor nulo. Es por esto por lo que en los siguientes análisis se consideró la utilización de 20 imágenes por calibración.

4.4. Sensibilidad con la distancia

En este análisis se estudia cual es la influencia del alejamiento del patrón de calibración de la cámara; es decir, cómo la ubicación del patrón con respecto al eje Z_c de la cámara afecta a los resultados. En él, además de utilizar el mismo modelo que en los análisis anteriores y con el objetivo de reafirmar las conclusiones extraídas, se evaluaron otros dos modelos de cámara que aparecen en la tabla 4.2.

El análisis consistió en establecer 3 rangos de alejamientos en los que se realizaron 100 calibraciones de 20 imágenes cada una. De esta forma se llevaron a cabo un total de 900 calibraciones teniendo en cuenta los 3 modelos de cámara evaluados.

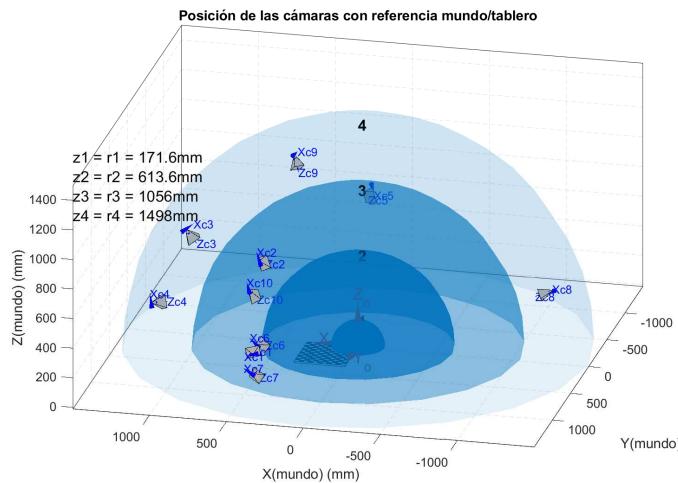


Figura 4.9: Punto de vista geométrico del análisis. Las esferas representan los tramos de alejamiento en los que se puede situar la cámara, cuyos radios aparecen en la imagen. En este caso, 10 cámaras aparecen ubicadas en el tramo 3.

Los tres rangos de alejamiento fueron los siguientes:

- *Tramo 1*: es el más próximo a la cámara, comprendido entre el mínimo alejamiento posible para que un patrón pudiese entrar en el ángulo de visión de la cámara y $1/3$ del alejamiento máximo, que consideraremos como 1,5m.
- *Tramo 2*: comprendido entre $1/3$ y $2/3$ del alejamiento máximo analizado.
- *Tramo 3*: comprendido entre $2/3$ y el alejamiento máximo.

Desde un punto de vista geométrico, si se considera que la cámara cambia de ubicación y no el patrón, esto equivale a situarla entre 2 esferas cuyos radios vienen determinados por los tramos de alejamiento como muestra la Figura 4.9.

Un primer resultado del análisis, para el modelo de cámara de la tabla 4.1, se puede observar en la Figura 4.10, en la que se muestra el error obtenido en todos los parámetros para las 100 calibraciones realizadas (cada una representa una línea) en cada rango de alejamiento. Se puede observar que los mejores resultados se obtienen para los menores valores de la coordenada Z_c del patrón, que corresponden a las dos columnas más a la izquierda en la figura.

Los valores promedio del error para estas calibraciones se muestran en la Figura 4.11. Esta es otra forma de verificar que conforme la distancia o alejamiento en Z_c disminuye, los resultados en la calibración mejoran al presentar un menor error.

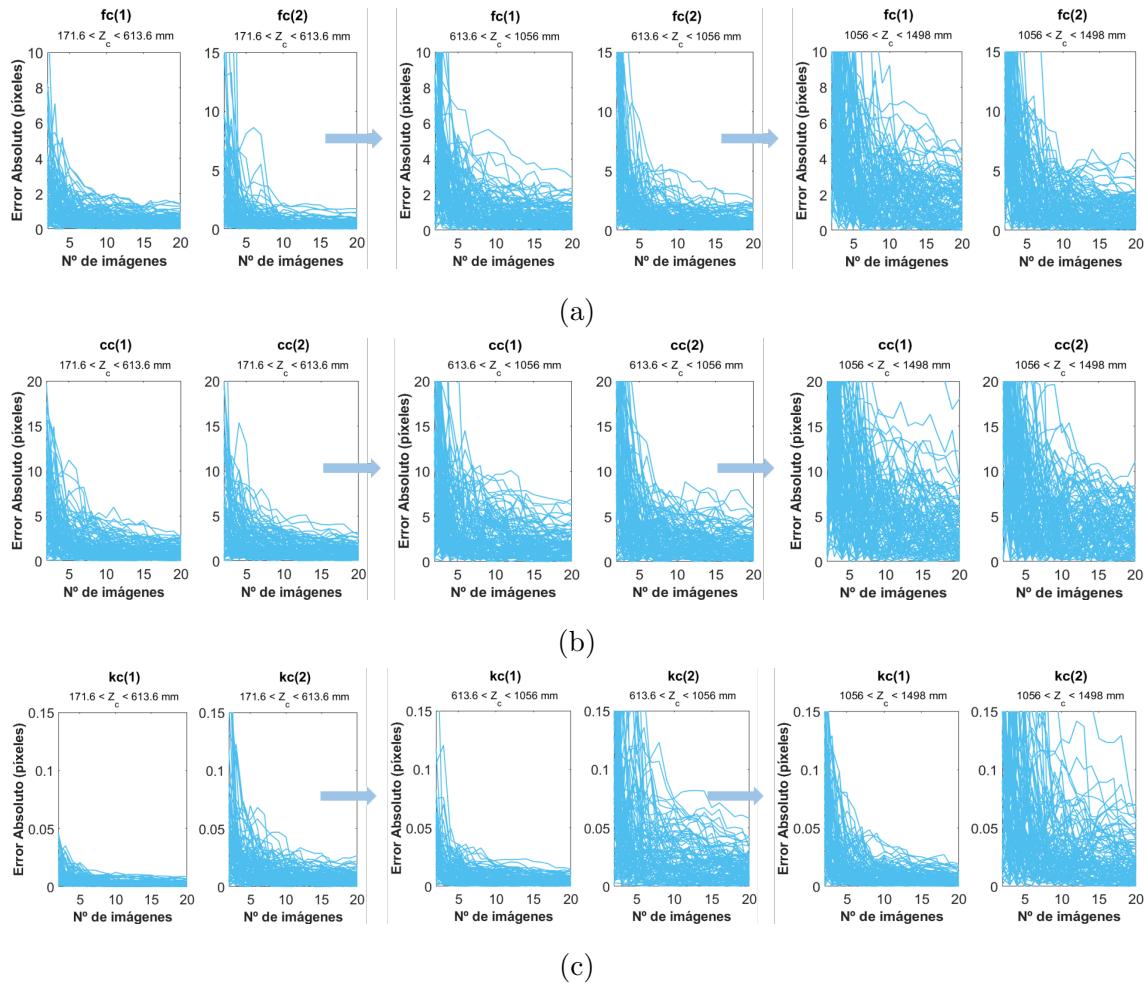


Figura 4.10: Error en cada una de las 100 calibraciones realizadas en cada tramo. Los tramos aparecen en el título de cada figura, incrementando el alejamiento de izquierda a derecha. **a:** Distancia focal (fc_i). **b:** Punto principal (cc_i). **c:** (kc_i).

Esta tendencia también se obtuvo en las otras dos cámaras evaluadas, como muestra de ello se adjuntan en la Figura 4.12 los valores promedio del error de la distancia focal fc_i para ambos modelos.

Otra forma de evaluar la precisión de una calibración es en base a la incertidumbre asociada a la estimación de los parámetros y que ya ha sido introducida en la sección 3.3.1. Por ello, durante este análisis también se recogieron los datos asociados a la incertidumbre de los parámetros.

Los datos de incertidumbre que mostraremos de aquí en adelante, representan la mitad de todo el rango de incertidumbre. Es decir, si el valor estimado de la primera componente de la distancia focal es $fc_1 = \hat{fc}_1 \pm k$ píxeles, nosotros haremos referencia a ese valor de k en vez de a $2k$.

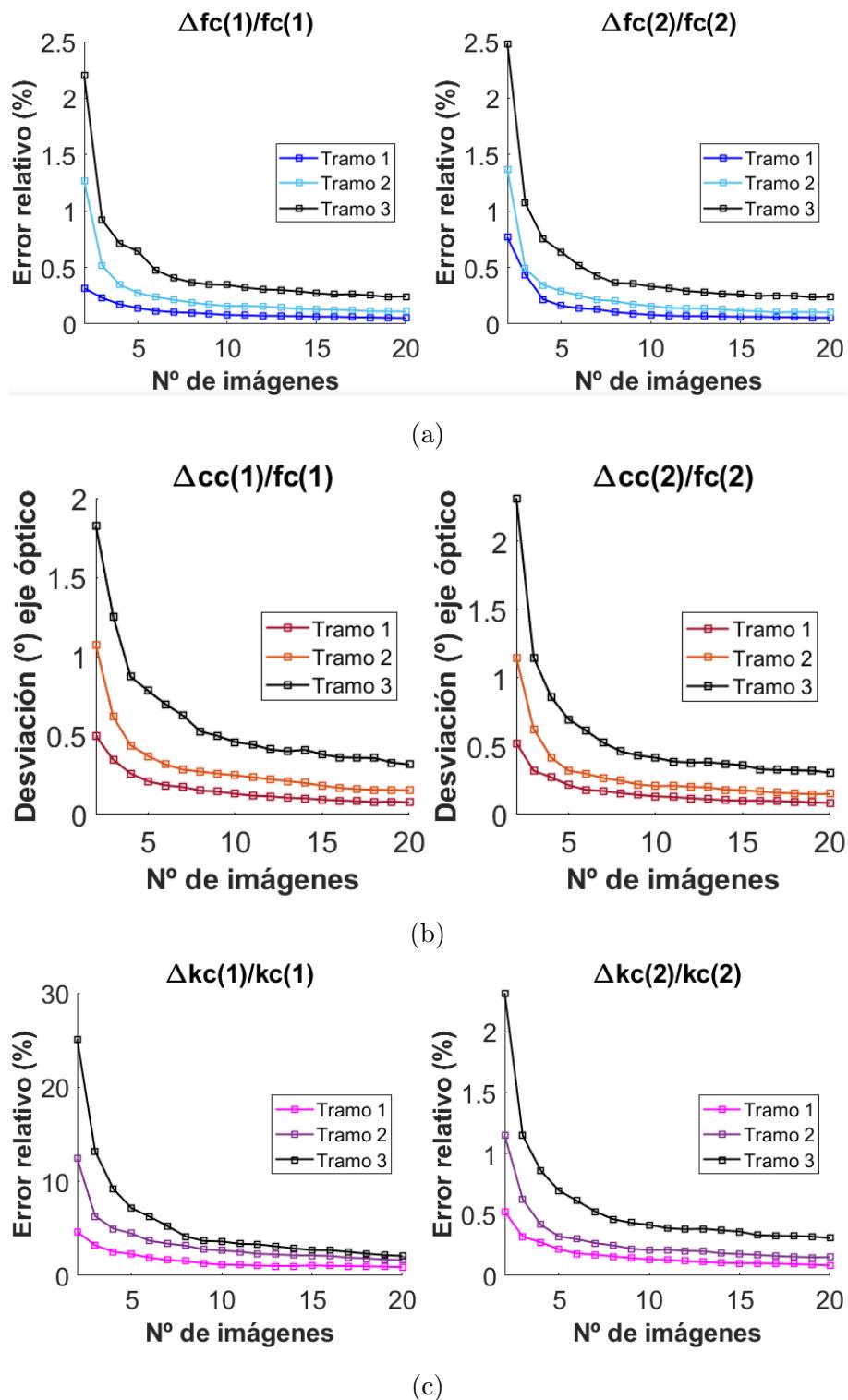


Figura 4.11: Valores promedio del error en los 3 tramos de alejamiento evaluados para el modelo de cámara de la tabla 4.1. **a:** Distancia focal (fc_i). **b:** Punto principal (cc_i). **c:** Distorsión (kc_i).

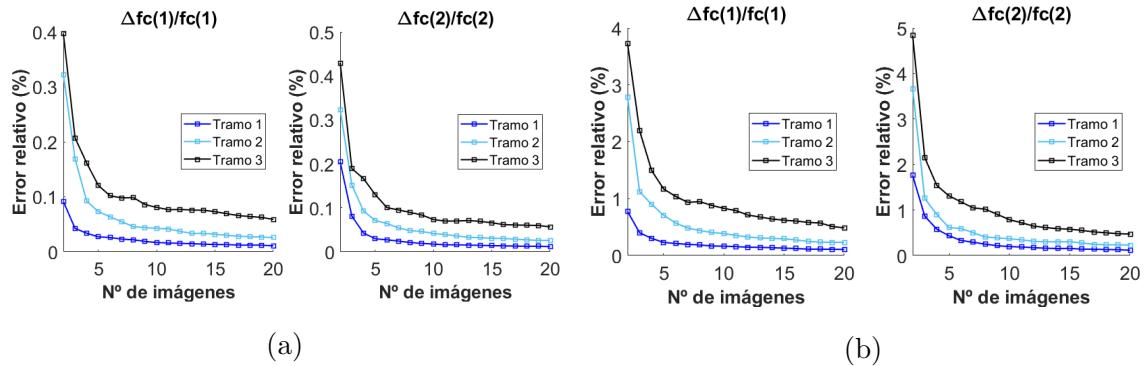


Figura 4.12: Valores promedio del error en la estimación de f_{ci} en los 3 tramos de alejamiento para los modelos de cámara de la tabla 4.2. **a:** modelo a). **b:** modelo b).

En la gráfica 4.13 se muestra la suma de las incertidumbres asociadas a la distancia focal (f_{ci}) y al punto principal (cc_i) (ambas en píxeles) de los tres modelos de cámara. De nuevo se observa que en el tramo más cercano a la cámara se obtienen unos mejores resultados al ser inferiores los niveles de incertidumbre.

4.4.1. Justificación teórica

Para justificar teóricamente estos resultados, podemos obtener analíticamente los elementos de la traza de la matriz hessiana, introducida en la sección 3.3 y tratada con más detalle en la sub-sección 3.3.1. Esta matriz viene dada por $H = J^T W J$, donde $J = \frac{d\hat{n}}{dx}$ es la matriz jacobiana de las coordenadas estimadas de las esquinas del patrón de calibración.

Para su cálculo consideraremos un modelo sin distorsión radial, ya que esto facilitará el entendimiento de los resultados. Esta simplificación no afectará significativamente al análisis, dado que los coeficientes de distorsión se utilizan para corregir puntos en el plano imagen. Dicha corrección únicamente se basa en información del plano imagen y no en información del entorno (influencia con la distancia a la cámara), mientras que la distancia focal es la que ubica los puntos del entorno en el plano imagen en función de su ubicación relativa a la cámara en 3D.

