

UNIVERSIDAD SAN FRANCISCO DE QUITO

Colegio de Ciencias e Ingeniería

Algoritmo de Calibración en MATLAB para Cámaras Digitales

Andrea María Witt García

Luis Miguel Prócel, MSc., Director de Tesis

Tesis de Grado presentada como requisito
para la obtención del Título de Ingeniero Electrónico

Quito, mayo de 2014

Universidad San Francisco de Quito

Colegio de Ciencias e Ingeniería

HOJA DE APROBACIÓN DE TESIS

Algoritmo de Calibración en MATLAB para Cámaras Digitales

Andrea María Witt García

Luis Miguel Procel, MSc.
Director de Tesis

Omar Aguirre S., MSc.
Director del Programa

Rene Játiva E., Ph.D (C).
Miembro del Comité de Tesis

Lionel Trojman, Ph.D.
Miembro del Comité de Tesis

Julio Ibarra, MSc.
Miembro del Comité de Tesis

Ximena Córdova, Ph.D.
Decana Escuela de Ingenierías
Colegio de Ciencias e Ingeniería

Quito, mayo de 2014

© DERECHOS DE AUTOR

Por medio del presente documento certifico que he leído la Política de Propiedad Intelectual de la Universidad San Francisco de Quito y estoy de acuerdo con su contenido, por lo que los derechos de propiedad intelectual del presente trabajo de investigación quedan sujetos a lo dispuesto en la Política.

Asimismo, autorizo a la USFQ para que realice la digitalización y publicación de este trabajo de investigación en el repositorio virtual, de conformidad a lo dispuesto en el Art. 144 de la Ley Orgánica de Educación Superior.

Firma:

Nombre: Andrea María Witt García

C. I.: 1715467849

Fecha: Quito, mayo de 2014

DEDICATORIA

Dedico esta tesis a mi familia, en especial a mi mamá y mi papá, por su apoyo incondicional y por ser mi motivación para seguir adelante siempre.

AGRADECIMIENTOS

Agradezco al Director de esta tesis, Luis Miguel Prócel, por su gran ayuda y apoyo durante el periodo de duración del proyecto de tesis. Quisiera agradecer también a todos mis profesores de Ingeniería Eléctrica y Electrónica de la USFQ, sin ellos esto no hubiera sido posible y finalmente agradezco a mis padres y mis tíos, quienes siempre estuvieron pendientes de mi durante este tiempo de trabajo.

RESUMEN

Para muchas aplicaciones modernas del campo de visión artificial, el procesamiento de imágenes, visión estéreo, realidad aumentada y otros se requiere conocer los parámetros de una cámara: la distancia focal, información acerca de los píxeles y la orientación. Este trabajo comprende la implementación de un algoritmo en MATLAB para extraer los parámetros internos y externos de una cámara digital. El proyecto estudia la técnica de calibración de Zhang (Zhang, A Flexible New Technique for Camera Calibration, 1998) y lo implementa como un algoritmo en MATLAB. Finalmente se prueba el algoritmo con datos reales y sintéticos para probar su confiabilidad y eficiencia.

ABSTRACT

For several modern subjects, such as computer vision, digital image processing, stereo vision, augmented reality and others, it is necessary to know in advance the parameters of a camera: the focal length, the characteristics of the pixels and the camera's orientation. This work includes the implementation of an algorithm in MATLAB to extract the internal and external parameters of an off-the-shelf digital camera. The project follows the calibration technique proposed by Zhang (Zhang, A Flexible New Technique for Camera Calibration, 1998) in order to program it as an algorithm using MATLAB. Ultimately, the algorithm is put under test with real and synthetic data to validate its efficiency and reliability.

TABLA DE CONTENIDO

RESUMEN	7
ABSTRACT	8
TABLAS.....	11
FIGURAS	12
Capítulo 1: Introducción.....	13
Introducción.....	13
Objetivos.....	14
Objetivo General.....	14
Objetivos Específicos	14
Justificación	15
Capítulo 2: Estudio de los Metodos de Calibración	16
Repaso General de los Métodos de Calibración.....	16
Categorización de los métodos de calibración	16
Técnicas de calibración	17
Capítulo 3: Modelo de Calibración de Zhang	21
Parámetros intrínsecos y extrínsecos de una cámara.....	21
Cámara estenopeica o pinole	21
Parámetros intrínsecos de una Cámara CCD.....	22
Parámetros Extrínsecos.....	23
Método de Zhang	25
Notación	25
Homografía.....	26
Algoritmo DLT.....	27
Resumen del Algoritmo DLT	28
Algoritmo DLT Normalizado.....	28
Resumen del Algoritmo DLT Normalizado	29
Solución en forma cerrada.....	29
Distorsión Radial	32
Estimación por Máxima Verosimilitud	34
Capítulo 4: Implementación del Algoritmo y Resultados	36
Resumen del proceso de calibración	36
Detección de los puntos de calibración	36
Coordenadas de los puntos de calibración en el espacio	37
Coordenadas de los puntos de calibración en las imágenes	38

Imágenes del plano de calibración.....	38
Detector de esquinas	39
Estimación de los parámetros de forma cerrada.....	41
Estimación de la Homografía	42
Estimación de la Matriz Intrínseca de la Cámara	43
Estimación de los factores de distorsión	44
Optimización de los parámetros con Levenberg-Marquardt	45
Simulación con datos sintéticos.....	48
Resultados finales	49
Capítulo 5: Conclusiones y Recomendaciones.....	51
Conclusiones.....	51
Recomendaciones	53
REFERENCIAS	54
ANEXO A	55
ANEXO B	56
ANEXO C.1	58
ANEXO C.2	60
ANEXO C.3	62
ANEXO C.4	64
ANEXO D	66
ANEXO E.....	67
ANEXO F.....	69
ANEXO G	71
ANEXO H	72
ANEXO I.....	73
ANEXO J.....	74
ANEXO K	75
ANEXO L.....	76
ANEXO M.....	77

TABLAS

Tabla 1: Homografías de las imágenes con el algoritmo DLT normalizado.....	42
Tabla 2: Resultado de la solución en forma cerrada de la matriz intrínseca de la cámara ..	43
Tabla 3: Resultados de la solución en forma cerrada de los parámetros extrínsecos de la cámara.....	44
Tabla 4: Parámetros de distorsión iniciales	45
Tabla 5: Resultado final de la Matriz Intrínseca.....	47
Tabla 6: Resultados finales de las matrices de rotación y los vectores de traslación.....	48
Tabla 7: Parámetros de distorsión finales.....	48
Tabla 8: Resultados de la simulación	49
Tabla 9: Resultados de la solución en forma cerrada y de Levenberg-Marquardt para 3 y 4 conjuntos de imágenes.....	50
Tabla 10: Error de los parámetros conocidos de la cámara.....	50

FIGURAS

Figura 1: Geometría de una cámara estenopeica, se proyecta el punto X por medio de una línea hacia el centro de proyección C representado por el origen del plano de coordenadas.	21
Figura 2: Transformación Euclidiana de coordenadas del mundo al marco de coordenadas de la cámara	24
Figura 3: Distorsión Radial: a. Efecto Barill, b. Efecto Cojín.....	33
Figura 4: Cuadrícula con los puntos de calibración	37
Figura 5: Imágenes del plano de calibración a distintas orientaciones.....	39
Figura 6: Esquinas detectadas en las imágenes del plano de calibración	40

CAPITULO 1: INTRODUCCION

Introducción

“Los humanos y numerosas especies animales dependen de su sistema visual para planificar o tomar acciones en el mundo” (Bigun, 2006). Gracias al sentido de la visión el ser humano ha podido procesar e interpretar información del medio que lo rodea. La observación visual de un fenómeno ha sido de gran importancia para la ciencia desde sus inicios. Sin embargo, hasta ese entonces sólo se contaba con documentación escrita o verbal y fue más tarde, con la invención de la fotografía, que se pudo documentar objetivamente los experimentos científicos. Dentro de algunas ciencias como la fotogrametría, las imágenes aéreas capturadas servían para hacer mapas topográficos y dentro de la física de partículas se usaban las imágenes de cámaras de burbujas de hidrógeno para distinguir partículas elementales. Pero, los cálculos eran complejos y era un proceso bastante largo así que las imágenes capturadas eran destinadas para la ilustración del fenómeno (Jahne, 2002).

Afortunadamente, con el avance tecnológico de las computadoras y de las cámaras de video, es posible ahora procesar las imágenes como datos. Actualmente es posible acceder a fenómenos complejos que antiguamente resultaban casi imposibles de estudiar con los métodos tradicionales de medición (Jahne, 2002). Por este motivo, se desarrollaron varias áreas en donde se aplica el procesamiento de imágenes pero son disciplinas propias, por ejemplo el campo de Visión Artificial. Esta disciplina vio la necesidad de que computadoras interactúen con el mundo real mediante la percepción visual, tanto para aplicaciones tecnológicas como para la solución de fenómenos complejos, pero que hoy en día es todavía una tarea irresuelta (Hartley & Zisserman, 2003) . Dentro de este campo, se

requiere extraer información del mundo físico a partir de imágenes, y para muchas aplicaciones esto “incluye la descripción de la manera en que la apariencia de objetos cambia cuando se los ve desde diferentes puntos de vista como una función de la forma del objeto y los parámetros de la cámara” (Hartley & Zisserman, 2003); parámetros como la posición de la cámara en el momento que capturó la fotografía o video, distorsión de píxeles y distancia focal. Es así entonces que surge la necesidad de crear algoritmos eficientes de calibración que nos permitan manipular dichos parámetros.

La calibración de una cámara, dentro del contexto de Visión Artificial, es de importancia en aplicaciones que requieren una interacción visual con el mundo real. La calibración es el proceso de determinar las características internas y ópticas de la cámara (parámetros intrínsecos) y/o la posición y orientación 3D de la cámara respecto a un sistema de coordenadas del mundo (Parámetros extrínsecos) (Tsai, 1987). De los métodos más significativos que se estudiaron para el presente proyecto, se eligió el artículo “A Flexible New Technique for Camera Calibration”, del autor Zhengyou Zhang, publicado por Microsoft Research en Diciembre 1998, para la implementación del algoritmo de calibración.

Objetivos

Objetivo General

El propósito final del proyecto es implementar en MATLAB el algoritmo de calibración propuesto por Zhang en su artículo “A Flexible New Technique for Camera Calibration”, de manera que recupere de forma confiable y eficiente los parámetros intrínsecos y extrínsecos de una cámara.

Objetivos Específicos

- Estudiar y comprender la teoría del método de calibración de Zhang.

- Tomar las fotografías adecuadas y obtener los puntos de calibración con los que se obtendrá la matriz de calibración de la cámara.
- Obtener todos los parámetros de la matriz de calibración de la cámara lo más cercanos posibles a sus valores reales.
- Realizar pruebas para analizar la confiabilidad del método

Justificación

La calibración de una cámara es necesaria para las disciplinas de hoy en día que desarrollan tecnología de punta. Por ejemplo, se requiere conocer la situación de la cámara, su posición y orientación en el mundo 3-D para la visión artificial de un robot que necesita interactuar de manera sofisticada con objetos del mundo real y poder discernir distancias. Igualmente, es necesario conocer la matriz de calibración de una cámara para aplicaciones de estereoscopía como la televisión 3D que es muy popular actualmente. Igualmente, esta información acerca de la cámara es necesaria para hallar relaciones de los puntos de una imagen con las coordenadas de los mismos en el mundo o viceversa. El desarrollo de un algoritmo de calibración preciso y relativamente eficiente con una herramienta al alcance del estudiante, como lo es MATLAB, es muy importante. Con este paso ya desarrollado se podría en un futuro crear aplicaciones como las mencionadas anteriormente o innovaciones con el uso herramientas y recursos de la estantería, lo cual es de interés para la Universidad San Francisco y para la comunidad científica.

CAPÍTULO 2: ESTUDIO DE LOS METODOS DE CALIBRACIÓN

Repaso General de los Métodos de Calibración

En el estudio de calibración de cámaras, se han ensayado diferentes métodos, cada uno con sus propios rasgos de acuerdo a un rango específico de aplicaciones (Grattoni, Pettiti, & Pollastri, 1991) y se los ha dividido en diferentes categorías de técnicas de calibración de acuerdo a sus características. Una de las metas de esta tesis era encontrar un método de calibración adecuado, que se ajuste a las necesidades y recursos del proyecto: programar un algoritmo de calibración en MATLAB que sea eficiente y no requiera de un montaje complicado para obtener los puntos de calibración. Para esto, se revisaron los artículos publicados más relevantes de este tema, la categorización de los métodos de calibración existentes y de aquellos métodos propuestos se analizaron las ventajas y desventajas de cada uno.

Categorización de los métodos de calibración

Dentro del presente tema, el artículo de Tsai, publicado en 1987 es uno de los referentes más importantes para estudios de métodos de calibración. En este artículo científico, Tsai divide los diferentes métodos existentes hasta la época en las siguientes categorías, explicando las ventajas y desventajas de las mismas, además del tipo de resolución numérica utilizado:

Categoría I: Técnicas que suponen Optimización no lineal de Gran Escala.

Categoría II: Técnicas que involucran el cálculo de la Matriz de Transformación de Perspectiva usando solución de ecuaciones lineales.

Categoría III: Método de dos planos.

Categoría IV: Técnica Geométrica.

En su artículo, Tsai explica bien las ventajas y desventajas y sobre todo las restricciones de cada categoría, sin embargo hay que tomar en cuenta que esta categorización se centra más en los métodos numéricos de resolución utilizados. Para aquella época de los años 80 en que el artículo fue publicado, la resolución con métodos numéricos representaba una restricción de gran peso, pero hoy en día existen herramientas muy eficientes para calcular múltiples variables en problemas matemáticos. Por esto, la categorización más apropiada de hoy en día de las técnicas de calibración es la de Zhang, la cual se basa en la dimensión de los objetos de calibración:

Calibración 3D o Fotogramétrica: la calibración se logra observando un objeto cuya dimensión 3D es conocida. Pueden ser dos o tres planos ortogonales, o uno solo montado en un mecanismo de traslación en el espacio. Estos métodos normalmente requieren equipos y montaje caros, pero son de gran precisión (Zhang, A Flexible New Technique for Camera Calibration, 1998).

Calibración 0D o auto-calibración: No se usa ningún objeto de calibración específico. Bastan tres imágenes para obtener correspondencias de puntos geométricos y así calcular los parámetros intrínsecos y extrínsecos (Zhang, A Flexible New Technique for Camera Calibration, 1998).

Técnicas de calibración

“A Versatile Camera Calibration Technique for High-Accuracy 3D Machine Vision Metrology Using Off-the-Shelf TV Cameras and Lenses” es el artículo de investigación realizado por Roger Tsai en 1987. El mismo trata una nueva técnica de calibración alcanzada en dos etapas. En realidad, en este artículo se presentan dos técnicas de calibración distintas basadas en el mismo principio y con el mismo montaje. La primera técnica de calibración usa una sola vista de un conjunto de puntos coplanares (plano estático en $z=0$) y la segunda es una sola vista de un conjunto de puntos no coplanares

(traslado del plano en el eje z). La primera etapa del método calcula la orientación y posición 3D de la cámara y un factor de escalamiento. La segunda etapa determina la longitud focal efectiva, los coeficientes de distorsión y posición en z de la cámara. Además, el artículo describe dos procesos para medir la precisión de la técnica de calibración: La primera forma de medición de coordenadas 3D se da a través de la triangulación estéreo, usando dos cámaras distintas. A partir de la obtención de las coordenadas 3D con la triangulación estéreo, se mide la precisión debido a la diferencia entre las coordenadas 3D conocidas de los puntos de prueba y los calculados. El segundo tipo de medición se realiza en base al radio de la zona de ambigüedad en el trazado de rayos. Finalmente, los resultados de error de calibración son prácticamente los mismos para ambos métodos de prueba que fueron aplicados a ambas técnicas de calibración y demuestran buena precisión de ambas técnicas. Las desventajas de este artículo para la implementación de este proyecto fueron el escenario y el conocimiento previo de algunos datos de la cámara. Si bien este algoritmo usa un solo plano, la parte compleja es el montaje que se usa. El escenario es complejo y costoso pues incluye iluminadores de fibra óptica, un microscopio y un montaje micrométrico en los ejes X y Y . Además, se requiere el conocimiento a priori de los factores dx y dy (distancia centro a centro entre elementos adyacentes del sensor) que normalmente vienen especificados por el fabricante de cámaras CCD (Tsai, 1987). De no tener este conocimiento, se debe usar un método de multiplano.

El siguiente artículo de interés es de los autores G.Q Wei, S.D. Ma: “A Complete Two-plane Camera Calibration Method and Experimental Comparisons”. El método hace un modelamiento de todos los tipos de distorsión de lente. Es un tipo de calibración implícita, pues no extrae explícitamente los parámetros intrínsecos y extrínsecos de una cámara, sino parámetros intermedios que sirven para aplicaciones de reconstrucción 3D y predicción de coordenadas de imagen. Esta técnica modela el problema de retroproyección

y proyección con interpolación global y modelando la distorsión del lente. El método usa un montaje simple pero a nivel de programación computacional este método es más intensivo que otros porque tiene más variables desconocidas.

El algoritmo de calibración descrito en el libro de Zisserman y Hartley se divide en dos partes. La primera parte calcula la matriz de proyección P de la cámara y la segunda parte se trata de descomponer P en las matrices K (matriz de calibración de parámetros internos) y R (matriz de orientación de la cámara) por medio del método de descomposición QR (Hartley & Zisserman, 2003). Sin embargo, este método considera el modelo de una cámara sin distorsión de lente, es decir con pixeles perfectamente cuadrados. La otra consideración que hace es que el factor de escalamiento es cero. Primero, se obtiene una solución lineal del problema de correspondencia de puntos del mundo a imagen, luego se minimiza el error geométrico y finalmente se desnormaliza la matriz resultante, es decir P . Por último, una vez calculada P se extraen las matrices K y R por medio de una descomposición de la matriz P .