De esta forma, definiendo una matriz de rotación genérica del patrón de calibración $R = Rot_x(\theta)Rot_y(\beta)Rot_z(\alpha)$, y representando las operaciones seno y coseno

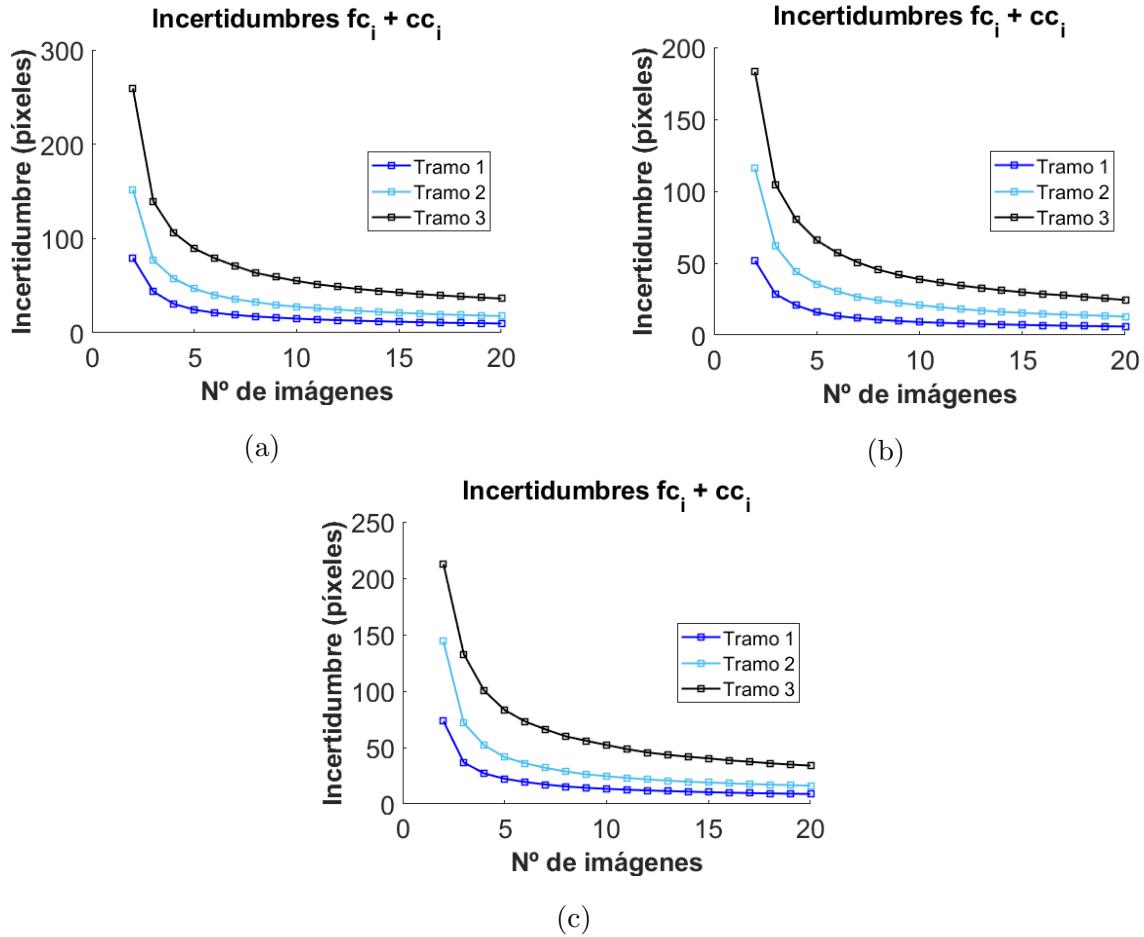


Figura 4.13: Valores promedio de la suma de incertidumbres asociadas a fc_i y cc_i en los 3 tramos de alejamiento. Modelos de cámara: **a:** tabla 4.1, **b** y **c:** tabla 4.2 modelos a) y b) respectivamente.

por S y C respectivamente, tenemos:

$$R = \begin{pmatrix} C(\alpha)C(\beta) & -C(\beta)S(\alpha) & S(\beta) \\ S(\alpha)C(\theta) + C(\alpha)S(\beta)S(\theta) & C(\alpha)C(\theta) - S(\alpha)S(\beta)S(\theta) & -C(\beta)S(\theta) \\ S(\alpha)S(\theta) - C(\alpha)S(\beta)C(\theta) & C(\alpha)S(\theta) + S(\alpha)S(\beta)C(\theta) & C(\beta)C(\theta) \end{pmatrix}$$

Ahora, definiendo una traslación del patrón de calibración genérica $t = (X_c, Y_c, Z_c)^T$ y unas coordenadas genéricas de una esquina desde la referencia mundo ubicada en el patrón $\tilde{M} = (X, Y, 0, 1)^T$, las coordenadas normalizadas (considerando la distancia focal de 1 unidad), y que habíamos denominado como \tilde{x}_d , vienen dadas por:

$$\tilde{x}_d = \begin{pmatrix} R & t \end{pmatrix} \tilde{M} \rightarrow x_d = \frac{1}{\tilde{x}_d(3)} \begin{pmatrix} \tilde{x}_d(1) \\ \tilde{x}_d(2) \end{pmatrix}$$

Por lo que las coordenadas en píxeles \hat{m} son:

$$\hat{m} = \begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} x_d(1)fc_1 + cc_1 \\ x_d(2)fc_2 + cc_2 \end{pmatrix}$$

Recordemos que la matriz jacobiana viene dada por $J = \frac{d\hat{m}}{dx}$, siendo x los parámetros a estimar, por lo que si definimos las siguientes sub-matrices asociadas a una esquina i de la imagen del patrón j :

$$A_{i,j} = \begin{pmatrix} \frac{\partial u}{\partial fc_1} & \frac{\partial u}{\partial fc_2} & \frac{\partial u}{\partial cc_1} & \frac{\partial u}{\partial cc_2} \\ \frac{\partial v}{\partial fc_1} & \frac{\partial v}{\partial fc_2} & \frac{\partial v}{\partial cc_1} & \frac{\partial v}{\partial cc_2} \end{pmatrix} \quad B_{i,j} = \begin{pmatrix} \frac{\partial u}{\partial \theta} & \frac{\partial u}{\partial \beta} & \frac{\partial u}{\partial \alpha} & \frac{\partial u}{\partial X_c} & \frac{\partial u}{\partial Y_c} & \frac{\partial u}{\partial Z_c} \\ \frac{\partial v}{\partial \theta} & \frac{\partial v}{\partial \beta} & \frac{\partial v}{\partial \alpha} & \frac{\partial v}{\partial X_c} & \frac{\partial v}{\partial Y_c} & \frac{\partial v}{\partial Z_c} \end{pmatrix}$$

podemos expresar la matriz jacobiana asociada al cálculo de las coordenadas de esta esquina de la siguiente forma:

$$J_{i,j} = \begin{pmatrix} A_{i,j} & B_{i,j} \end{pmatrix}$$

Aplicando esto para todas las esquinas ne del patrón de calibración que aparece en la imagen j tenemos:

$$J_j = \begin{pmatrix} A_j & B_j \end{pmatrix} = \begin{pmatrix} A_{1,j} & B_{1,j} \\ A_{2,j} & B_{2,j} \\ \vdots & \vdots \\ A_{ne,j} & B_{ne,j} \end{pmatrix}$$

Finalmente, generalizando para un número de imágenes ni , la matriz jacobiana del sistema nos queda de la siguiente forma:

$$J = \begin{pmatrix} A_1 & B_1 & 0 & \dots & 0 \\ A_2 & 0 & B_2 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ A_{ni} & 0 & 0 & \dots & B_{ni} \end{pmatrix}$$

A partir de esto, ya es posible obtener la expresión analítica de la matriz hessiana

na²:

$$I(x) = H(x) = J^T J = \begin{pmatrix} \sum_{j=1}^{ni} A_j^T A_j & A_1^T B_1 & A_2^T B_2 & \dots & A_{ni}^T B_{ni} \\ B_1^T A_1 & B_1^T B_1 & 0 & \dots & 0 \\ B_2^T A_2 & 0 & B_2^T B_2 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ B_{ni}^T A_{ni} & 0 & 0 & \dots & B_{ni}^T B_{ni} \end{pmatrix}$$

Donde $I(x)$ es la matriz de información de Fisher observada (ver sección 3.3.1).

Como nos estamos centrándo en la incertidumbre de los parámetros intrínsecos (fc, cc), consideraremos que las componentes relativas a la información aportada por las distintas imágenes del patrón de calibración en cuanto a estos parámetros, quedan recogidas en el bloque matricial $\sum_{i=1}^{ni} A_i^T A_i$, más concretamente en la traza de dicha matriz.

Esto se aprecia mejor si mostramos la traza en forma de derivadas parciales y particularizada para una imagen j :

$$diag(A_j^T A_j) = \begin{pmatrix} \left(\frac{\partial \hat{m}_1(1)}{\partial fc_1} \right)^2 + \left(\frac{\partial \hat{m}_1(2)}{\partial fc_1} \right)^2 + \dots + \left(\frac{\partial \hat{m}_{ne}(1)}{\partial fc_1} \right)^2 + \left(\frac{\partial \hat{m}_{ne}(2)}{\partial fc_1} \right)^2 \\ \left(\frac{\partial \hat{m}_1(1)}{\partial fc_2} \right)^2 + \left(\frac{\partial \hat{m}_1(2)}{\partial fc_2} \right)^2 + \dots + \left(\frac{\partial \hat{m}_{ne}(1)}{\partial fc_2} \right)^2 + \left(\frac{\partial \hat{m}_{ne}(2)}{\partial fc_2} \right)^2 \\ \left(\frac{\partial \hat{m}_1(1)}{\partial cc_1} \right)^2 + \left(\frac{\partial \hat{m}_1(2)}{\partial cc_1} \right)^2 + \dots + \left(\frac{\partial \hat{m}_{ne}(1)}{\partial cc_1} \right)^2 + \left(\frac{\partial \hat{m}_{ne}(2)}{\partial cc_1} \right)^2 \\ \left(\frac{\partial \hat{m}_1(1)}{\partial cc_2} \right)^2 + \left(\frac{\partial \hat{m}_1(2)}{\partial cc_2} \right)^2 + \dots + \left(\frac{\partial \hat{m}_{ne}(1)}{\partial cc_2} \right)^2 + \left(\frac{\partial \hat{m}_{ne}(2)}{\partial cc_2} \right)^2 \end{pmatrix}$$

Se puede observar que cada fila está formada por las segundas derivadas de las coordenadas de las esquinas calculadas con respecto a fc_1 , fc_2 , cc_1 y cc_2 respectivamente. Calculando cada fila por separado y asociándolo a su respectivo parámetro intrínseco, se obtiene:

²En este desarrollo se está considerando a la matriz de pesos W como una matriz identidad, de esta forma estamos considerando un error en la detección de las esquinas caracterizado por una $\sigma = 1$ píxel. Para generalizarlo a cualquier σ bastaría con multiplicar todos los elementos de la matriz hessiana por el inverso de dicho valor.

$$\begin{aligned}
 fc_1 &\rightarrow \sum_{i=1}^{ne} \left(\frac{Xc + \hat{M}_i(1)C(\alpha)C(\beta) - \hat{M}_i(2)C(\beta)S(\alpha)}{Zc + \hat{M}_i(1)(S(\alpha)S(\theta) - C(\alpha)S(\beta)C(\theta)) + \hat{M}_i(2)(C(\alpha)S(\theta) + S(\alpha)S(\beta)C(\theta))} \right)^2 \\
 fc_2 &\rightarrow \sum_{i=1}^{ne} \left(\frac{Yc + \hat{M}_i(1)(S(\alpha)C(\theta) + C(\alpha)S(\beta)S(\theta)) + \hat{M}_i(2)(C(\alpha)C(\theta) - S(\alpha)S(\beta)S(\theta))}{Zc + \hat{M}_i(1)(S(\alpha)S(\theta) - C(\alpha)S(\beta)C(\theta)) + \hat{M}_i(2)(C(\alpha)S(\theta) + S(\alpha)S(\beta)C(\theta))} \right)^2 \\
 cc_1 &\rightarrow \sum_{i=1}^{ne} 1 = ne \quad cc_2 \rightarrow \sum_{i=1}^{ne} 1 = ne
 \end{aligned}$$

Obsérvese que en los términos asociados a fc_1 y fc_2 , el parámetro que define el alejamiento Zc aparece únicamente en el denominador. Además, Zc domina al resto de términos al aumentar, al ser éstos últimos proporcionales a productos de senos y cosenos.

Por lo tanto podemos concluir que conforme el alejamiento en Zc relativo a la cámara aumenta, la información añadida por las coordenadas de las esquinas disminuye. Esto se puede relacionar con la cuantificación de la incertidumbre asociada al cálculo de dichos parámetros, ya que tal y como se explicó en la sección 3.3.1, la inversa de la matriz hessiana representa la matriz de covarianza de la estimación de los parámetros.

Es posible dar una intuición a este comportamiento: Si el patrón se aleja indefinidamente de la cámara, todas las coordenadas de las esquinas tenderán a aparecer en un único punto en la imagen. La información que se podría extraer del patrón en este caso sería nula.

Tras estas observaciones extraídas mediante simulación y verificadas de manera teórica, se concluyó que para mejorar la precisión de la calibración había que situar lo más cerca posible el patrón de la cámara, para así reducir al máximo el valor de Z_c . Para demostrar esto de la manera más formal y correcta posible, se debería calcular analíticamente la inversa de la matriz hessiana, pero debido al coste computacional y al tiempo que requeriría, no se planteó dentro de los objetivos de este TFG.