Un modelo que considera la distorsión de lentes de forma completa, y que incluye la componente radial y tangencial es el de J. Weng, P. Cohen y M. Henriou en el artículo "Camera Calibration with Distortion Models and Accuracy Evaluation". El mismo está constituido por dos etapas. La primera busca una solución en forma cerrada para todos los parámetros de calibración, es decir a partir del despeje de formulas y ecuaciones, y con estos resultados como suposición principal se procede en el segundo paso a una optimización no lineal. La diferencia con otros algoritmos que se basan igualmente en dos pasos, es que en este caso se desglosan los diferentes tipos de distorsión y se los separa del procedimiento para que no haya interacciones dañinas en la optimización y el algoritmo sea más estable. La ventaja principal es que a este método se lo puede aplicar en cámaras con lente de ángulo amplio y además provee una forma de evaluar la precisión de la

calibración. El problema de esto es que el montaje resulta complicado, pues usa dos cámaras diferentes de lente de ángulo amplio, y al modelar más detalladamente la distorsión, el tratamiento computacional es más complejo (Tsai, 1987).

Finalmente, el método que se eligió fue el del artículo “Flexible Camera Calibration By Viewing a Plane From Unknown Orientations” por Zhengyou Zhang. Esta técnica se resume en los siguientes pasos: se imprime un patrón y se lo adhiere a una superficie plana. De este plano se toman varias imágenes en diferentes orientaciones, moviendo el plano o la cámara. El movimiento no se lo debe conocer necesariamente. Se detectan los puntos de calibración de las imágenes. A partir de esto, se estiman los 5 parámetros intrínsecos y todos los parámetros extrínsecos utilizando la solución cerrada en forma matricial. Luego, se modela la forma de distorsión del lente estimando los coeficientes de distorsión radial por el método lineal de mínimos cuadrados y finalmente se refinan todos los parámetros usando la técnica completa de estimación de máxima verosimilitud (Levenberg-Marquardt). La ventaja de este método es que el montaje es simple y comparado a otras técnicas que tienen montajes complejos y caros, este método provee mucha flexibilidad usando una cámara regular que está al alcance de cualquier persona sin requerirse un conocimiento previo de las características de la misma. Igualmente esta técnica resulta completa ya que calibra parámetros intrínsecos y extrínsecos y modela la distorsión del lente (desplazamiento de los puntos respecto a su posición ideal en la imagen).

CAPITULO 3: MODELO DE CALIBRACION DE ZHANG

Parámetros intrínsecos y extrínsecos de una cámara

Para comprender de qué se trata la calibración de una cámara y la descripción de parámetros intrínsecos y extrínsecos, se necesita conocer el modelo de una cámara. Este modelo matemático describe cómo los puntos del mundo real, en coordenadas 3D, son proyectados a través del aparato fotográfico a un plano de 2D representado por la imagen o fotografía.

Cámara estenopeica o pinole

Primero, es necesario hacer un modelo de la cámara estenopeica o pinole, para progresivamente ampliar este modelo a las cámaras CCD (charge-coupled device, por sus siglas en inglés), que son las cámaras digitales usadas hoy en día. Bajo este modelo, un punto en el espacio 3D, de coordenadas $\mathbf{X}=(X, Y, Z)'$ es proyectado al plano imagen por medio de una línea que une el punto \mathbf{X} con el centro de proyección como se muestra en el grafico siguiente (Hartley & Zisserman, 2003):

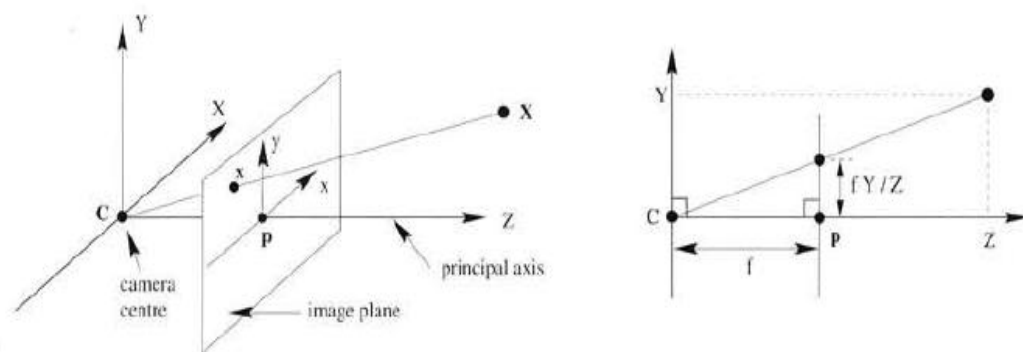


Figura 1: Geometría de una cámara estenopeica, se proyecta el punto X por medio de una línea hacia el centro de proyección C representado por el origen del plano de coordenadas (Hartley & Zisserman, 2003).

Por triángulos similares se llega a la siguiente correspondencia del espacio Euclidiano \mathbb{R}^3 al \mathbb{R}^2 :

$$(X, Y, Z)^T \rightarrow \left(\frac{fX}{Z}, \frac{fY}{Z}\right)^T \quad (2.1)$$

El centro de proyección también representa el centro de la cámara, o centro óptico. La línea que une el punto \mathbf{X} con el centro óptico de la cámara se lo conoce como el eje o rayo principal de la cámara y el punto donde el mismo interseca con el plano de la imagen se lo llama punto principal de la cámara (Hartley & Zisserman, 2003).

Esta correspondencia, mediante coordenadas homogéneas, se la puede expresar en forma matricial:

$$\begin{pmatrix} fX \\ fY \\ Z \end{pmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \quad (2.2)$$

De forma compacta, (2.2) se puede escribir como:

$$\mathbf{x} = \mathbf{P}\mathbf{X} \quad (2.3)$$

Siendo \mathbf{P} la matriz de proyección de la cámara pinole, que es el caso ideal.

Parámetros intrínsecos de una Cámara CCD

La expresión (2.2) asume que el origen de coordenadas del plano imagen está en el punto principal, pero en general este no es el caso, entonces la expresión se la generaliza como sigue, siendo las coordenadas (p_x, p_y) las coordenadas del punto principal (Hartley & Zisserman, 2003):

$$\begin{pmatrix} fX + Zp_x \\ fY + Zp_y \\ Z \end{pmatrix} = \begin{bmatrix} f & 0 & p_x & 0 \\ 0 & f & p_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \quad (2.4)$$

La matriz obtenida en la expresión (2.4) es la matriz \mathbf{K} , o *matriz intrínseca* de la cámara. Extendiendo este modelo al de las cámaras CCD, hay la posibilidad de que los píxeles no sean rectangulares necesariamente. “Si las coordenadas de la imagen son medidas en píxeles, entonces se introducen también el efecto de factores de escalamiento

desiguales en cada dirección” (Hartley & Zisserman, 2003). La transformación entonces de coordenadas del mundo a coordenadas en pixeles se hace debido a la multiplicación de los factores m_x y m_y que representan el número de pixeles por unidad de distancia. Entonces, la matriz intrínseca \mathbf{K} de una cámara CCD es la siguiente:

$$\mathbf{K} = \begin{bmatrix} \alpha & s & u_0 \\ 0 & \beta & v_0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.5)$$

Donde $\alpha = fm_x$ y $\beta = fm_y$ representan la distancia focal de la cámara en unidades de pixeles en las direcciones x y y respectivamente. Igualmente $u_0 = m_x p_x$ y $v_0 = m_y p_y$ son las coordenadas del punto principal en pixeles (Hartley & Zisserman, 2003). Además, para no perder la generalidad del caso, se añade un factor s que representa el ángulo entre los ejes x y y de los pixeles de la imagen (Bouguet).

Parámetros Extrínsecos

De forma concisa, la expresión (2.4) con la matriz \mathbf{K} de una cámara CCD (2.5) es de la forma:

$$\mathbf{x} = \mathbf{K}[\mathbf{I}|\mathbf{0}]\mathbf{X}_{cam} \quad (2.6)$$

Aquí, las coordenadas del punto \mathbf{X} , se las ha expresado en términos del *marco de coordenadas de la cámara*, pues se asume que la cámara está ubicada en el origen de un sistema de coordenadas Euclidianas donde el eje principal de la cámara apunta hacia abajo del eje Z. Sin embargo, los puntos del mundo son expresados normalmente en términos del *marco de coordenadas del mundo*. Estos dos marcos de coordenadas mencionados están relacionados mediante una rotación y traslación (Hartley & Zisserman, 2003) como se muestra en la figura 2:

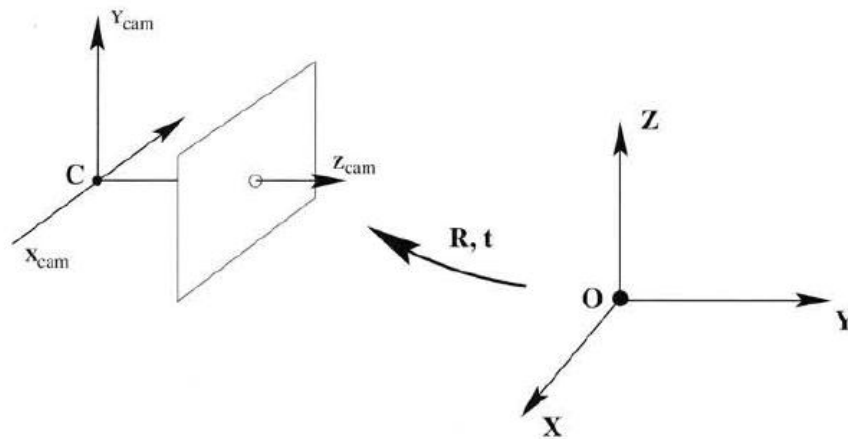


Figura 2: Transformación Euclidiana de coordenadas del mundo al marco de coordenadas de la cámara (Hartley & Zisserman, 2003).