4.5. Sensibilidad con la orientación

Tras haber extraído la conclusión anterior, se creó otro conjunto de aproximadamente 4500 patrones de calibración con el Algoritmo 1 incluyendo la condición de

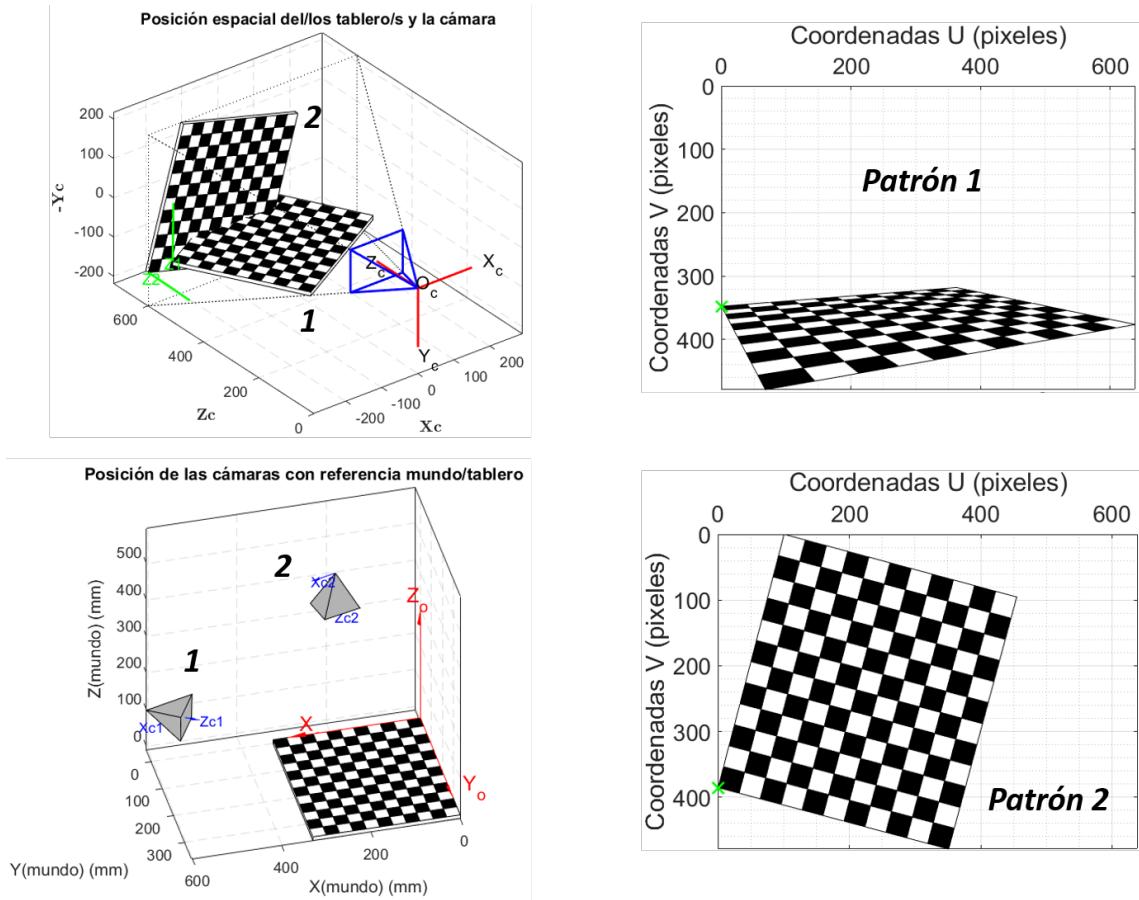


Figura 4.14: Ejemplo de 2 patrones situados lo más cerca posible de la cámara.

que todos ellos debían estar lo más cerca posible de la cámara. En la Figura 4.14 se muestra un ejemplo de esto.

Para analizar la influencia de la orientación del patrón, se decidió estudiar cómo influía el ángulo que formaban en 3D la normal del patrón de calibración y el eje óptico de la cámara.

Debido a las condiciones de visibilidad impuestas, este ángulo podía situarse únicamente en un rango $\in (90^\circ, 180^\circ)$; aunque, para una mayor facilidad de entendimiento el ángulo calculado fue el existente entre el eje óptico y la normal del plano en sentido contrario; de esta forma el nuevo rango se sitúa entre los $(0^\circ, 90^\circ)$. De aquí en adelante nos referiremos a dicho ángulo como θ .

Un total de 1000 calibraciones de 15 imágenes cada una fueron llevadas a cabo (aunque también se realizaron calibraciones cuando el número de imágenes era ≥ 2 , de esta forma el total llevado a cabo es de 1000×14). A partir de ellas, se observó que podía existir una relación entre θ y la precisión de la calibración debido a 2

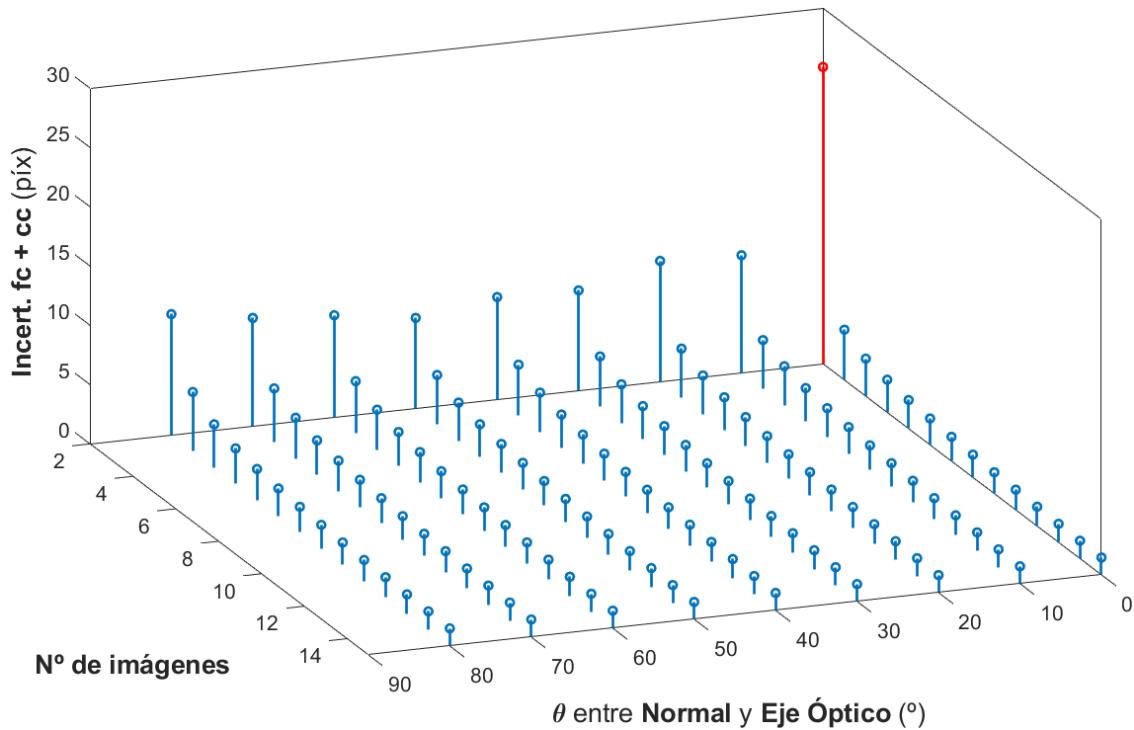


Figura 4.15: Valores promedio de la suma de la incertidumbre en la estimación de fc y cc. Cada barra representa el valor promedio de esta incertidumbre de todas aquellas calibraciones en las que en el número de imágenes indicado se ha utilizado al menos un patrón con un valor de θ igual o superior en 10° al asociado.

observaciones:

- Peores resultados cuando se calibraba con 2 imágenes si había presencia de patrones de calibración con un $\theta \leq 10^\circ$. Esto se puede observar en la Figura 4.15.
- Peor precisión si se supera un promedio de $\theta = 70^\circ$ conforme aumenta el número de imágenes, tal y como se muestra en la Figura 4.16.

Tras estas observaciones se analizó el problema desde la perspectiva de encontrar el valor del ángulo θ para el que se obtenían menores errores. Dicho de otra forma, se decidió analizar la precisión de la calibración fijando un θ y distribuyendo el patrón de calibración por todo el plano imagen gracias a un giro realizado en torno al eje óptico (así θ no varía).

Este método como veremos más adelante, obtuvo unos resultados positivos. Una ventaja es que se evitan las rotaciones degeneradas introducidas en la sección 3.2.2

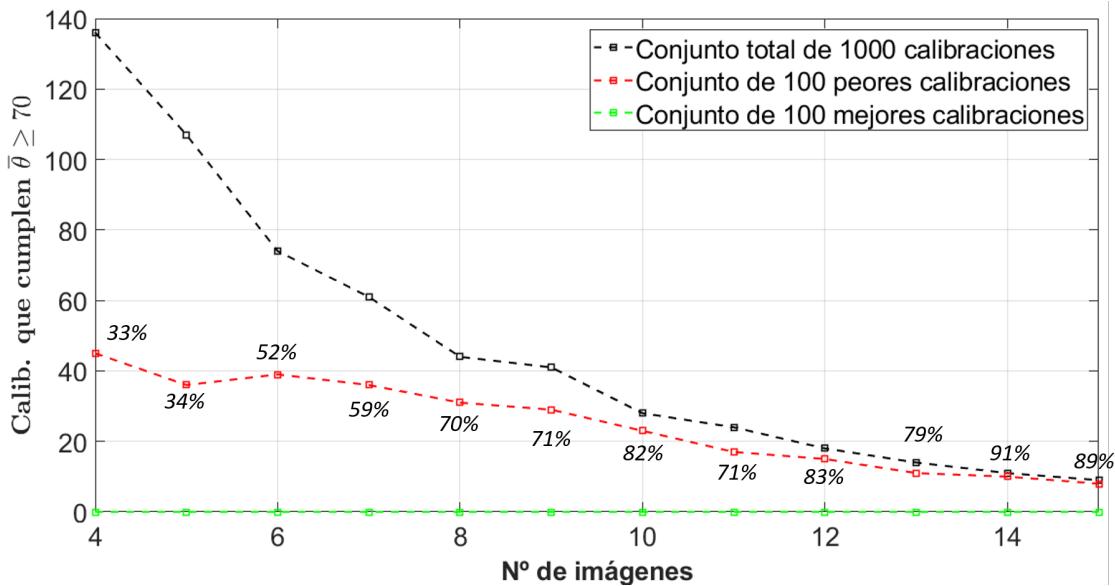


Figura 4.16: Análisis sobre las 100 peores y mejores calibraciones (en cuanto a incertidumbre) que muestra cuántas de ellas y en qué % con respecto al total (línea negra) cumplen que $\bar{\theta} \geq 70^\circ$.

debido a que en este caso la rotación (R) del patrón con respecto a la cámara viene dada por:

$$R_o = Rot_x(180^\circ)Rot_z(-90^\circ) \quad \rightarrow \quad R = Rot_z(\alpha)R_oRot_x(\theta)$$

Donde R_o es una rotación que sirve para situar el patrón de calibración frente a la cámara y $Rot_z(\alpha)$ es la rotación realizada sobre el eje óptico que permite distribuir el patrón por la imagen. De esta forma si $\theta \neq 0$, hay una diferencia de $Rot_x(\theta)$ entre 2 rotaciones consecutivas y nunca una Rot_z , evitando así las rotaciones degeneradas.

Para obtener el valor óptimo de θ , se consideró a aquel que minimizase la suma de las incertidumbres asociadas a los parámetros fc y cc . Se excluyó de esta suma a la incertidumbre de los coeficientes kc al venir dados en unas unidades distintas a los píxeles; sin embargo, esta consideración no se pensó como negativa ya que al minimizar la incertidumbre del resto de parámetros, se contribuye a una mejora global de la calibración.

Analizar el ángulo θ óptimo para un único modelo de cámara tal y como hemos hecho hasta ahora sería un análisis incompleto. Por ello ejecutamos el método del Algoritmo 2 para evaluar aproximadamente 200 modelos diferentes de cámara

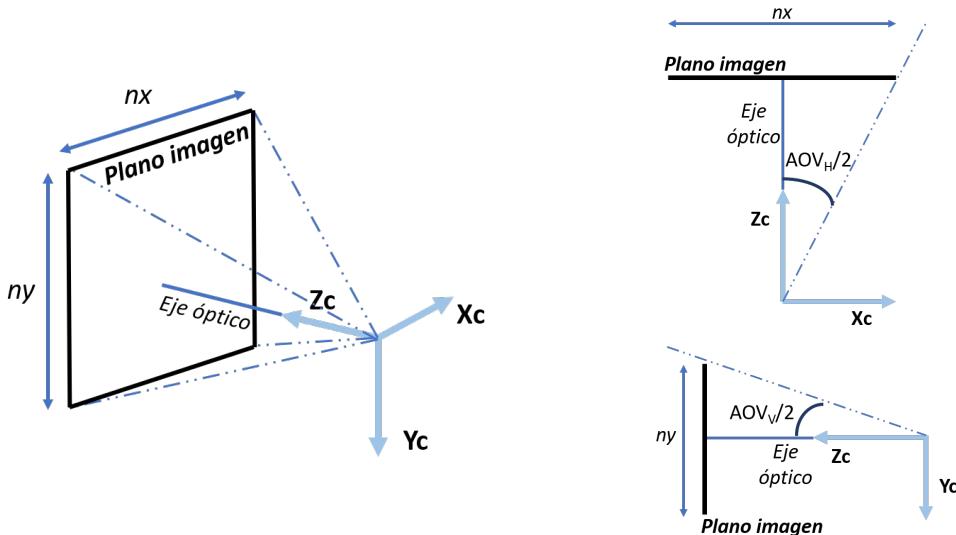


Figura 4.17: Definición gráfica de las 2 componentes del ángulo de visión.

variando el *ángulo de visión* (AOV) horizontal y vertical:

$$AOV_H = 2 \arctan \left(\frac{0,5nx}{fc_1} \right) \quad AOV_V = 2 \arctan \left(\frac{0,5ny}{fc_2} \right)$$

Donde recordemos que nx y ny son la resolución en píxeles horizontal y vertical. Estas dos componentes representan el ángulo medido desde el eje óptico que la cámara es capaz de capturar. Gráficamente se puede ver en la Figura 4.17.

Tal y como se puede ver en el algoritmo, únicamente se varía la distancia focal para cambiar el valor del ángulo de visión. Otro análisis muy relevante es la comparación con otros modelos de cámara en los que también haya variación de la resolución.

El ruido en la detección de las esquinas se consideró igual para todos los valores de θ y de un valor de 0,1 píxeles. Esta consideración será mejorada tal y como se verá en el capítulo correspondiente a la implementación con imágenes reales.

Los resultados a este análisis se muestran en la Figura 4.18a, a partir de la cual se puede extraer una relación aproximadamente cuadrática entre el ángulo de visión y el ángulo θ que proporcionaba una menor incertidumbre y que viene dada por:

$$\theta_{optimal} = 0,00923 AOV^2 - 1,09 AOV + 79 \quad (4.1)$$

Tal y como se muestra en la figura, esta relación en realidad está asociada al ángulo de visión horizontal; sin embargo, se adoptó también para la componente vertical al

Algoritmo 2: Análisis para encontrar el ángulo θ óptimo

```

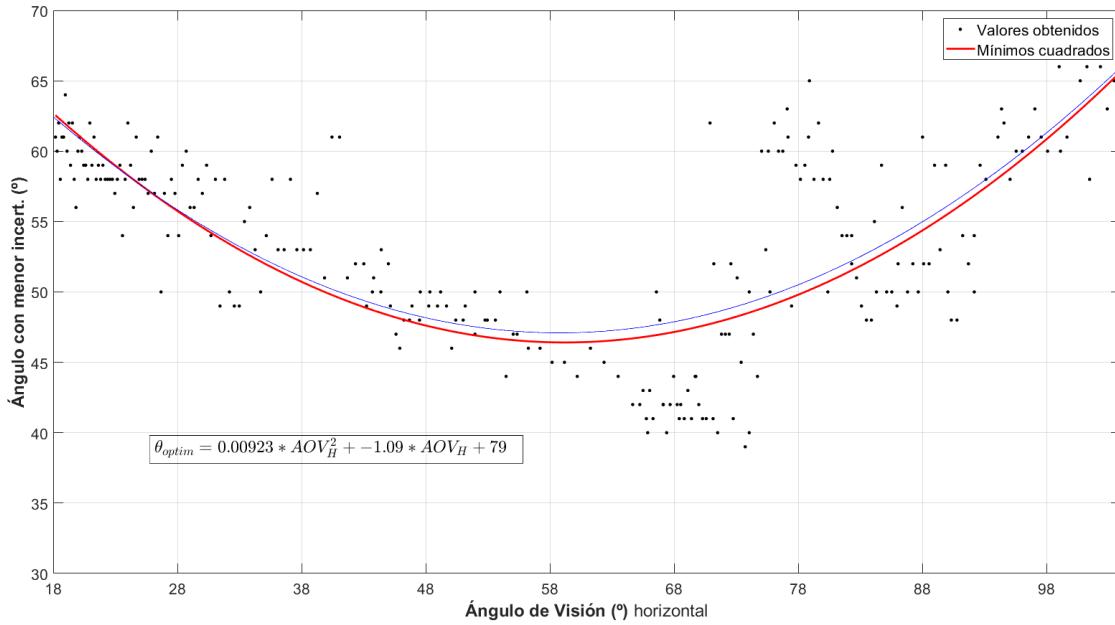
1:  $[AOV_o, AOV_f] = [18^\circ, 102^\circ]$  {Ángulo de visión inicial y final}
2:  $[nx, ny] = [640, 480]$  {Resolución horizontal y vertical}
3:  $cc = [0, 5nx, 0, 5ny]$  {Consideramos el punto principal en el centro de la
   imagen}
4:  $patron_{geom}$  {Geometría del patrón de calibración}
5:  $\sigma_{deteccion}$  {Desviación típica del error en la detección de las esquinas}
6: for  $cam = 1$  to 200 do
7:    $AOV_H = \frac{AOV_f - AOV_o}{200} cam + AOV_o$ 
8:    $[fc_1, fc_2] = f(AOV_H, nx)$  {Aplicamos la ec. del AOV considerando
    $fc_1 = fc_2\}$ 
9:   for  $\theta = 1^\circ$  to  $90^\circ$  con paso =  $1^\circ$  do
10:    for  $\alpha = 0^\circ$  to  $359^\circ$  con paso =  $24^\circ$  do
11:       $R_{patron} = Rot_z(\alpha)R_oRot_x(\theta)$ 
12:       $T_{patron} = Crear\_traslacion(fc, cc, nx, ny, patron_{geom}, R_{patron})$ 
13:       $esquinas = Crear\_esquinas(cam, R_{patron}, T_{patron}, \theta, \sigma_{deteccion})$ 
14:    end for
15:     $cali = Realizar\_calibración(cam, \theta, R_{patron}, T_{patron}, esquinas)$ 
16:     $all\_cal = Guardar\_calibración(all\_cal, cali)$ 
17:  end for
18:   $cal\_min\_incert = Selecc\_θ\_con\_mín\_incert(all\_cal)$ 
19: end for

```

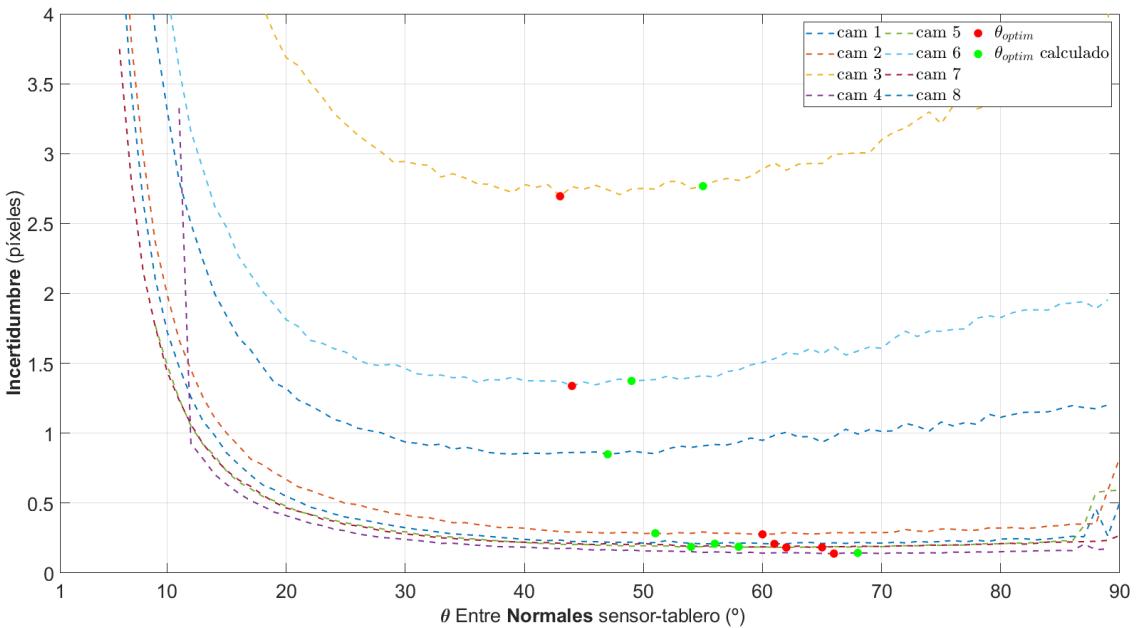
ser similares y al haber barrido un mayor rango de la componente horizontal y por lo tanto ser más preciso.

Para probar si esta relación se podía considerar como válida se evaluaron diferentes modelos de cámara, a los que se les sometió un análisis similar al presentado en el algoritmo 2. De esta forma se pudo comparar el θ óptimo en cuanto a incertidumbre en los parámetros fc y cc y el calculado por la expresión 4.1. Los modelos de cámara evaluados se recogen en la tabla 4.3 y los resultados de esto se muestran en la Figura 4.18b.

Algo a destacar de la Figura 4.18b es el hecho de que los valores de θ calculados mediante la expresión 4.1 (puntos verdes) difieren ligeramente de los valores óptimos



(a) Valor de θ con menor incertidumbre asociada a cc y fc en función del ángulo de visión. La curva azul es el ajuste cuadrático teniendo en cuenta todos los puntos y la roja es el resultado de excluir aquellos puntos que se desvían más de una desviación típica del ajuste azul. La expresión 4.1 se corresponde con la curva roja.



(b) Incertidumbre en función del valor de θ con el que se lleva a cabo la calibración para las cámaras de la tabla 4.3. El valor de θ asociado a un menor valor de incertidumbre aparece marcado en rojo, mientras que el θ calculado mediante la expresión 4.1 aparece marcado en verde.

Figura 4.18

Parámetros		1)	2)	3)	4)	5)	6)	7)	8)	Unidades
<i>Resolución</i>	Horiz. (nx)	1600	2100	1400	2000	800	1000	1400	2500	Píxeles
	Vert. (ny)	1200	1900	1300	1600	700	1000	1400	2000	
<i>Distancia focal</i>	fc1	700	1200	2500	660	400	1200	660	2000	Píxeles
	fc2	700	1200	2500	660	400	1200	660	2000	
<i>Punto principal</i>	cc1 (u_o)	799,5	1049,5	699,5	999,5	399,5	499,5	699,5	1249,5	Píxeles
	cc2 (v_o)	599,5	949,5	649,5	799,5	349,5	499,5	699,5	999,5	
<i>Coef. de distorsión</i>	kc1	-0,26	-0,35	-0,1	-0,3	-0,1	-0,3	-0,26	-0,2	mm^{-2}
	kc2	0,13	0,2	0	0,1	0,15	0,13	0,13	0,1	
<i>Áng. de visión</i>	<i>AOV_H</i>	97	82	31	113	90	45	93	64	(°)

Tabla 4.3: Modelos de cámara usados para validar la expresión 4.1.

reales (puntos rojos), y además esta diferencia se puede explicar por la existencia de una zona de estabilidad de la incertidumbre que en general se mantiene entre los 40° y 70°.

Como conclusión ante estas observaciones, se pensó en establecer un método que consistiese en situar el patrón lo más cerca posible con las orientaciones determinadas por el θ óptimo calculado. Para probar su validez, se comparó con el estado del arte así como con una colocación aleatoria de los patrones de calibración.

4.6. Comparación con el estado del arte

El método desarrollado se comparó con el trabajo llevado a cabo por A. Richardson [18], conocido como *AprilCal*, con el trabajo de S. Peng y P. Sturm [15], conocido como *Calibration Wizard* y con el método implementado por R. Boada en su Trabajo Fin de Máster [3].

Todos estos métodos tienen en común que son incrementales; es decir, se parte de un pequeño número de imágenes (generalmente 2 ó 3) con las que se realiza una calibración inicial para tener una primera estimación de los parámetros.

A partir de esta estimación inicial, se calcula qué ubicación del patrón es la mejor en base a distintos criterios y una vez determinada, se realiza otra foto al patrón y se calibra de nuevo la cámara hasta alcanzar un cierto grado de precisión. Los fundamentos del criterio que sigue cada método se explican a continuación:

1. AprilCal: en él, se selecciona qué ubicación es la mejor de en torno a 60 ubicaciones posibles del patrón, las cuales están distribuidas por todo el plano imagen. La selección de la ubicación óptima es aquella que presenta una menor incertidumbre en los parámetros intrínsecos.
2. Calibration Wizard: la metodología es similar al anterior; es decir, se selecciona la ubicación del patrón que minimiza la incertidumbre de los parámetros intrínsecos. Sin embargo, en vez de considerar un conjunto de 60 posibles ubicaciones, se realiza una optimización global que tiene en cuenta todo el espacio de trabajo 3D.
3. TFM de R. Boada: este método presenta una variación al de AprilCal. La ubicación seleccionada es la que maximiza el parámetro denominado como *MaxERE*; es decir, aquella con la que se maximiza el error entre las coordenadas de las esquinas detectadas en la imagen y las calculadas con la estimación actual de los parámetros intrínsecos.

Como se puede apreciar, estos métodos presentan un coste computacional mayor que el método que estamos proponiendo, ya que todos ellos tienen que realizar sucesivas calibraciones por cada imagen a la vez que los cálculos asociados al cómputo de la siguiente mejor ubicación del patrón de calibración.

Con el método que se está proponiendo, únicamente se necesitaría una calibración inicial con 1 ó 2 imágenes para tener una estimación de la distancia focal (f_{ci}) y poder estimar así el ángulo de visión de la cámara. De esta forma se evitaría la necesidad de realizar una calibración por cada imagen y además los cálculos asociados a determinar la mejor ubicación del patrón se realizarían nada más conocer el ángulo de visión.

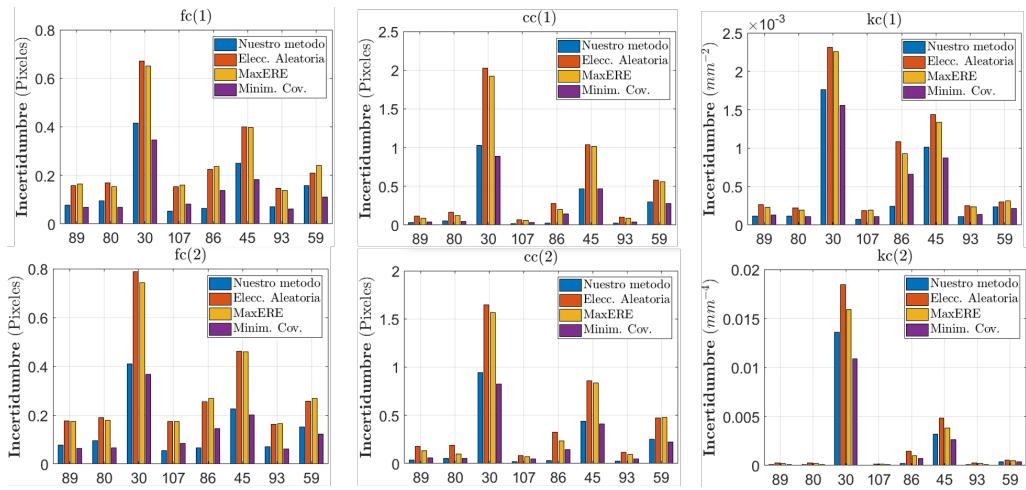
Para comparar con todos los métodos, en vez de considerar 60 patrones únicamente como en 1) y 3), se consideraron en torno a 1.400 patrones creados mediante el algoritmo 1 con un $\Delta\text{ang} = 40^\circ$, siendo así más exhaustivos en la comparación con estos dos métodos. Sin embargo, en cuanto al método 2) esto supone una aproximación, ya que no se están utilizando todas las posibles ubicaciones 3D.

De esta forma, al utilizar un mismo número de patrones de calibración en la evaluación de los 3 métodos, se juntó a los métodos 1) y 2) en uno sólo al ser esa la

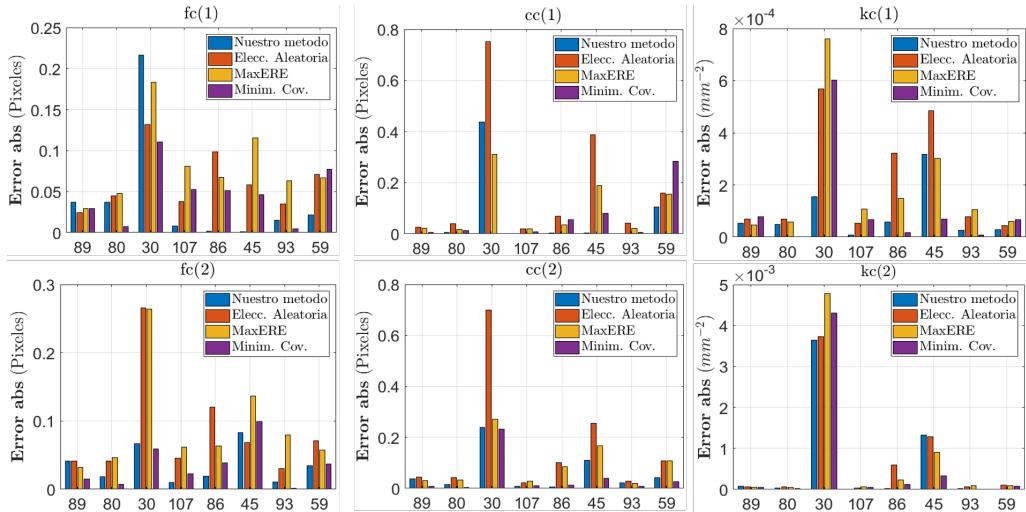
principal diferencia entre ellos.

Una última consideración realizada, fue el hecho de considerar que en la detección de las esquinas todos los métodos con sus respectivas colocaciones de patrones de calibración elegidas presentaban una misma desviación típica de $\sigma = 0,1$ píxeles.

Los modelos de cámara evaluados fueron los de la tabla 4.3, la cual se consideró como una buena muestra debido a la gran variación de los parámetros. Los resultados de la comparación en cuanto a error e incertidumbre en la estimación de los parámetros se muestran en la Figura 4.19 y de manera esquemática en la tabla 4.4.



(a) Incertidumbre en la estimación.



(b) Error absoluto en la estimación.