Ahora bien, si \tilde{X} es un vector no homogéneo que representa un punto del espacio en términos del marco de coordenadas del mundo, y \tilde{X}_{cam} representa el mismo punto dentro del marco de coordenadas de la cámara, entonces se relacionan por medio de $\tilde{X}_{cam} = R(\tilde{X} - \tilde{C})$, donde \tilde{C} son las coordenadas del centro de la cámara en el marco de coordenadas del mundo y R es una matriz de rotación 3×3 que representa la orientación del marco del marco de coordenadas de la cámara (Hartley & Zisserman, 2003). Es conveniente no hacer explícito el centro de la cámara, y simplemente representar la transformación tal que: $\tilde{X}_{cam} = R\tilde{X} + t$.

Reemplazando la última expresión en (2.6) obtenemos la formula completa para la matriz de una cámara:

$$P = K[R|t] \quad (2.7)$$

Y expandiendo (2.3), se obtiene la transformación de los puntos 3D del mundo a 2D de la imagen:

$$x = K[R|t]X \quad (2.8)$$

Donde la matriz intrínseca K contiene los parámetros internos de la cámara que se describieron en la sección anterior, y además la matriz extrínseca $[R|t]$ contiene los

parámetros extrínsecos o externos de la cámara, pues son la posición y orientación de la cámara respecto al sistema de coordenadas del mundo (Hartley & Zisserman, 2003)

Método de Zhang

A continuación se da la solución propuesta por Zhang en su artículo “Flexible Camera Calibration by Viewing a Plane from Unknown Orientations”. Primero, presentamos la notación con la que se trabaja en el proyecto, después se da la solución a la homografía y a partir de esto la solución a la calibración.

Notación

Utilizando la misma notación que Zhang en su reporte técnico, volvemos a expresar la ecuación (2.8) de la misma manera que se presenta en el artículo mencionado: $\tilde{\mathbf{m}} = [u, v, 1]^T$ denota un vector homogéneo de un punto 2D y $\tilde{\mathbf{M}} = [X, Y, Z, 1]^T$ el correspondiente punto 3D. Sin embargo, no se toma en cuenta la coordenada Z, siendo $\tilde{\mathbf{M}} = [X, Y, 1]^T$, pues se asume que el plano está en $Z=0$ del sistema de coordenadas del mundo. Además existe el factor de escalamiento s .

$$s\tilde{\mathbf{m}} = \mathbf{A}[\mathbf{R} \ \mathbf{t}]\tilde{\mathbf{M}} \quad (2.9)$$

Donde \mathbf{A} , es la matriz intrínseca de la cámara y \mathbf{R} , \mathbf{t} los parámetros extrínsecos de rotación y traslación de la cámara y s un escalar arbitrario. De \mathbf{A} obtenemos la siguiente información de los parámetros intrínsecos:

$$\mathbf{A} = \begin{bmatrix} \alpha & c & u_o \\ 0 & \beta & v_o \\ 0 & 0 & 1 \end{bmatrix} \quad (2.10)$$

(u_o, v_o) Son las coordenadas del punto principal.

α, β Son la distancia focal de la imagen por los factores de escala en pixeles en los ejes u, v .

c Es el parámetro que describe el ángulo de los dos ejes de la imagen.

Homografía

De (2.9) obtenemos la expresión para la transformación proyectiva de una cámara.

La transformación proyectiva también se la conoce como homografía, ambos términos son sinónimos (Hartley & Zisserman, 2003):

$$s\tilde{\mathbf{m}} = \mathbf{H}\tilde{\mathbf{M}} \quad (2.10)$$

$$\mathbf{H} = \mathbf{A}[\mathbf{R} \mathbf{t}] \quad (2.11)$$

El problema de la Homografía es el siguiente:

Dado un conjunto de correspondencias de puntos $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i$ entre dos planos 2D, se requiere encontrar la matriz no singular 3x3 que relaciona dichos puntos:

$$\mathbf{x}' = \mathbf{H}\mathbf{x} \quad (2.12)$$

Para resolver este problema se deben considerar primero el número de correspondencias que son necesarias para la resolución de (2.12). Para esto se hacen las siguientes observaciones:

- Cada correspondencia de puntos provee tres ecuaciones con los elementos de la matriz \mathbf{H} , pero solamente dos son linealmente independientes, pues un punto tiene solamente dos grados de libertad debido a que se encuentra en un plano.
- La matriz \mathbf{H} tiene ocho grados de libertad.
- Debido a que cada pareja de puntos genera dos ecuaciones linealmente independientes con los parámetros de \mathbf{H} , entonces cuatro correspondencias son suficientes para obtener \mathbf{H} .
- Se considera que no hay 3 puntos alineados en el modelo.
- Si se tuviesen más de 4 correspondencias, no se puede establecer una \mathbf{H} válida, debido a ruido y distorsiones. Se debe buscar la mejor \mathbf{H} dados todos los puntos mediante la minimización de una función de costo.

Algoritmo DLT

Con estas premisas, se considera un algoritmo simple y lineal para una primera determinación de H usando 4 correspondencias, propuesto en el libro de Hartley y Zisserman. Este algoritmo se conoce como Transformación Lineal Directa, o DLT por sus siglas en inglés.

Partiendo de la ecuación (2.11) que está en coordenadas homogéneas, se la rescribe como un producto cruz y se la iguala a cero:

$$\mathbf{x}'_i \times H\mathbf{x}_i = 0 \quad (2.12)$$

Tenemos además la notación \mathbf{h}^{jT} que representa la j -ésima fila de la matriz H . Entonces:

$$H\mathbf{x}_i = \begin{bmatrix} \mathbf{h}^{1T} \mathbf{x}_i \\ \mathbf{h}^{2T} \mathbf{x}_i \\ \mathbf{h}^{3T} \mathbf{x}_i \end{bmatrix} \quad (2.13)$$

Igualmente tenemos las coordenadas homogéneas de la imagen $\mathbf{x}'_i = (x'_i, y'_i, w'_i)^T$, entonces el producto cruz será:

$$\mathbf{x}'_i \times H\mathbf{x}_i = \begin{pmatrix} y'_i \mathbf{h}^{3T} \mathbf{x}_i - w'_i \mathbf{h}^{2T} \mathbf{x}_i \\ w'_i \mathbf{h}^{1T} \mathbf{x}_i - x'_i \mathbf{h}^{3T} \mathbf{x}_i \\ x'_i \mathbf{h}^{2T} \mathbf{x}_i - y'_i \mathbf{h}^{1T} \mathbf{x}_i \end{pmatrix} \quad (2.14)$$

Tomando en cuenta que lo siguiente se cumple,

$$\mathbf{h}^{jT} \mathbf{x}_i = \mathbf{x}_i^T \mathbf{h}^j \quad (2.15)$$

Se obtiene el siguiente sistema de ecuaciones, donde ya se ha eliminado la tercera ecuación al no ser linealmente independiente:

$$\begin{pmatrix} \mathbf{0}^T & -w'_i \mathbf{x}_i^T & y'_i \mathbf{x}_i^T \\ w'_i \mathbf{x}_i^T & \mathbf{0}^T & -x'_i \mathbf{x}_i^T \end{pmatrix} \begin{pmatrix} \mathbf{h}_1 \\ \mathbf{h}_2 \\ \mathbf{h}_3 \end{pmatrix} = 0 \quad (2.16)$$

El sistema es de tipo $\mathbf{A}_i \mathbf{h} = 0$, donde la matriz \mathbf{A} es una matriz de dimensiones 2×9 , y \mathbf{h} un vector de 9 elementos. A partir de (2.16) es posible construir un sistema de 8 ecuaciones con 9 incógnitas, pues con 4 correspondencias se puede armar la matriz \mathbf{A} , siendo su tamaño 8×9 .

Resumen del Algoritmo DLT

OBJETIVO: dadas $n \geq 4$ correspondencias 2D-2D, determine la matriz de la homografía \mathbf{H} , tal que $\mathbf{x}_i' = \mathbf{H}\mathbf{x}_i$.

1. Para cada correspondencia $\mathbf{x}_i \leftrightarrow \mathbf{x}_i'$ calcular la matriz \mathbf{A}_i .
Solamente se necesitan las dos primeras filas de la misma.
2. Construir las n matrices \mathbf{A}_i 2×9 y formar una sola matriz $2n \times 9$.
3. Descomponer en valores singulares la matriz \mathbf{A} . El vector singular asociado al menor valor singular será la solución \mathbf{h} .
4. Determinar \mathbf{H} a partir de \mathbf{h} por filas.

Algoritmo DLT Normalizado

A veces las coordenadas de una imagen vienen dadas con el origen en el centro y en otras ocasiones el origen está en la esquina superior izquierda de la imagen (Hartley & Zisserman, 2003). Igualmente, el factor de escala se selecciona de forma arbitraria y la correspondencia de coordenadas puede venir dada en diferentes unidades. El algoritmo DLT es sensible a estos factores y el resultado es variante (Hartley & Zisserman, 2003). Para resolver esto, se recomienda hacer una normalización a los datos antes de aplicar el algoritmo y aplicar transformaciones de similitud, pues así el resultado no se ve afectado por la escala y orientación escogidas (Hartley & Zisserman, 2003).