Figura 4.19: Comparación. De izquierda a derecha los parámetros son: fc_i , cc_i y kc_i . El eje x representa el $A\bar{O}V$ de cada cámara.

Mejoras en los 8 modelos de cámara evaluados						
Parámetros	Con respecto a los métodos:					
	Minim. Covarianza (AprilCal y Calib. Wizard)		MaxEre (TFM)		Elección aleatoria	
	Incertidumbre	Error	Incertidumbre	Error	Incertidumbre	Error
fc_1	2/8	4/8	8/8	6/8	8/8	6/8
fc_2	2/8	4/8	8/8	7/8	8/8	7/8
cc_1	6/8	7/8	8/8	7/8	8/8	8/8
cc_2	5/8	2/8	8/8	6/8	8/8	8/8
kc_1	4/8	4/8	8/8	6/8	8/8	8/8
kc_2	4/8	4/8	8/8	6/8	8/8	6/8
TOTAL	23/48	25/48	48/48	38/48	48/48	43/48

Tabla 4.4: Comparación con el estado del arte en la que se muestra la cantidad de mejoras en la estimación con respecto al total de cámaras (8) y en la última fila con respecto al total de evaluaciones (48).

A partir de estos resultados podemos observar que el método que se está proponiendo produce unos resultados similares a los métodos 1) y 2) denominados como “Minim. Cov.” tanto en la gráfica 4.19 como en la tabla 4.4 ya que en torno al 50% se consiguen ofrecer unos resultados mejores.

Por otro lado, en cuanto al método 3) y a la elección aleatoria de la ubicación de los patrones de calibración, denominados como “MaxERE” y “Elecc. Aleatoria”, el método que se está proponiendo consigue ofrecer unos resultados mejores el 100% de las veces en cuanto a incertidumbre y el 79% y 89% de las veces en cuanto a error absoluto respectivamente.

Teniendo en cuenta el mayor coste computacional asociado a los métodos 1) y 2), se puede concluir que el método presentado tiene ventajas sobre el estado del arte. Por ello, se decidió desarrollar en *MATLAB* una herramienta que permitiese validar con imágenes reales estas conclusiones y que se presenta en el siguiente capítulo.

Capítulo 5

IMPLEMENTACIÓN EN IMÁGENES REALES

Para llevar a la práctica el método propuesto, y que consiste en establecer el patrón de calibración lo más cerca posible de la cámara a la vez que situarlo con el ángulo θ calculado a partir del ángulo de visión de la cámara, se desarrolló en *MATLAB* una herramienta que permite mostrar en la grabación en vivo la proyección del patrón de calibración con la posición adecuada.

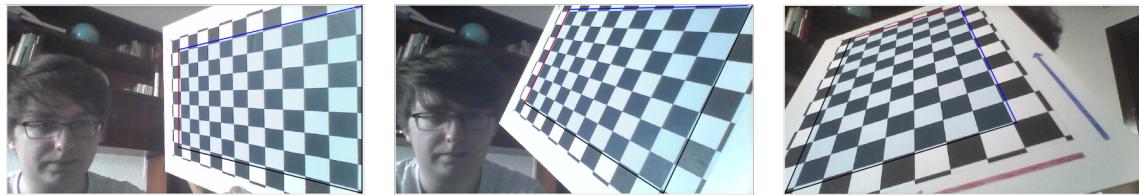
En la Figura 5.1 se muestran varios ejemplos de esto. Nótese el hecho de que el patrón de calibración real utilizado es de 9×13 cuadrados; sin embargo, por costumbre, las calibraciones con ese patrón se realizaron como si fuese uno de 7×11 ; es decir, sin considerar a los cuadrados de los bordes más extremos.

El modelo de cámara que se evalúo en este parte fue una *webcam logitech C920*, a la que se le sometió una calibración con el método propuesto y una calibración con imágenes “estándar”; es decir, intentando distribuir el patrón por todo el plano imagen para tener un resultados honestos con los que comparar.

Como última consideración, la métrica que se utilizó en esta parte para comparar los métodos fue la de la incertidumbre, dado que no se es conocedor de los verdaderos valores de los parámetros de la cámara que calibrámos. De esta forma, métricas como las del error relativo o absoluto, que pudimos llevar a cabo en la simulación, no es posible utilizarlas en esta fase.



(a) Rectángulos proyectados sobre la grabación de la cámara en con el θ óptimo calculado.



(b) Muestra del patrón real colocado sobre la proyección de los rectángulos.



(c) Imágenes en blanco y negro con las esquinas del patrón de calibración detectadas.

Figura 5.1

Tal y como se puede ver en la tabla 5.1, los resultados de esta comparación fueron negativos, al ofrecer nuestro método mayor incertidumbre en la estimación de los parámetros. Tras ello, realizamos un análisis para explorar las causas de esta aparente contradicción entre los resultados de simulación y las imágenes reales.

Gracias a un análisis exhaustivo, se observó que la causa que provocaba unos peores resultados era el hecho de que nuestro método presentaba una mayor desviación típica del error de reproyección ($\sigma_{\Delta m}$), tal y como se ve en la tabla 5.1.

$\sigma_{\Delta m}$ se puede asociar a la desviación típica en la detección de las esquinas [5] y es tal y como está implementado en [4]. De esta forma la expresión asociada al cálculo de la matriz de covarianza, tratada en la sección 3.3.1, se puede expresar de la siguiente forma:

$$Cov = H^{-1} \approx (J^T W J)^{-1} = (W J^T J)^{-1} = W^{-1} (J^T J)^{-1}$$

Donde recordemos que W es la matriz de pesos semi-definida positiva y diagonal, donde los elementos de su traza se corresponden con el inverso de la varianza asociada a la detección de las esquinas y que se considera igual para todas. Es decir, los

Parámetros	Nuestro método (15 imágenes)		Calibración “estándar” (15 imágenes)	
	$Incertidumbre_i$	$3\sigma_i$	$Incertidumbre_i$	$3\sigma_i$
fc_1	0,982 píx.	2,47	0,645 píx.	4,38
fc_2	1,01 píx.	2,54	0,564 píx.	3,84
cc_1	2,06 píx.	5,18	1,15 píx.	7,82
cc_2	1,99 píx.	5,00	1,11 píx.	7,55
kc_1	$0,00970 \text{ mm}^{-2}$	$0,0244 \frac{\text{mm}^{-2}}{\text{pix}}$	$0,00740 \text{ mm}^{-2}$	$0,0503 \frac{\text{mm}^{-2}}{\text{pix}}$
kc_2	$0,0279 \text{ mm}^{-4}$	$0,0701 \frac{\text{mm}^{-4}}{\text{pix}}$	$0,0407 \text{ mm}^{-4}$	$0,277 \frac{\text{mm}^{-4}}{\text{pix}}$
$\sigma_{\Delta m}$	0,398 píx.		0,147 píx.	

Tabla 5.1: Primera comparación con imágenes reales. Modelo: *webcam logitech C920*.

elementos de su traza vienen dados por $diag(W) = \left(\frac{1}{\sigma_{\Delta m}}\right)^2$.

Si consideramos el parámetro con el elemento asociado $(J^T J(i, i))^{-1/2}$ y lo denominamos como σ_i , la incertidumbre asociada al parámetro es posible calcularla como $\pm 3\sigma_i\sigma_{\Delta m}$. En la tabla 5.1 se observa como nuestro método en todos los casos ofrece un menor valor de $3\sigma_i$, lo que resulta algo positivo pero que se ve perjudicado por la desviación en la detección en las esquinas $\sigma_{\Delta m}$.

Por ello se pensó en analizar si existía una influencia entre el ángulo θ calculado para nuestro método, con la precisión o desviación típica existente en la detección de las esquinas. Trabajos como [21] han mostrado que la precisión en la detección de esquinas de Harris [9], método que está implementado en [4], empeora cuando aumenta el ángulo entre el eje óptico y la normal de la superficie en la que está la esquina.

Para este análisis, se realizaron un total de 10 calibraciones de 15 imágenes reales cada una, con distintos valores de θ y utilizando la misma webcam. Esto fue posible mediante la herramienta visual desarrollada en el TFG (Figura 5.1¹).

En cada calibración se recogió el valor de $\sigma_{\Delta m}$ observándose la tendencia mostrada en la gráfica 5.2. Dicha tendencia se decidió aproximar por un valor constante de $\sigma_{\Delta m} \approx 0,28$ píxeles entre los valores de $\theta \in [5^\circ, 40^\circ]$ y por una tendencia lineal

¹Los valores de $\theta < 5^\circ$ y $\theta > 65^\circ$ no fueron tenidos en cuenta. En el primer caso la calibración no convergía por aproximarse a rotaciones degeneradas (ocurren cuando $\theta = 0^\circ$ como ya se ha explicado). En el segundo caso el elevado escorzo con el que el patrón es observado dificultaba considerablemente la detección de las esquinas.

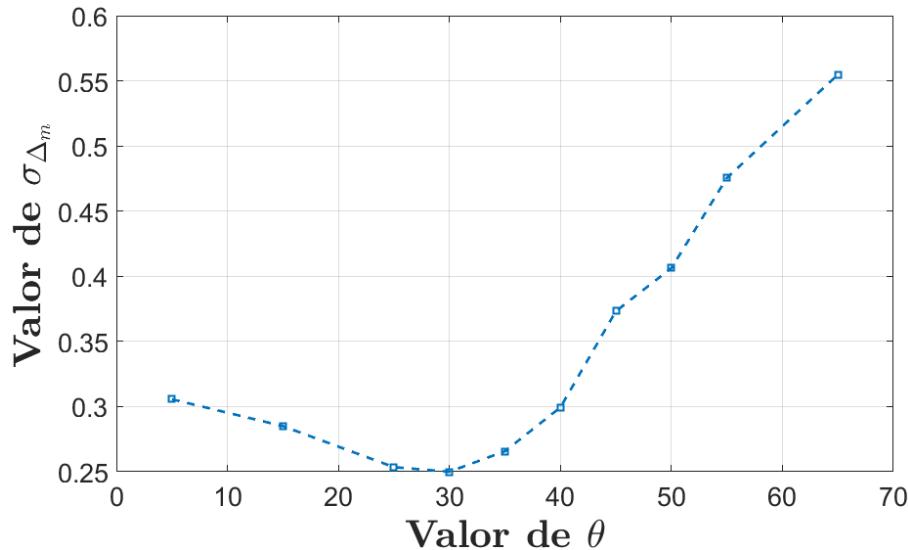


Figura 5.2: Desviación típica en la detección de las esquinas (σ_{Δ_m}) en función de θ .

con $\theta > 40^\circ$. Lo que nos sirvió para confirmar que en efecto, el ruido en la detección de las esquinas variaba con θ .

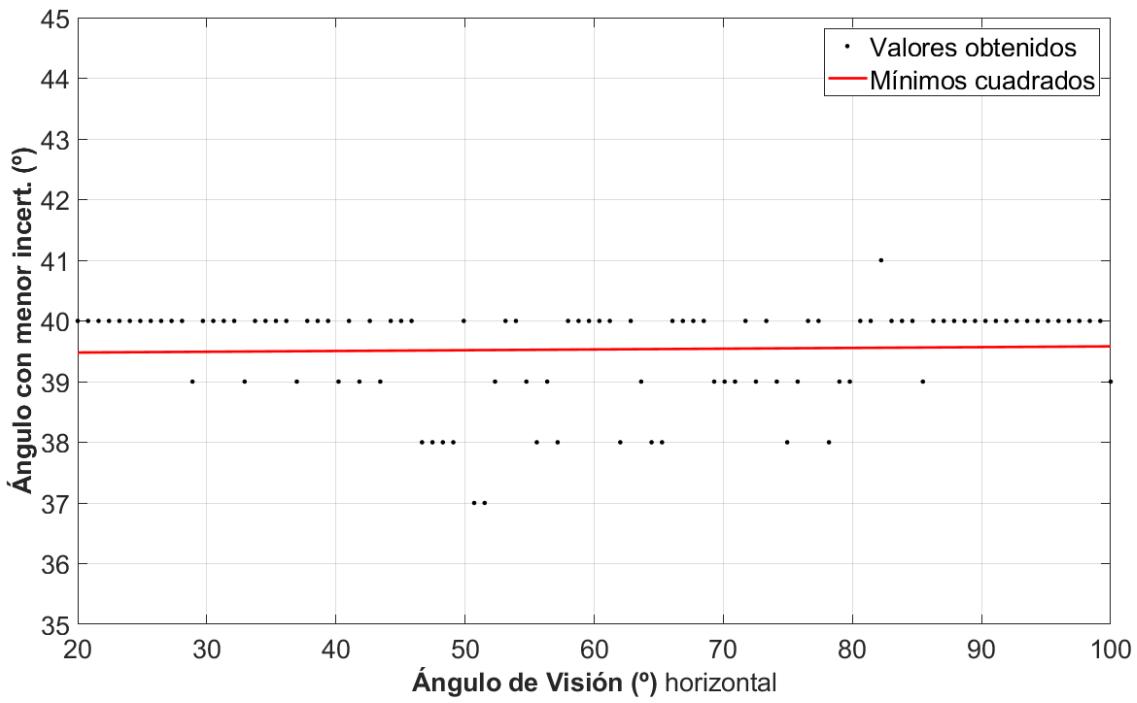
Sabiendo esto, lo siguiente que se hizo fue llevar a cabo de nuevo el algoritmo 2 incluyendo la relación que se acaba de mostrar. Es decir, se calculó de nuevo cual era el valor de θ óptimo en función del ángulo de visión. Los resultados se pueden ver en la Figura 5.3a y tal como se observa el valor de θ óptimo pasa a ser uno prácticamente constante e igual a 40° .

De nuevo, para poder confirmar si esta estimación es válida de acuerdo a las suposiciones realizadas, se llevó a cabo una simulación con los modelos de cámara de la tabla 4.3 para verificar así si la estimación de 40° se aproxima al valor de θ que conlleva una menor incertidumbre. Los resultados aparecen en la Figura 5.3b.

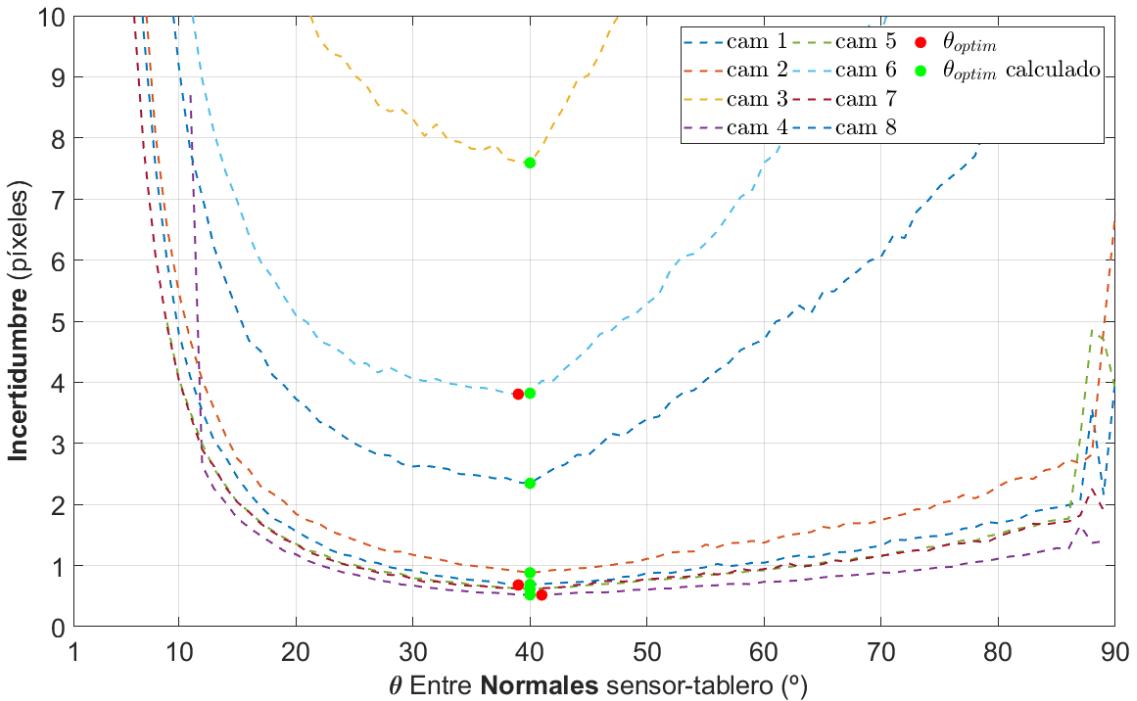
De esta forma se ve que efectivamente la estimación se cumple aún variando, no solo la distancia focal, sino todos los demás parámetros ya que como mucho el error en la estimación del valor óptimo es de 1° .

Con los resultados de este análisis, se realizaron de nuevo los experimentos con imágenes reales. Esto solo tendría sentido si se utilizase una cámara distinta a la que se usó para extraer los datos de ruido en la detección de las esquinas, ya que sino se estaría predisponiendo que los resultados saliesen óptimos para esa cámara.

Es por esto por lo que se utilizó la otra cámara que se disponía con adaptabilidad



(a) Valor de θ con menor incertidumbre asociada a cc y fc en función del ángulo de visión.
Se observa que prácticamente es un valor constante.



(b) Incertidumbre en función del valor de θ con el que se lleva a cabo la calibración para las cámaras de la tabla 4.3. El valor de θ asociado a un menor valor de incertidumbre aparece marcado en rojo, mientras que el θ que ha sido considerado como óptimo, aparece marcado en verde. Se observa una correcta estimación.

Figura 5.3

Parámetros	Nuestro método		Con consideraciones muy similares		Con consideraciones similares		Sin consideraciones similares	
	$Incumbre_i$	$3\sigma_i$	$Incumbre_i$	$3\sigma_i$	$Incumbre_i$	$3\sigma_i$	$Incumbre_i$	$3\sigma_i$
fc_1	1,60 píx.	3,07	1,92 píx.	6,48	1,909 píx.	8,37	3,10 píx.	14,83
fc_2	1,82 píx.	3,49	1,81 píx.	6,10	2,03 píx.	8,90	2,90 píx.	13,88
cc_1	2,96 píx.	5,68	2,90 píx.	9,78	3,33 píx.	14,61	3,98 píx.	19,04
cc_2	2,51 píx.	4,82	2,75 píx.	9,27	2,62 píx.	11,49	3,30 píx.	15,79
kc_1	$0,00550 \text{ mm}^{-2}$	$0,0106 \frac{\text{mm}^{-2}}{\text{pix}}$	$0,00870 \text{ mm}^{-2}$	$0,0294 \frac{\text{mm}^{-2}}{\text{pix}}$	$0,00580 \text{ mm}^{-2}$	$0,0254 \frac{\text{mm}^{-2}}{\text{pix}}$	$0,0066 \text{ mm}^{-2}$	$0,0316 \frac{\text{mm}^{-2}}{\text{pix}}$
kc_2	$0,00950 \text{ mm}^{-4}$	$0,0182 \frac{\text{mm}^{-4}}{\text{pix}}$	$0,0390 \text{ mm}^{-4}$	$0,132 \frac{\text{mm}^{-4}}{\text{pix}}$	$0,0192 \text{ mm}^{-4}$	$0,0842 \frac{\text{mm}^{-4}}{\text{pix}}$	$0,0135 \text{ mm}^{-4}$	$0,0646 \frac{\text{mm}^{-4}}{\text{pix}}$
$\sigma_{\Delta m}$	0,521 píx.		0,296 píx.		0,228 píx.		0,209 píx.	

Tabla 5.2: Comparación con imágenes reales incluyendo la tendencia estimada de ruido en la detección de esquinas. El modelo utilizado es la cámara integrada en el portátil *Toshiba Qosmio X870*. Nuestro método de izquierda a derecha, es comparado con una calibración en la que 1) se sitúan los patrones cerca de la cámara y rotando el patrón por todo el plano imagen, 2) Los patrones se alejan ligeramente y se sigue rotando por todo el plano imagen, 3) Los patrones se alejan más, el patrón se distribuye por todo el plano imagen, pero la rotación sobre su normal es menor. Todas las imágenes utilizadas se pueden ver en el anexo B.

a *MATLAB* y es el modelo integrado en el portátil de la marca *Toshiba* y modelo *Qosmio X870*. Con ella se obtuvieron los resultados que se ven en la tabla 5.2, donde se compara nuestro método con calibraciones llevadas a cabo con distintos grupos de imágenes que cumplen condiciones diferentes, para un mismo número de 15 imágenes.

En ella se observa que nuestro método ofrece una menor incertidumbre en todos los parámetros de todos los conjuntos de imágenes, exceptuando a los parámetros cc_1 y fc_2 del conjunto de imágenes más similar (situado en la tabla 5.2 entre las columnas 4-5) aunque esta diferencia es muy pequeña (0,06 y 0,01 píxeles respectivamente).

Además se observa que el término $3\sigma_i$ es inferior en nuestro método para todos los parámetros de todos los conjuntos de imágenes, lo que nos da entender que si el ruido en la detección de las esquinas fuese igual, nuestro método ofrecería la mitad de incertidumbre o incluso menos.

Como último análisis del TFG para evaluar la precisión ofrecida por nuestro método, se decidió recoger los valores de incertidumbre de las calibraciones de la tabla 5.2 tras cada vez que se añadía una imagen. Como punto de partida se consideraron 5 imágenes al ser una cantidad que permitía obtener resultados estables para todos los tipos de calibración de la tabla.

Los resultados se presentan en las Figuras 5.4 y 5.5. En la primera de ellas aparece el valor de incertidumbre teniendo en cuenta la diferencia existente de error en la detección de las esquinas ($\sigma_{\Delta m}$) y en la segunda, se muestra el valor de $3\sigma_i$ (recordemos que σ_i representa la raíz cuadrada de los elementos de la traza de $(J^T J)^{-1}$ y que cuanto más bajos sean, se contribuirá a una menor incertidumbre o desviación típica en la estimación).

En cuanto a la primera figura, las conclusiones son similares a las de la tabla 5.2; ya que tras un cierto número de imágenes, el método que se está proponiendo ofrece mejores resultados en los mismos parámetros. Igualmente es interesante comentar que, salvo en los parámetros de distorsión, los 3 métodos que son mejores en todas las imágenes son aquellos que tienen condiciones similares.

Por último en cuanto a la segunda figura, se observa que nuestro método ofrece un menor valor en todos los parámetros, algo que es positivo y muestra que si no fuese por el ruido existente en la detección de las esquinas, se ofrecerían unos mejores resultados en general. Aquí además se observa que en todos los parámetros, las 3 mejores calibraciones son aquellas que comparten condiciones similares al método propuesto.

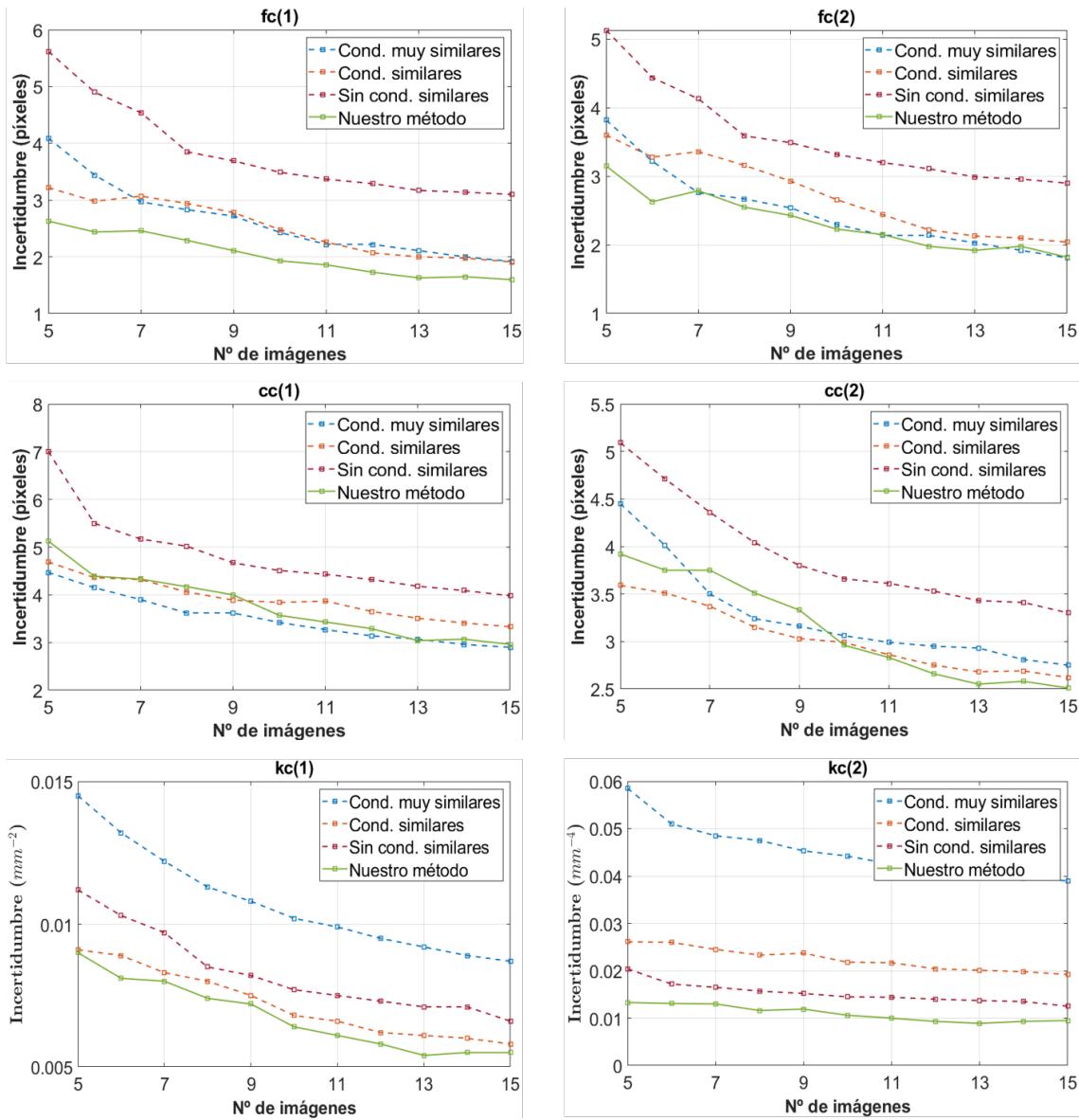


Figura 5.4: Comparación de la incertidumbre ofrecida por nuestro método con la ofrecida por calibraciones con consideraciones diferentes. En esta figura se incluye el hecho de que nuestro método tiene un mayor error en la detección de las esquinas.

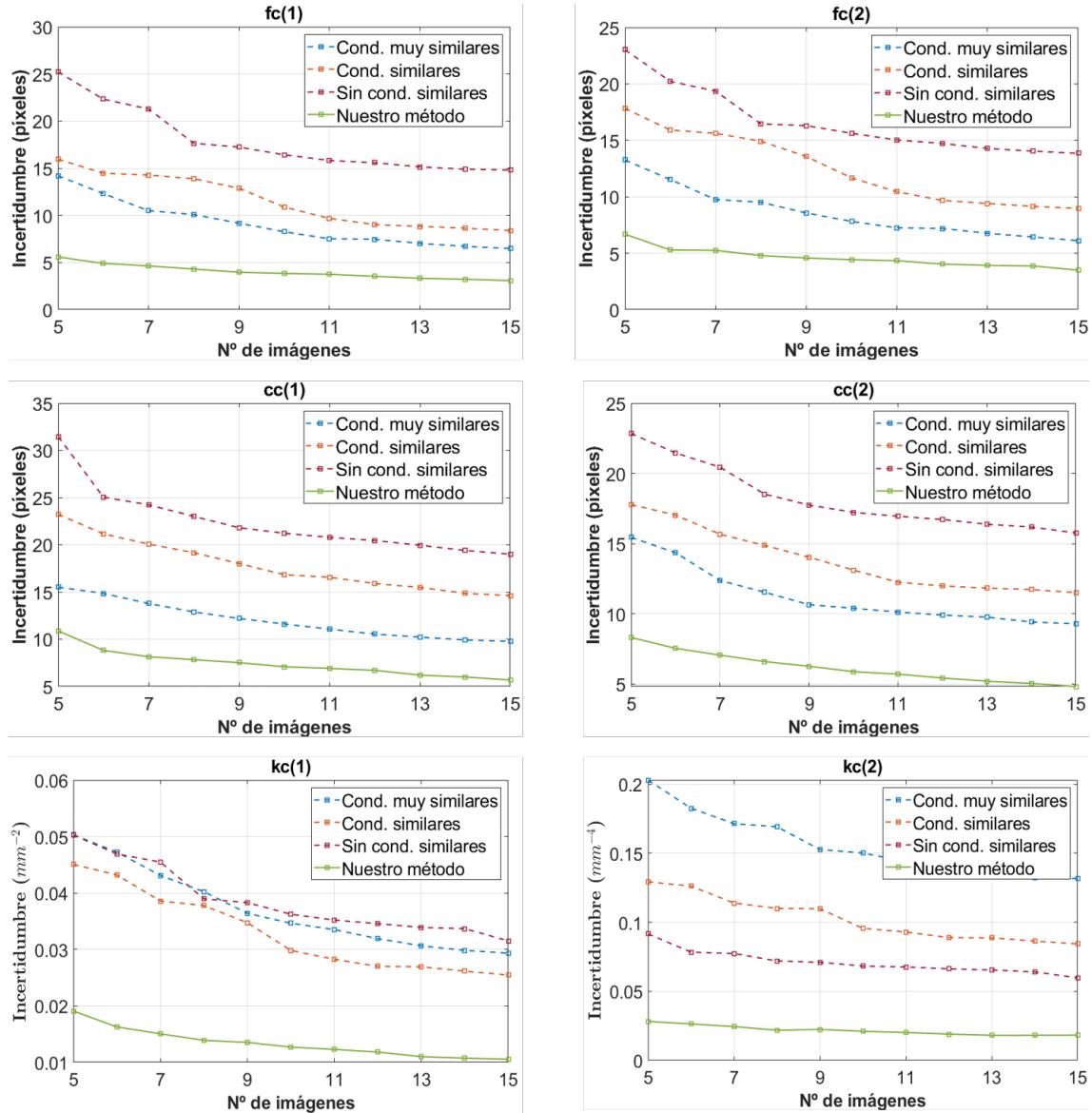


Figura 5.5: Comparación del valor $3\sigma_i$ (elementos de la traza de $(J^T J)^{-1/2}$), ofrecida por nuestro método con la ofrecida por calibraciones con consideraciones diferentes. Por tanto, aquí no se incluye el hecho de que nuestro método tiene un mayor error en la detección de las esquinas.