El algoritmo DLT con normalización propuesto por Zisserman es el siguiente:

Resumen del Algoritmo DLT Normalizado

OBJETIVO: dadas $n \geq 4$ correspondencias 2D-2D, determine la matriz de la homografía H , tal que $\mathbf{x}_i' = H\mathbf{x}_i$.

1. *Normalización de \mathbf{x}* : Calcular una transformada de similitud T , que consista de una traslación y escalamiento, que transforme los puntos \mathbf{x}_i a un nuevo conjunto de puntos $\tilde{\mathbf{x}}_i$ tal que el centroide de dichos puntos $\tilde{\mathbf{x}}_i$ es el origen de coordenadas $(0,0)^T$, y su distancia promedio del origen es $\sqrt{2}$.
2. *Normalización de \mathbf{x}'* : Calcular una transformación de similitud T' para los puntos en la segunda imagen, transformando los puntos \mathbf{x}_i' a $\tilde{\mathbf{x}}_i'$.
3. *DLT*: aplicar el algoritmo DLT descrito en la sección anterior a las correspondencias $\tilde{\mathbf{x}}_i \leftrightarrow \tilde{\mathbf{x}}_i'$ para obtener una homografía \tilde{H} .
4. *Desnormalización*: Aplicar $H = (T')^{-1}\tilde{H}T$

Con este algoritmo, se calcula H para cada imagen y su correspondiente plano en el espacio. Una vez estimada H , se procede a la solución de la calibración, la cual se describe en las siguientes secciones de este capítulo.

Solución en forma cerrada

Expandiendo la expresión (2.11) obtenemos la relación siguiente; donde \mathbf{r}_1 y \mathbf{r}_2 son los vectores de las columnas de la matriz de rotación R , y \mathbf{h}_1 , \mathbf{h}_2 y \mathbf{h}_3 son los vectores de las columnas de la homografía:

$$[\mathbf{h}_1 \ \mathbf{h}_2 \ \mathbf{h}_3] = \lambda A[\mathbf{r}_1 \ \mathbf{r}_2 \ \mathbf{t}] \quad (2.17)$$

Debido a la ortonormalidad \mathbf{r}_1 y \mathbf{r}_2 , obtenemos las dos siguientes restricciones dada una homografía (Zhang, Flexible Camera Calibration By Viewing a Plane From Unknown Orientations, 1998):

$$\mathbf{h}_1^T A^{-T} A^{-1} \mathbf{h}_2 = 0 \quad (2.18)$$

$$\mathbf{h}_1^T A^{-T} A^{-1} \mathbf{h}_1 = \mathbf{h}_2^T A^{-T} A^{-1} \mathbf{h}_2 \quad (2.19)$$

Por otro lado se simplifica la notación de $A^{-T} A^{-1}$ por la matriz \mathbf{B} :

$$\mathbf{B} = A^{-T} A^{-1} = \begin{bmatrix} B_{11} & B_{12} & B_{13} \\ B_{12} & B_{22} & B_{23} \\ B_{13} & B_{23} & B_{33} \end{bmatrix} \quad (2.20)$$

Notando la simetría de la matriz \mathbf{B} , se puede definir dicha matriz como un vector \mathbf{b} de 6D:

$$\mathbf{b} = [B_{11}, B_{12}, B_{22}, B_{13}, B_{23}, B_{33}]^T \quad (2.21)$$

Además, siendo $\mathbf{h}_i = [h_{i1}, h_{i2}, h_{i3}]^T$ el i -ésimo vector columna de la homografía \mathbf{H} , tenemos la relación siguiente proveniente de (2.19):

$$\mathbf{h}_i^T \mathbf{B} \mathbf{h}_j = \mathbf{v}_{ij}^T \mathbf{b} \quad (2.22)$$

Con $\mathbf{v}_{ij} = [h_{i1}h_{j1}, h_{i1}h_{j2} + h_{i2}h_{j1}, h_{i2}h_{j2}, h_{i3}h_{j1} + h_{i1}h_{j3}, h_{i3}h_{j2} + h_{i2}h_{j3}, h_{i3}h_{j3}]^T$.

Con (2.21) y (2.22) se pueden describir las restricciones de (2.18) y (2.19) como dos ecuaciones homogéneas:

$$\begin{bmatrix} \mathbf{v}_{12}^T \\ (\mathbf{v}_{11} - \mathbf{v}_{22})^T \end{bmatrix} \mathbf{b} = 0 \quad (2.23)$$

Que de forma compacta, es un sistema de tipo $\mathbf{Vb} = 0$.

Al tener n imágenes del plano de calibración, se puede formar un sistema de $2n$ ecuaciones, es decir, una matriz $2n \times 6$. Si se cuenta con más de 3 imágenes del plano, es posible obtener una única solución \mathbf{b} (Zhang, Flexible Camera Calibration By Viewing a Plane From Unknown Orientations, 1998). La solución (2.23) resulta ser el vector singular derecho de \mathbf{V} asociado con el valor singular más pequeño.

Una vez calculado \mathbf{b} entonces, es posible obtener la matriz \mathbf{A} de parámetros intrínsecos de la cámara y una vez calculada \mathbf{A} se obtiene también la matriz de rotación y traslación de la cámara. El procedimiento se explica más adelante. Para \mathbf{A} sin embargo, una vez obtenido \mathbf{b} se calcula cada parámetro de forma independiente con las siguientes formulas propuestas en el artículo de Zhang:

$$v_o = (B_{12}B_{13} - B_{11}B_{23}) / (B_{11}B_{22} - B_{12}^2)$$

$$\lambda = B_{33} - [B_{13}^2 + v_o(B_{12}B_{13} - B_{11}B_{23})] / B_{11}$$

$$\alpha = \sqrt{\lambda / B_{11}}$$

$$\beta = \sqrt{\lambda B_{11} / (B_{11}B_{22} - B_{12}^2)}$$

$$c = -B_{12}\alpha^2\beta / \lambda$$

$$u_o = \frac{cv_o}{\alpha} - B_{13}\alpha^2 / \lambda$$

Una vez estimados los parámetros internos de la cámara, se puede formar la matriz de parámetros intrínsecos \mathbf{A} . Con \mathbf{A} se puede hacer un primer cálculo de la matriz de rotación y el vector de traslación con las relaciones siguientes:

$$\mathbf{r}_1 = \lambda \mathbf{A}^{-1} \mathbf{h}_1, \mathbf{r}_2 = \lambda \mathbf{A}^{-1} \mathbf{h}_2, \mathbf{r}_3 = \mathbf{r}_1 \times \mathbf{r}_2$$

$$\text{Con } \lambda = \frac{1}{\|\mathbf{A}^{-1} \mathbf{h}_1\|} = \frac{1}{\|\mathbf{A}^{-1} \mathbf{h}_2\|}.$$

Con los vectores columna de la matriz de rotación, se forma una matriz de rotación cercana a R , que la llamaremos Q . Se requiere obtener la mejor matriz R dada Q . Este problema se resuelve haciendo la descomposición del valor singular de Q de tal forma que $Q = USV^T$, y la solución está dada por $R = UV^T$ (Zhang, A Flexible New Technique for Camera Calibration, 1998).

Distorsión Radial

La distorsión geométrica afecta la posición de los puntos en el plano de la imagen. Para corregir la distorsión es necesario analizar las varias fuentes que la causan y los diferentes tipos de distorsión. Normalmente se consideran el error radial y el tangencial. El error radial es causado por fallas en la curvatura del lente, y por otro lado el mal ensamblaje de la cámara y lente causa distorsión radial y también tangencial (Weng, Cohen, & Herniou, 1992). En el método de Zhang, solamente se consideran los factores de distorsión radial, pues se asume que la cámara que se usa es de buena calidad y la distorsión tangencial introducida es casi nula.

La distorsión radial hace que un punto de la imagen se desplace desde su posición ideal hacia adentro o hacia afuera. Un desplazamiento negativo, es decir hacia adentro, crea el efecto de disminución de escala, conocido como efecto de barril y un desplazamiento positivo, hacia afuera, hace que la escala aumente. Este último fenómeno se lo conoce como el efecto de cojín (Weng, Cohen, & Herniou, 1992).