Capítulo 6

CONCLUSIONES

En este Trabajo de Fin de Grado, se ha planteado un método para la calibración de una cámara con bajo error. La problemática se ha abordado desde una perspectiva global, buscando las transformaciones (posición y orientación) de las cámaras que reducen el error de calibración. Este trabajo se diferencia así del estado del arte, que busca la siguiente mejor transformación a partir de una calibración dada. Como consecuencia de abordar el problema de manera general nuestro análisis concluye con las posiciones desde las cuales tomar las imágenes antes de la calibración y reduce el coste computacional de manera significativa.

Para ello se han analizado cómo influyen aspectos como la distancia entre el patrón de calibración y la cámara y su orientación con respecto a ésta. El análisis se ha realizado desde un punto de vista teórico y mediante simulación; determinándose finalmente la posibilidad de alcanzar una mejora en la precisión si se minimiza la distancia y se utiliza un cierto ángulo entre el patrón y el eje óptico de la cámara.

El método propuesto se ha evaluado en simulación con respecto al estado del arte, y se ha mostrado que ante unas mismas condiciones de ruido en la detección de las esquinas, es capaz de ofrecer resultados mejores o similares al estado del arte y con un coste computacional menor. Es decir, hemos encontrado configuraciones geométricas que conducen a un error de calibración bajo.

Se ha evaluado también el método propuesto con imágenes reales, lo que ha permitido observar un mayor error en la detección de las esquinas que en una calibración

realizada sin éste método. El error en la detección de las esquinas se propaga a los resultados de calibración, y ante errores variables el análisis anterior de las configuraciones geométricas no es totalmente válido.

Ante esto, se ha estudiado la relación entre el ruido y el ángulo entre el patrón y la cámara, lo que ha permitido mejorar los resultados e incluso ofrecer una estimación de la mayoría de los parámetros con una menor incertidumbre que en el resto de calibraciones analizadas, utilizando una cámara diferente a la utilizada para la estimación de dicha relación.

Igualmente se ha observado que el método seguía teniendo un mayor error en la detección de las esquinas, por lo que una posible línea futura a este trabajo sería probar distintos algoritmos de detección, o si fuese necesario intentar desarrollar uno propio. Además, tal y como se ha mostrado, nuestra propuesta no siempre ofrece una estimación con una menor incertidumbre. Por ello, una posible línea futura podría ser profundizar en el análisis para buscar posibles configuraciones con errores todavía más bajos. Podrían analizarse aspectos como, por ejemplo, conjuntos de imágenes tomados con distintos ángulos, o relajar la condición de que una de las esquinas del patrón esté muy cercana al borde de la imagen.

Bibliografía

- [1] Y. Abdel-Aziz, H. Karara, and M. Hauck. Direct linear transformation from comparator coordinates into object space coordinates in close-range photogrammetry. *Photogrammetric Engineering & Remote Sensing*, 81(2):103–107, 2015.
- [2] S. S. Beauchemin and R. Bajcsy. Modelling and removing radial and tangential distortions in spherical lenses. In *Multi-Image Analysis*, pages 1–21. Springer, 2001.
- [3] R. Boada Farràs. Active camera calibration for robotic systems. Master’s thesis, Universitat Politècnica de Catalunya, 2016.
- [4] J.-Y. Bouguet. Camera calibration toolbox for matlab. 2001.
- [5] N. Börlin. Nonlinear optimization: Least squares problems - the gauss newton method. 2007.
- [6] P. Corke. *Robotics, vision and control: fundamental algorithms in MATLAB® second, completely revised*, volume 118. Springer, 2017.
- [7] C. B. Duane. Close-range camera calibration. *Photogramm. Eng*, 37(8):855–866, 1971.
- [8] G. H. Golub and C. Reinsch. Singular value decomposition and least squares solutions. In *Linear Algebra*, pages 134–151. Springer, 1971.
- [9] C. G. Harris, M. Stephens, et al. A combined corner and edge detector. In *Alvey vision conference*, volume 15, pages 10–5244. Citeseer, 1988.
- [10] R. Hartley and A. Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003.

- [11] J. Heikkila, O. Silven, et al. A four-step camera calibration procedure with implicit image correction. In *cvpr*, volume 97, page 1106. Citeseer, 1997.
- [12] R. P. Jonas and M. D. Thorpe. Double gauss lens design: a review of some classics using modern methods. In *International Optical Design Conference*, page MB2. Optical Society of America, 2006.
- [13] MATLAB. *version 9.6 (R2019a)*. The MathWorks Inc., Natick, Massachusetts, 2019.
- [14] Y. Pawitan. *In all likelihood: statistical modelling and inference using likelihood*. Oxford University Press, 2001.
- [15] S. Peng and P. Sturm. Calibration wizard: A guidance system for camera calibration. *arXiv preprint arXiv:1811.03264*, 2018.
- [16] W. Qi, F. Li, and L. Zhenzhong. Review on camera calibration. In *2010 Chinese Control and Decision Conference*, pages 3354–3358. IEEE, 2010.
- [17] F. Remondino and C. Fraser. Digital camera calibration methods: considerations and comparisons. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 36(5):266–272, 2006.
- [18] A. Richardson, J. Strom, and E. Olson. Aprilcal: Assisted and repeatable camera calibration. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1814–1821. IEEE, 2013.
- [19] P. Rojtberg and A. Kuijper. Efficient pose selection for interactive camera calibration. In *2018 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 31–36. IEEE, 2018.
- [20] J. Salvi, X. Armangué, and J. Batlle. A comparative review of camera calibrating methods with accuracy evaluation. *Pattern recognition*, 35(7):1617–1635, 2002.
- [21] C. Schmid, R. Mohr, and C. Bauckhage. Evaluation of interest point detectors. *International Journal of computer vision*, 37(2):151–172, 2000.
- [22] P. Sturm, S. Ramalingam, J.-P. Tardif, S. Gasparini, J. Barreto, et al. Camera models and fundamental concepts used in geometric computer vision. *Foundations and Trends® in Computer Graphics and Vision*, 6(1–2):1–183, 2011.

- [23] R. Szeliski. *Computer vision: algorithms and applications*. Springer Science & Business Media, 2010.
- [24] B. Triggs. Autocalibration from planar scenes. In *European conference on computer vision*, pages 89–105. Springer, 1998.
- [25] B. Triggs, P. F. McLauchlan, R. I. Hartley, and A. W. Fitzgibbon. Bundle adjustment—a modern synthesis. In *International workshop on vision algorithms*, pages 298–372. Springer, 1999.
- [26] R. Tsai. A versatile camera calibration technique for high-accuracy 3d machine vision metrology using off-the-shelf tv cameras and lenses. *IEEE Journal on Robotics and Automation*, 3(4):323–344, 1987.
- [27] Z. Zhang. A flexible new technique for camera calibration. *IEEE Transactions on pattern analysis and machine intelligence*, 22, 2000.
- [28] Z. Zhang. Emerging topics in computer vision, ch. camera calibration, 2004.

Anexos

Anexo A

Creación de las traslaciones

A continuación se muestra la función desarrollada que permitió la colocación de los patrones de calibración dentro del campo de visión de la cámara para cualquier rotación.

Las entradas de este código son:

- Rci : matriz de rotación asociada al patrón.
- $Poses_vert$: representa las coordenadas, desde la referencia ubicada en el patrón, de los vértices del mismo.
- $Fov_vertices$: representa las coordendas de los límites del ángulo de visión desde la referencia cámara para un alejamiento unidad (en este caso 1 mm).

```
1
2 %Vértices en la referencia cámara (sin aplicar traslación):
3 poses_vertc = Rci*poses_vert;
4
5 %Nos creamos a continuación una variable con los vértices
6 %ordenados de menor a mayor z (alejamiento de la cámara).
7 [poses_vertcxyz , p_vxyz_ind] = sort(poses_vertc ,2);
8
9 poses_vertcz = poses_vertc(:, p_vxyz_ind (3,:)); %organizamos
%los vértices según Zc (de menor a mayor)
```

```

10 %A continuación determinamos las diferencias en Xc, Yc y Zc
   entre cada pareja de vértices del tablero.
11 %Primero , creando los índices (poses_ind) que por filas
   indica la pareja de vértices correspondientes de la que
   se calcula su diferencia.
12 poses_ind = [1 2; 1 3; 1 4; 2 3; 2 4; 3 4]; %Parejas de vé
   rtices analizadas

13
14 poses_diff_x = poses_vertc(1,poses_ind(:,1)) - poses_vertc
   (1,poses_ind(:,2));
15 poses_diff_y = poses_vertc(2,poses_ind(:,1)) - poses_vertc
   (2,poses_ind(:,2));
16 poses_diff_z = poses_vertc(3,poses_ind(:,1)) - poses_vertc
   (3,poses_ind(:,2));

17
18 % En base a qué vértice se encuentra más a la izda/arriba se
   determina un componente de la expresión que más adelante
   determinará la z_min del tablero con la rotación
   correspondiente:
19 poses_multk_x = 1/fc(1)*((poses_diff_x < 0)*cc(1) +
   poses_diff_x >= 0)*((nx-1)-cc(1));
20 poses_multk_y = 1/fc(2)*((poses_diff_y < 0)*cc(2) +
   poses_diff_y >= 0)*((ny-1)-cc(2));

21
22 % Calculamos la menor z de cada par de vértices del tablero:
23 poses_zpair = [poses_vertc(3,poses_ind(:,1)); poses_vertc(3,
   poses_ind(:,2))];
24 poses_min_zpair = min(poses_zpair);

25
26 % Calculamos el alejamiento mínimo del tablero para la
   rotación dada:
27 % 1) Se calcula la z_minima para el primero de los vértices
   de cada pareja considerada , es decir , se está tratando a
   cada par individualmente:
```

```

28 z_min_x = fc(1)/(nx-1)*(abs(poses_diff_x)-poses_diff_z.*  

29 poses_multk_x);  

30 % 2) Como el primer vértice entre cada par puede ser el más  

31 cercano o el más lejano, referenciamos todo al vértice má  

32 s cercano de cada par:  

33 z_min_x = z_min_x + (poses_diff_z < 0).*poses_diff_z;  

34 % 3) Y a continuación referenciamos todas las distancias a  

35 un mismo vértice, el vértice número 1:  

36 z_min_x2 = z_min_x - poses_min_zpair + poses_vertcz(3,1);  

37 [z_min_x_aux, z_min_x_ind] = max(z_min_x2);  

38 z_min_x = z_min_x(z_min_x_ind);  

39 poses_zv_x = poses_min_zpair(z_min_x_ind);  

40 % Todo esto ha sido para las distancias horizontales, a  

41 continuación se hacen para las verticales:  

42 z_min_y = fc(2)/(ny-1)*(abs(poses_diff_y)-poses_diff_z.*  

43 poses_multk_y);  

44 z_min_y = z_min_y + (poses_diff_z < 0).*poses_diff_z;  

45 z_min_y2 = z_min_y - poses_min_zpair + poses_vertcz(3,1);  

46 [z_min_y_aux, z_min_y_ind] = max(z_min_y2);  

47 z_min_y = z_min_y(z_min_y_ind);  

48 poses_zv_y = poses_min_zpair(z_min_y_ind);  

49 %Ahora, el máximo de z_min_x y z_min_y será el valor mínimo  

50 que se tendrá que alejar el tablero para estar dentro  

51 completamente del campo de visión de la cámara:  

52 z_min_poses = max([z_min_x_aux z_min_y_aux]);
```

```

53 if (z_min_poses == z_min_x_aux)
54     poses_zv = poses_zv_x;
55     z_min_poses = z_min_x;
56 else
57     poses_zv = poses_zv_y;
58     z_min_poses = z_min_y;
59 end
60
61 %Como criterio para colocar el origen del tablero , lo
62     intentaremos colocar lo más arriba y a la izquierda
63     posible , por ello nos falta determinar , cuál de los
64     puntos está más alejado del campo de visión tanto por
65     arriba como por la izquierda , ya que estos vértices serán
66     los que determinarán el desplazamiento necesario del
67     origen tanto en Xc como en Yc.
68
68 %Caculamos primero para ello , las z de todos los vértices:
69 poses_vertcz2 = z_min_poses + poses_vertc(3,:) - poses_zv;
70
71 %Distancia con respecto al límite superior izquierdo del
72     campo de visión:
73 poses_fov_lim = poses_vertcz2 *[-cc(1)/fc(1) -cc(2)/fc(2)];
74     %4x2 ( limites del fov tanto en x como en y para la z de
75     cada vértice)
76 poses_fov_lim_dist = poses_vertc([1 2],:) ' - poses_fov_lim ;
77 [dist_v1_rel , dist_v1_ind] = min(poses_fov_lim_dist);
78
79 %Z's del vértices de más a la izda del FOV ( z_1 ) , de más
80     arriba ( z_2 ) y la z asociada al origen ( z_1 )
81 z_1_tpose = z_min_poses + poses_vertc(3,dist_v1_ind(1)) -
82     poses_zv;
83 z_2_tpose = z_min_poses + poses_vertc(3,dist_v1_ind(2)) -
84     poses_zv;
85 z_3_tpose = z_min_poses + poses_vertc(3,1) - poses_zv;

```

```

75
76 %Las distancias que será necesario desplazar el origen para
    que quepa todo dentro del FOV:
77 dist_v1x = poses_vertc(1,1) - poses_vertc(1,dist_v1.ind(1));
78 dist_v1y = poses_vertc(2,1) - poses_vertc(2,dist_v1.ind(2));
79
80 % A continuación, definimos los pares de vértices (sus Z)
    que definen la longitud vertical y horizontal del FOV
    necesaria .
81 % Consideramos la distancia asociada a los vértices que
    determinan el alejamiento mínimo necesario , ya que por lo
    menos hasta un determinado alejamiento se mantendrán
    como las distancias que más ocupan dentro del FOV de
    todos los pares de vértices posibles .
82 poses.ind.pairx = poses.ind(z_min_x.ind,:);
83 poses.ind.pairy = poses.ind(z_min_y.ind,:);
84
85 tab_long_x = abs(poses_vertc(1,poses.ind.pairx(1)) -
    poses_vertc(1,poses.ind.pairx(2)));
86 tab_long_y = abs(poses_vertc(2,poses.ind.pairy(1)) -
    poses_vertc(2,poses.ind.pairy(2)));
87
88 % La z asociada a estos vértices la calculamos a continuación
    n. En la primera fila ponemos los vértices asociados al
    FOVx y en la segunda los asociados al FOVy:
89 poses.fov_vz = [ poses.vertcz2(poses.ind.pairx(1))
    poses.vertcz2(poses.ind.pairx(2));
    poses.vertcz2(poses.ind.pairy(1))
    poses.vertcz2(poses.ind.pairy(2))];
90
91
92 if (poses.diff_x(poses.ind.pairx) < 0)
93     poses.coef_x = [cc(1)/fc(1), ((nx-1)-cc(1))/fc(1)];
94 else
95     poses.coef_x = [( (nx-1)-cc(1))/fc(1), cc(1)/fc(1)];

```

```

96 end

97

98 if ( poses_diff_y(poses_ind_pairy) < 0)
99     poses_coef_y = [ cc(2)/fc(2) , ((ny-1)-cc(2))/fc(2) ];
100 else
101     poses_coef_y = [((ny-1)-cc(2))/fc(2) , cc(2)/fc(2) ];
102 end

103

104 %En función de las variables [k1 k2 k3 k4] creadas en la
    función simulación_test_visibility , creamos las
    siguientes variables que se utilizarán posteriormente
    para condiciones:

105 paux_x = -t_lim(2) + t_lim(4); %Nota: posibles valores =
    0,1,-1
106 paux_y = t_lim(1) - t_lim(3);

107

108 % Con esto , ya podemos calcular la posición de todos los
    tableros:

109 z_actual = poses_fov_vz;
110 z1_tpose_actual = z_1_tpose;
111 z2_tpose_actual = z_2_tpose;
112 z3_tpose_actual = z_3_tpose;

113

114 %Para las condiciones
115 pdiff_visib_x = mean([ min(poses_vertc(1,:)) max(poses_vertc
    (1,:))] - poses_vertc(1,1); %Diferencia entre el punto
    medio y el origen en Xc
116 pdiff_visib_y = mean([ min(poses_vertc(2,:)) max(poses_vertc
    (2,:))] - poses_vertc(2,1); %Diferencia entre el punto
    medio y el origen en Yc

117

118 aux = 0;

119

120 %while ( z3_tpose_actual < alejam_zlim ) && (~flag_cp)

```

```

121 while aux < 6
122
123 %aux = kk+1;
124 kkx = floor(1/ diff_tpose *(poses_coef_x(1)*z_actual(1,1)
+ poses_coef_x(2)*z_actual(1,2) - tab_long_x) +
1.001);
125 kky = floor(1/ diff_tpose *(poses_coef_y(1)*z_actual(2,1)
+ poses_coef_y(2)*z_actual(2,2) - tab_long_y) +
1.001);
126 %kk = kk + kkx*kky;
127
128 %Colocación de los checkerboards en los planos
    considerados:
129
130 for aux2 = 1:kky
131     for aux3 = 1:kkx
132         comp1 = z1_tpose_actual*fov_vertices(1,1) +
dist_v1x +(aux3-1)*diff_tpose;
133         comp2 = z2_tpose_actual*fov_vertices(1,2) +
dist_v1y +(aux2-1)*diff_tpose;
134
135     if ((comp1 + pdiff_visib_x) * paux_x >= 0) && ((comp2 + pdiff_visib_y) * paux_y >= 0) %
        Comprobaciones de visibilidad
136         aux = aux+1;
137         Tci = [comp1,comp2, z3_tpose_actual]';
138         kk = kk+1;
139         simulacion_obtener_x;
140         if (aux_safe == 0)
141             eval(['cam_', num2str(n_cam) '.tab_1.rot_',
                    ' num2str(rcont) '.tr.tr_ ', num2str(
                    aux) ' = Tci;'])
142         else
143             aux = aux-1;

```

```
144           end  
145           end  
146       end  
147   %         if flag_cp  
148   %         break;  
149   %       end  
150   end  
151  
152   z_actual = z_actual + diffz_tpose;  
153   z1_tpose_actual = z1_tpose_actual + diffz_tpose;  
154   z2_tpose_actual = z2_tpose_actual + diffz_tpose;  
155   z3_tpose_actual = z3_tpose_actual + diffz_tpose;  
156  
157 end  
158  
159 nRc_creadas = kk;
```

Anexo B

Imágenes utilizadas en la evaluación experimental

A continuación, en las Figuras B.1, B.2, B.3 y B.4, y con motivo de ilustrar los experimentos realizados, se presentan las fotos que se utilizaron en los 4 diferentes métodos de llevar a cabo la calibración y que fueron usadas en la comparación final del capítulo 5. Estas fotos dieron como resultado a los datos de la tabla 5.2 y a las Figuras 5.4 y 5.5.

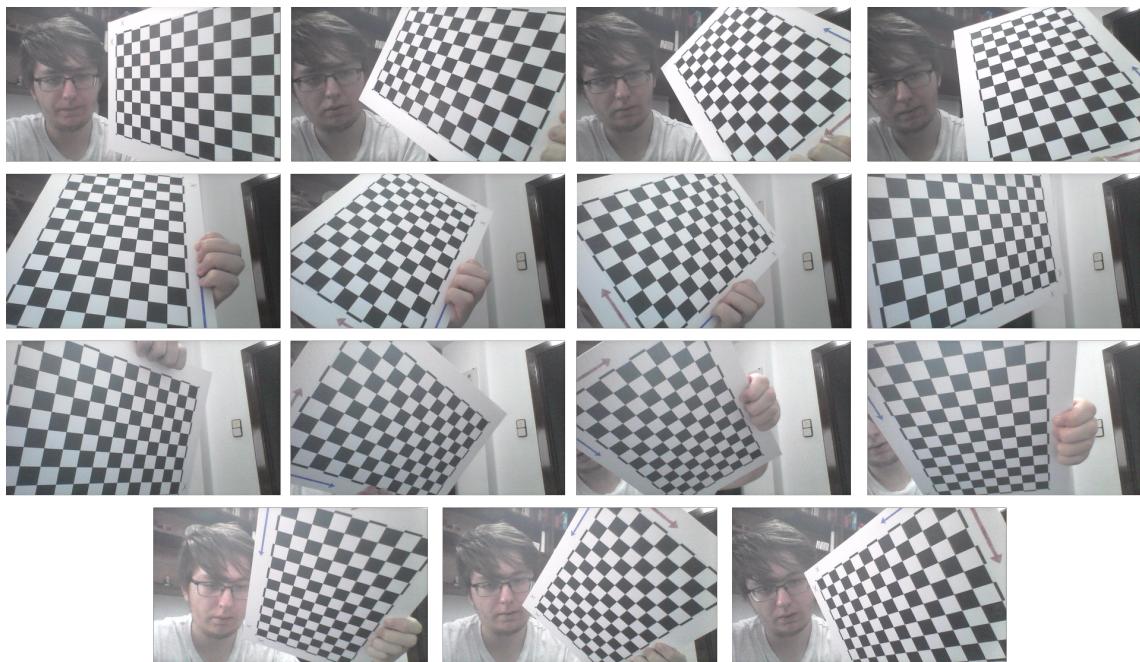


Figura B.1: Conjunto de imágenes realizadas con el método propuesto: situar lo más cerca posible los patrones de calibración y con un ángulo entre normal y eje óptico de $\theta = 40^\circ$.

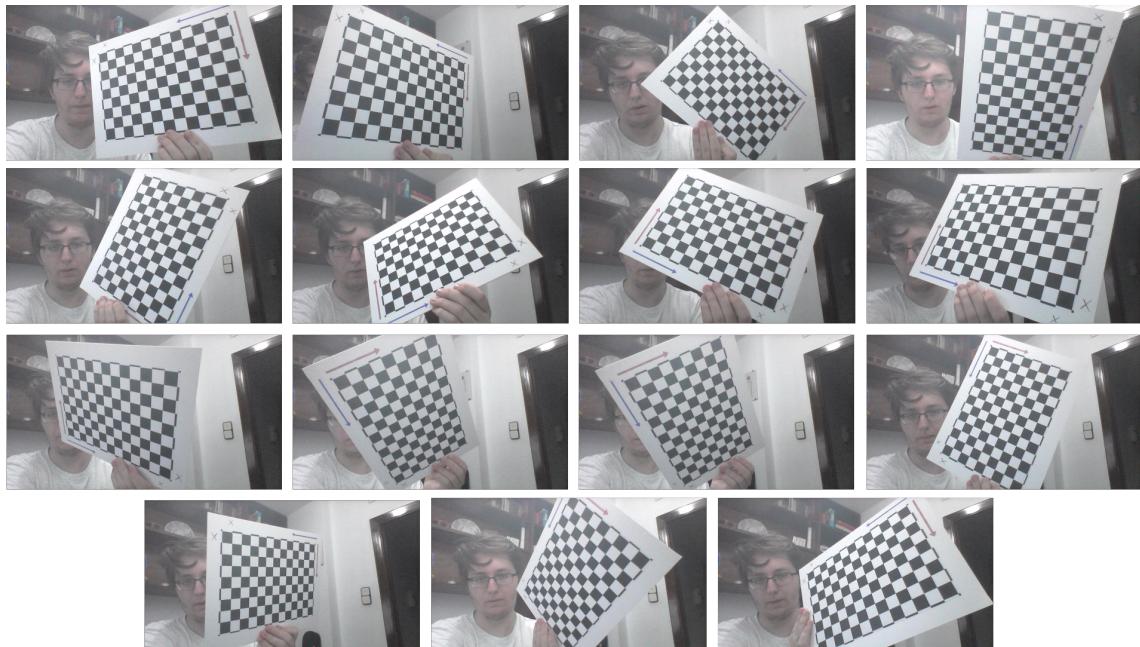


Figura B.2: Conjunto de imágenes realizadas con condiciones muy similares en cuanto a cercanía, distribución por el plano imagen del patrón y rotación sobre la normal del mismo.

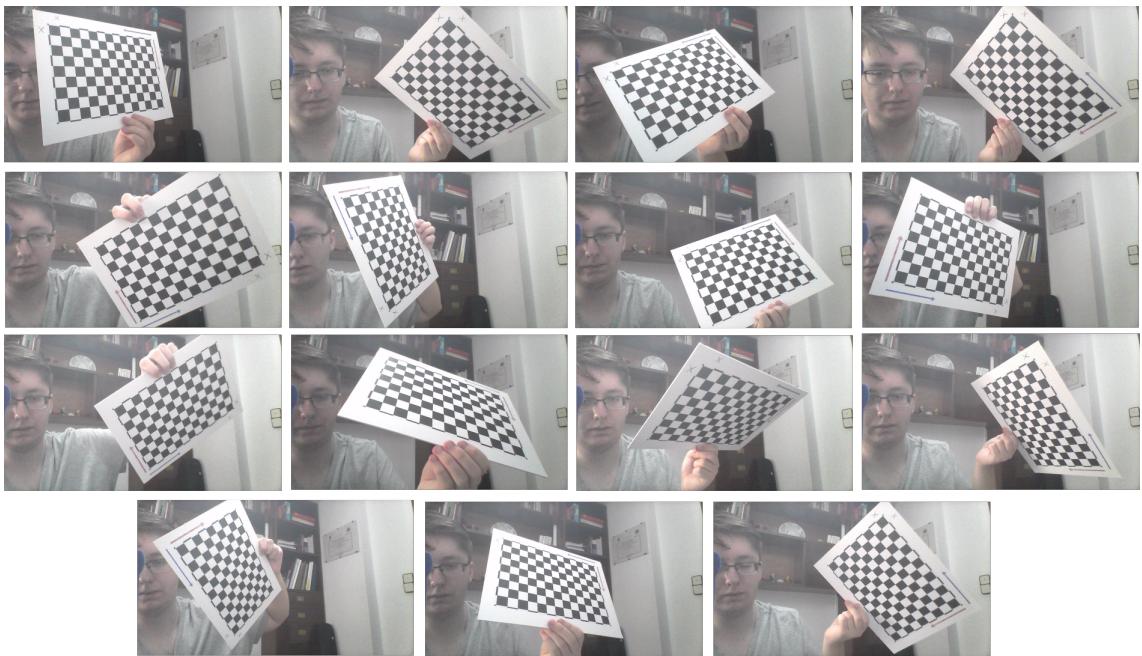


Figura B.3: Conjunto de imágenes realizadas con condiciones similares en cuanto a cercanía, distribución por el plano imagen del patrón y rotación sobre la normal del mismo.

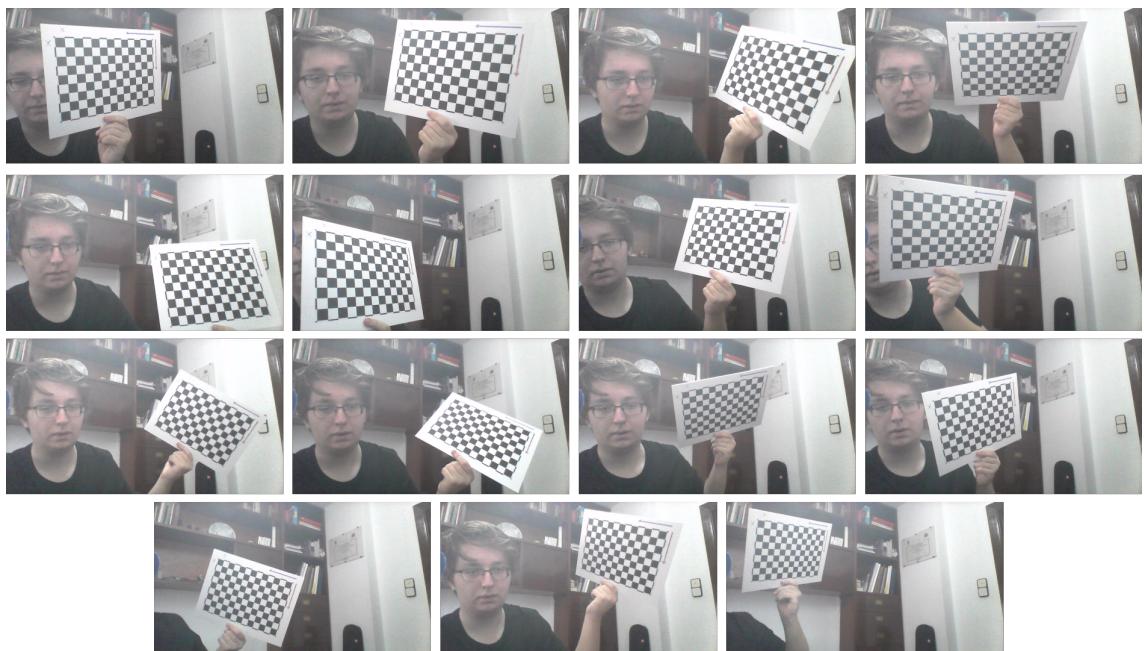


Figura B.4: Conjunto de imágenes realizadas sin condiciones similares en cuanto a cercanía y rotación sobre la normal del mismo; aunque la condición de distribuir el patrón por el plano imagen se mantiene.