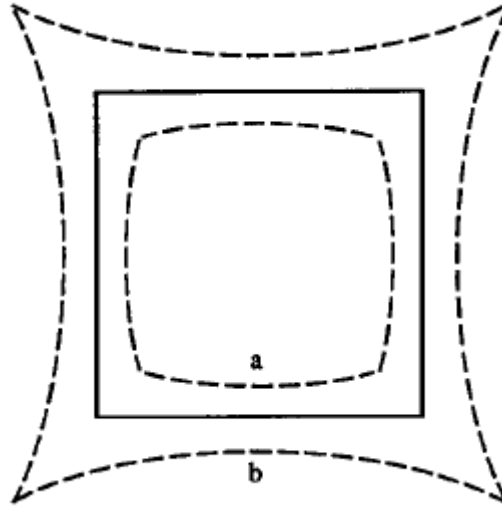


Figura 3: Distorsión Radial: a. Efecto Barill, b. Efecto Cojín (Weng, Cohen, & Herniou, 1992)

Las coordenadas ideales de la imagen, en pixeles, son u y v y sus correspondientes coordenadas reales (con distorsión) se las denota como \check{u} y \check{v} . Igualmente, x, y son las coordenadas ideales normalizadas de la imagen (estas son las coordenadas dentro del marco de coordenadas de la cámara, después de aplicar la matriz de rotación y traslación al punto en el espacio) y \check{x}, \check{y} sus correspondientes coordenadas reales con distorsión. La relación de estas coordenadas viene dada por las siguientes ecuaciones, siendo k_1 y k_2 los coeficientes de distorsión radial (Zhang, Flexible Camera Calibration By Viewing a Plane From Unknown Orientations, 1998):

$$\check{x} = x + x[k_1(x^2 + y^2) + k_2(x^2 + y^2)^2]$$

$$\check{y} = y + y[k_1(x^2 + y^2) + k_2(x^2 + y^2)^2]$$

Ya que el centro de la distorsión radial es el mismo que el punto principal, obtenemos por medio de las relaciones $\check{u} = u_0 + \alpha\check{x} + c\check{y}$ y $\check{v} = v_0 + \beta\check{y}$:

$$\check{u} = u + (u - u_0)[k_1(x^2 + y^2) + k_2(x^2 + y^2)^2] \quad (2.24)$$

$$\check{v} = v + (v - v_0)[k_1(x^2 + y^2) + k_2(x^2 + y^2)^2] \quad (2.25)$$

A partir de estas expresiones se obtienen dos ecuaciones por cada punto en cada imagen:

$$\begin{bmatrix} (u - u_o)(x^2 + y^2) & (u - u_o)(x^2 + y^2)^2 \\ (v - v_o)(x^2 + y^2) & (v - v_o)(x^2 + y^2)^2 \end{bmatrix} \begin{bmatrix} k_1 \\ k_2 \end{bmatrix} = \begin{bmatrix} \check{u} - u \\ \check{v} - v \end{bmatrix} \quad (2.26)$$

Dado m puntos en n imágenes, se obtiene un sistema de $2mn$ ecuaciones, que en notación matricial corresponde a un sistema $\mathbf{D}\mathbf{k} = \mathbf{d}$, donde $\mathbf{k} = [k_1, k_2]$. La solución viene dada por (Zhang, Flexible Camera Calibration By Viewing a Plane From Unknown Orientations, 1998):

$$\mathbf{k} = (\mathbf{D}^T \mathbf{D})^{-1} \mathbf{D}^T \mathbf{d} \quad (2.27)$$

Con los dos factores de distorsión obtenidos con la solución (2.26) ya está solucionada la calibración de forma cerrada. El siguiente paso en el método de Zhang es hacer una optimización para alcanzar el menor error posible y acercarnos a los valores reales de la cámara.

Estimación por Máxima Verosimilitud

La solución de forma cerrada obtenida en la sección anterior solamente minimiza un funcional de distancia algebraica, que no es muy significativo. Es por esto, que Zhang en su método propone refinar los resultados mediante una optimización de máxima verosimilitud. Esto se obtiene, minimizando el siguiente funcional (Zhang, Flexible Camera Calibration By Viewing a Plane From Unknown Orientations, 1998):

$$\sum_{i=1}^n \sum_{j=1}^m \|\mathbf{m}_{ij} - \check{\mathbf{m}}(\mathbf{A}, k_1, k_2, \mathbf{R}_i, \mathbf{t}_i, \mathbf{M}_j)\|^2 \quad (2.28)$$

En el reporte técnico de Zhang se utiliza el algoritmo de Levenberg-Marquardt para minimizar el error. Este es un método para resolver problemas de mínimos cuadrados no lineales (Gavin, 2013).

Como podemos observar en el funcional, lo que se minimiza es la diferencia entre los puntos de la imagen obtenidos y los puntos estimados de acuerdo a los parámetros intrínsecos y extrínsecos calculados con la solución en forma cerrada, incluidos los parámetros de distorsión.

Para aplicar la minimización de Levenberg-Marquardt, es además necesario parametrizar la matriz de rotación \mathbf{R} . Esto se hace por medio de la fórmula de Rodrigues, que obtiene de dicha matriz, el vector del eje de rotación y el ángulo de rotación con las formulas siguientes (Hartley & Zisserman, 2003):

$$\theta = \cos^{-1}[(\text{trace}(\mathbf{R}) - 1) / 2]$$

$$\mathbf{r} = \theta \begin{bmatrix} \mathbf{R}(3,2) - \mathbf{R}(2,3) \\ \mathbf{R}(1,3) - \mathbf{R}(3,1) \\ \mathbf{R}(2,1) - \mathbf{R}(1,2) \end{bmatrix} \frac{1}{2 \sin \theta}$$

CAPÍTULO 4: IMPLEMENTACIÓN DEL ALGORITMO Y RESULTADOS

En el anterior capítulo se describió detalladamente la teoría del modelo de una cámara. A partir de este modelo, se propusieron las soluciones matemáticas y quedó justificada la solución al problema de la calibración. A continuación se presenta la forma en que la solución fue implementada en MATLAB, siguiendo paso a paso el método anteriormente descrito y finalmente se presentan los resultados finales del algoritmo.

Resumen del proceso de calibración

Zhang resume el proceso de calibración con los siguientes pasos, los cuales van de acuerdo a la teoría explicada en el anterior capítulo (Zhang, Flexible Camera Calibration By Viewing a Plane From Unknown Orientations, 1998):

1. Imprimir un patrón y fijarlo a una superficie plana.
2. Tomar fotos del patrón a diferentes orientaciones, moviendo la cámara o el patrón; las coordenadas de los puntos del patrón deben ser conocidas.
3. Detectar los puntos de calibración de las imágenes.
4. Aplicar la solución de forma cerrada para calcular los parámetros intrínsecos y extrínsecos de la cámara.
5. Obtener los factores de distorsión.
6. Refinar los parámetros minimizando el funcional (2.28).

A continuación se presenta la implementación de cada uno de estos pasos.

Detección de los puntos de calibración

El primer paso del proceso de calibración es imprimir un plano que contenga los puntos de calibración. El patrón escogido fue una cuadrícula de 16 rectángulos cuyas esquinas son los puntos de calibración:

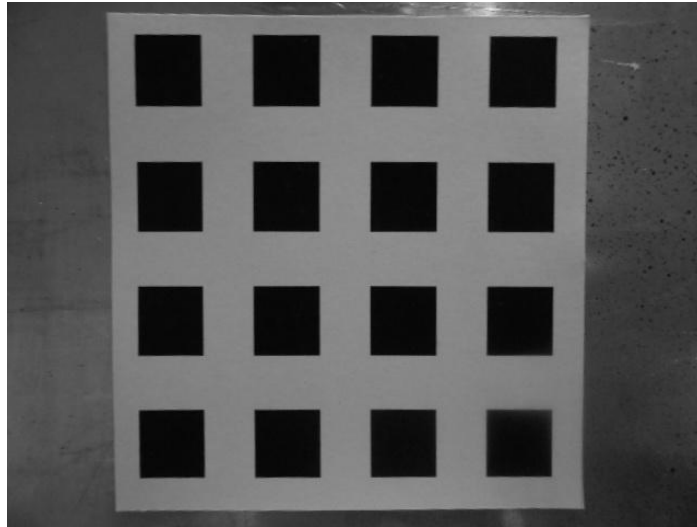


Figura 4: Cuadrícula con los puntos de calibración

Coordenadas de los puntos de calibración en el espacio

El patrón debe tener una geometría simple, pues se debe fijar un marco de coordenadas en el espacio y conocer la posición de cada punto del plano. Para este caso, el origen del marco de coordenadas del espacio está ubicado en la esquina inferior izquierda del rectángulo ubicado en la esquina inferior izquierda de la cuadrícula. Ya que el plano es simétrico, es fácil medir la posición de cada punto en unidades de centímetros con una regla o cuadrícula.

El plano de calibración tiene 4 rectángulos en dirección X y 4 cuadrados en la dirección Y de 1.4cm x 1.3cm cada uno y donde cada esquina es un punto de calibración, es decir que en total hay 64 puntos de calibración con un espaciamiento de 1.15cm en dirección X y 1.05cm en dirección Y. Las dimensiones del plano son 9.05cm x 8.35cm.

Es importante asumir que el patrón está en el plano $Z=0$. Entonces, las ubicaciones 2D de los puntos, en el marco de coordenadas del espacio, están dadas por las coordenadas que se muestran en el Anexo B.

Coordenadas de los puntos de calibración en las imágenes

Imágenes del plano de calibración

Una vez elegido el patrón con los puntos de calibración y se ha fijado el marco de coordenadas espaciales se puede proceder a tomar las fotos del plano a distintas orientaciones. Este algoritmo se aplica a las cámaras digitales regulares. La cámara no necesita ser profesional. En este proyecto se usó una cámara Nikon COOLPIX de 16.0 Megapíxeles. Sin embargo, la resolución a la que se tomaron las fotos fue de 640 x 480 píxeles. De acuerdo a los resultados obtenidos en el reporte técnico de Zhang, es recomendable aplicar el algoritmo a partir de 2 imágenes, pues el error se reduce significativamente.

Para fotografiar el plano a distintas orientaciones, se utilizó un montaje para el plano, el cual permitía rotar al mismo en todas las direcciones y además una traslación en todos los ejes. En cuanto a la cámara, la misma fue ajustada a un trípode para mantenerla fija y simplemente mover el plano.

Las orientaciones del plano no deben ser conocidas, pero hay que intentar que el ángulo de rotación del mismo sea superior a 5° ya que los resultados fallan al ser los planos casi paralelos (Zhang, Flexible Camera Calibration By Viewing a Plane From Unknown Orientations, 1998). En este caso, se obtuvieron 4 imágenes que se presentan a continuación:

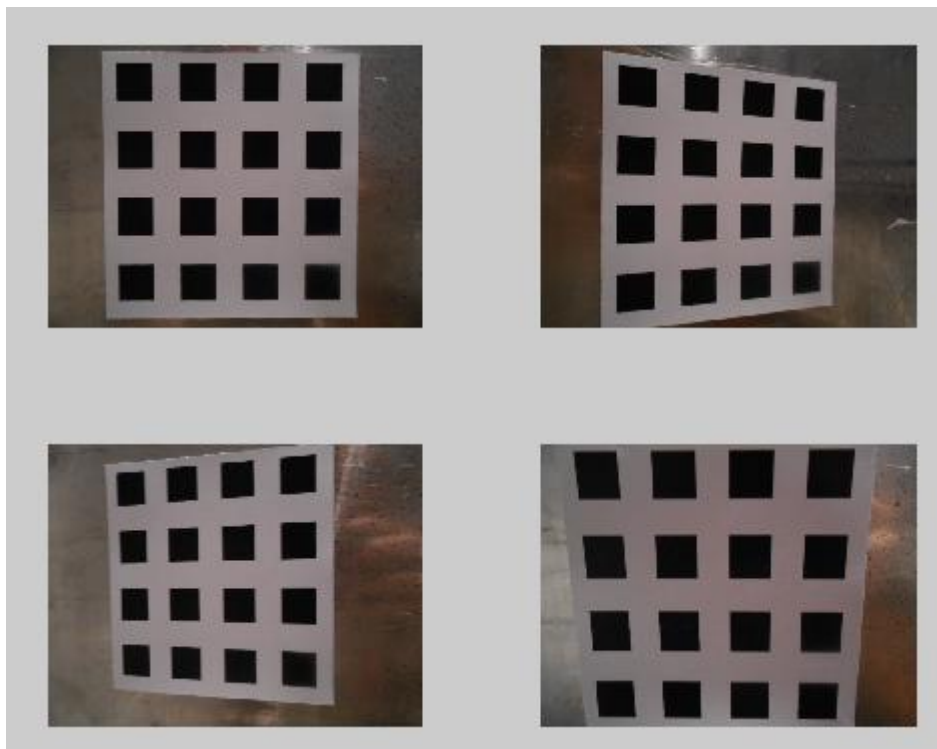


Figura 5: Imágenes del plano de calibración a distintas orientaciones

Detector de esquinas

Como se expresó antes, los puntos de calibración son las esquinas de los rectángulos de la cuadrícula. Para obtener sus coordenadas en píxeles, hay que procesar las imágenes con un detector de esquinas. El objetivo de este proyecto es crear un algoritmo de calibración usando la herramienta MATLAB, por lo cual se usó un detector de esquinas del toolbox de Computer Vision. Para usar el detector, primero fue necesario transformar las imágenes a escala de grises, con la función implementada de MATLAB `rgb2gray`.

El detector de esquinas viene dado en forma de objeto, el cual en MATLAB es llamado por `vision.cornerdetector`. Este objeto encuentra las esquinas en una imagen en escala de grises y retorna la ubicación de las esquinas en una matriz de coordenadas x y y . El mismo tiene 3 métodos para detectar las esquinas; el usuario puede elegir que método quiere que el objeto utilice entre: La detección de esquinas de Harris & Stephens, el método del mínimo valor propio por Shi & Thomasi o el método por comparación de intensidad local de Rosten & Drummond (Matlab, 2012).

Las coordenadas de las esquinas detectadas con el objeto se las adjunto al Anexo C; aquí se presentan las imágenes con las esquinas que fueron detectadas por los comandos de MATLAB:

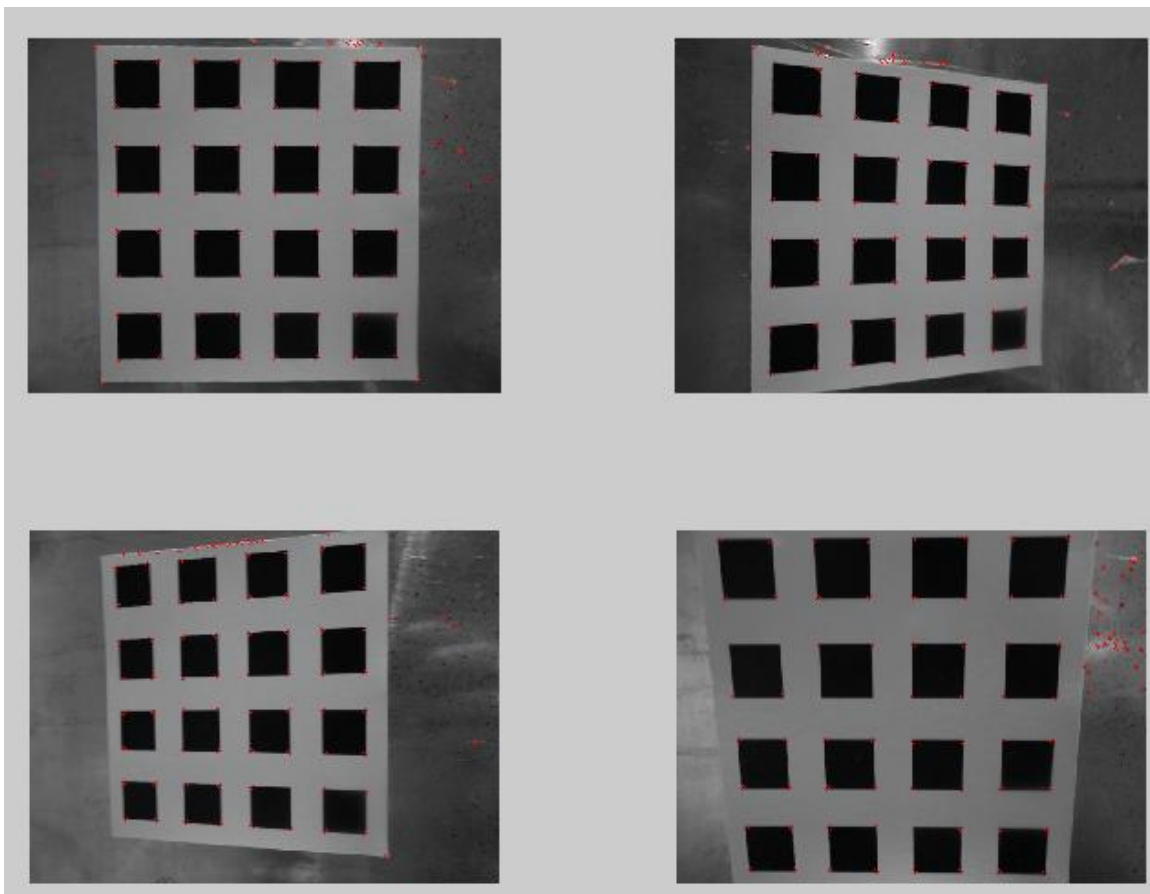


Figura 6: Esquinas detectadas en las imágenes del plano de calibración

En las imágenes se pueden ver las esquinas de los rectángulos del plano de calibración pero, ya que son imágenes pequeñas, no son visibles los círculos rojos que las marcan. Un problema en este paso es que el detector de esquinas no solo detecta los puntos de calibración, sino también detecta esquinas que se hallan fuera de la cuadrícula. Esto representa un problema en cuanto a la automatización del algoritmo mas no en los resultados, ya que en la matriz que el objeto retorna con las coordenadas de las esquinas hay que eliminar a mano aquellos puntos que no pertenecen al plano.

La rutina que se programó para convertir las imágenes a escala de grises y detectar las esquinas se encuentra en el Anexo D.

Estimación de los parámetros de forma cerrada

El programa principal que se ejecuta para dar la solución de forma cerrada es la función `calibZ2`. Se lo llama en forma de función como sigue:

```
[H0,A0,R0,t0]=calibZ2(Xw,Xi,n).
```

Los parámetros de entrada son los siguientes:

```
Xw: Matriz Np x 3 de puntos homogéneos 3D en Z=0
Xi: Matriz Np x 3 de puntos homogéneos 2D
n: número de planos
```

Donde N_p es el número total de puntos, en este caso se tienen 64 puntos en cada plano, y dado que se trabaja con 4 imágenes, son en total 256 puntos, con $n = 4$. La función retorna las homografías de cada imagen, la matriz intrínseca de la cámara y la matriz de rotación y el vector de traslación de cada imagen (4 matrices de rotación y 4 vectores de traslación):

```
H0: Conjunto de homografías 3nx3
A0: Matriz Intrínseca 3x3
R0: conjunto de matrices de rotación 3nx3
t0: conjunto vectores de traslación 3xn
```

A continuación se explica cómo se implementó la función principal mediante la descripción de las subrutinas que se usan. Primero, se calcula la homografía mediante el algoritmo DLT normalizado, segundo se calcula la matriz B por medio de la relación (2.23) para obtener los parámetros internos de la cámara, y finalmente se calculan las matrices de rotación y los vectores de traslación como se explicó en el capítulo 2.

La implementación en MATLAB del programa principal está en el Anexo E.

Estimación de la Homografía

De acuerdo a la revisión teórica del capítulo 2, lo primero que se necesita estimar es la homografía que viene dada por la matriz \mathbf{H} . Se concluyó que el mejor método para este fin es el algoritmo DLT normalizado.

Se programó una subrutina llamada: `dltmH(pw, pi)`. En esta función se implementó el algoritmo DLT con normalización de datos. La misma recibe los siguientes parámetros de entrada con la condición de que n sea mayor o igual a 4:

`pw`: matriz $n \times 3$ con coordenadas 3D homogéneas en $Z=0$
`pi`: matriz $n \times 3$ con coordenadas 2D homogéneas

Esta subrutina es el primer paso dentro del programa principal de calibración, y debido a que se tienen 4 imágenes, cuyas coordenadas corresponden al mismo plano a 4 orientaciones distintas, entonces se obtiene un conjunto de 4 homografías:

Tabla 1: Homografías de las imágenes con el algoritmo DLT normalizado

H1	42.0103	-0.928	124.2766
	0.1273	-48.831	438.6526
	0.0003	-0.003	1.011
H2	45.0323	0.2417	117.1671
	5.8322	-45.5101	411.255
	0.0223	-0.0015	0.9052
H3	31.9604	-1.9361	142.3514
	-4.1565	-44.7646	427.8228
	-0.0164	-0.004	1.0907
H4	47.0849	-6.7578	110.8802
	-0.6174	-59.0131	506.5413
	-0.0028	-0.0201	1.0966

El detalle del algoritmo DLT normalizado que se implementó en MATLAB se encuentra en el Anexo F.

Estimación de la Matriz Intrínseca de la Cámara

Una vez obtenidas las homografías de cada imagen, se procede a estimar los parámetros internos de la cámara. El objetivo es resolver el sistema (2.23), que con los resultados de las homografías ya puede lograrse. En realidad, dentro del programa principal de calibración no hay una subrutina que calcule los parámetros internos de la cámara, pero si esta implementada una subrutina que calcula los elementos de la matriz del sistema (2.23). Dicha subrutina es llamada con la función $v = \text{const}(H)$ (ver Anexo E).

Con lo que esta subrutina devuelve, se forma el sistema (2.23) y se lo resuelve mediante la descomposición en valores singulares de la matriz v con la función de MATLAB $\text{svd}(V)$. Una vez obtenido el vector b del sistema, se calculan todos los parámetros internos de la cámara y se forma la matriz intrínseca A_0 .

A partir del resultado de la matriz intrínseca, se forman las matrices de rotación de cada imagen y además los vectores de traslación, de acuerdo a las ecuaciones dadas en la solución de forma cerrada que Zhang propone en su artículo y que están detalladas en el capítulo anterior. Igualmente, esta parte esta implementada en el programa principal que se programó en MATLAB como `calibZ2` (Ver Anexo E).

Los resultados de la solución de forma cerrada son los siguientes:

Tabla 2: Resultado de la solución en forma cerrada de la matriz intrínseca de la cámara

A0	580.0973	-0.223	326.1796
	0	667.4294	241.1683
	0	0	1

Tabla 3: Resultados de la solución en forma cerrada de los parámetros extrínsecos de la cámara

R1	1	0.001	0.0047	t1	-4.9031
	0.0008	-0.9991	0.0418		4.0415
	0.0048	-0.0418	-0.9991		13.9978
R2	0.946	0.0178	0.3237	t2	-4.4599
	0.0099	-0.9996	0.0261		4.2016
	0.3241	-0.0215	-0.9458		13.1551
R3	0.9691	-0.0186	-0.2462	t3	-5.5421
	-0.0032	-0.998	0.0627		3.7201
	-0.2468	-0.0599	-0.9672		16.4351
R4	0.9994	-0.0059	-0.0334	t4	-5.1367
	0.0023	-0.9707	0.2402		4.3802
	-0.0338	-0.2401	-0.9701		13.2437

Estimación de los factores de distorsión

La estimación de los factores de distorsión se hizo igualmente en base a la teoría presentada por Zhang en su reporte técnico. Usando (2.24) y (2.25) para cada punto en todas las imágenes, se obtuvo un sistema de 512 ecuaciones homogéneas. Usando MATLAB, se implementó la función `distorRad` para calcular los parámetros de distorsión k_1 y k_2 (ver Anexo G). El mismo usa una subrutina llamada `distR` que devuelve la matriz D y el vector d del sistema (2.26) para una sola imagen. Posteriormente, dentro del programa principal `distorRad`, se usa la subrutina mencionada para calcular D y d de cada imagen y formar el sistema completo de (2.26), se resuelve el sistema y la función devuelve los parámetros de distorsión radial.

El uso de la función es de la forma siguiente: $k = \text{distorRad}(X_w, X_i, n, A, R, t)$.

Los parámetros de entrada son:

X_w : Matriz $(N_p \times n) \times 3$ de coordenadas homogéneas de puntos 3D en $Z=0$
 X_i : Matriz $(N_p \times n) \times 3$ de coordenadas homogéneas de puntos 2D

n: Numero de planos
 A: Matriz Intrínseca 3x3
 R: Matrices de rotación 3nx3
 t: Vectores de traslación 3xn

Donde N_p es el número de puntos en cada plano, que en este caso es 64. Por otro lado la salida de la función es k , un vector 2x1 que contiene los coeficientes de distorsión radial.

Los resultados obtenidos de los parámetros de distorsión con las 4 imágenes usadas, y los resultados de los parámetros internos de la cámara obtenidos de la sección anterior son los siguientes:

Tabla 4: Parámetros de distorsión iniciales

Parámetros de distorsión	
k_1	-0.0519
k_2	0.2392

Dado que el primer parámetro de distorsión es negativo, de acuerdo a la teoría vista, la cámara estaría introduciendo un efecto de barril. Sin embargo, estos resultados no pueden ser conclusivos ya que para el cálculo de los parámetros internos y externos de la cámara se asumió que no había distorsión. El verdadero valor de distorsión resulta al final del algoritmo cuando se han refinado todos los parámetros.

Optimización de los parámetros con Levenberg-Marquardt

Con los parámetros dados por la solución en forma cerrada y los parámetros de distorsión, se procede a hacer una optimización no lineal. El método de Levenberg-Marquardt es una técnica usada para resolver problemas no lineales de mínimos cuadrados. Estos problemas de mínimos cuadrados se dan cuando se ajusta una función parametrizada a un conjunto de datos medidos minimizando la suma de los cuadrados de los errores entre los datos y la función (Gavin, 2013).

Como se explicó en el capítulo 2, el funcional que se debe minimizar es (2.28), es decir se minimiza la suma del cuadrado de los errores entre los puntos medidos y los puntos estimados de la imagen, para así obtener los parámetros intrínsecos y extrínsecos de la cámara lo más cercanos a la realidad.

Para llevar a cabo este objetivo y hacer uso de los recursos disponibles, se buscó una función de MATLAB que nos permita hacer la optimización. El toolbox de Optimization de MATLAB, contiene la función `lsqcurvefit` que resuelve problemas no lineales de mínimos cuadrados, con la opción de usar el método de Levenberg Marquardt. Se llama la función de la siguiente forma: $X = \text{lsqcurvefit}(\text{FUN}, X_0, \text{XDATA}, \text{YDATA})$; la función empieza en X_0 y encuentra los coeficientes X que mejor se ajustan a la función no lineal FUN según YDATA . La función FUN acepta los parámetros de entrada X_0 y XDATA y retorna un vector de valores F , donde F es del mismo tamaño que YDATA , evaluado en X_0 y XDATA (MATLAB, 2012).

La programación del programa principal para hacer la optimización se la hizo mediante un programa principal en forma de función llamado `LM` (Anexo H), cuyo uso va de acuerdo a $[A, R, t, k] = \text{LM}(X_i, X_w, A_0, R_0, t_0, k_0, n)$. Los parámetros de entrada y de salida se explican a continuación:

Input: X_i : Matriz $(N_p \times n) \times 3$ de coordenadas homogéneas de puntos 2D
 X_w : Matriz $(N_p \times n) \times 3$ de coordenadas homogéneas de puntos 3D en $Z=0$
 A_0 : matriz intrínseca 3×3
 R_0 : conjunto de matrices de rotación, 3×3 (n = número de planos)
 t_0 : conjunto de vectores de traslación, $3 \times n$
 k_0 : vector de distorsión radial
 n : número de planos

Output: A : Matriz Intrínseca final 3×3
 R : conjunto de matrices de rotación finales, 3×3 (n = número de planos)
 t : conjunto de vectores de traslación finales, $3 \times n$
 k : vector de distorsión radial final

El primer paso de este programa usa una subrutina llamada `vectorP` (Anexo I). La misma toma las matrices A , R , t y el vector k de parámetros de distorsión y los une para convertirlos en un solo vector. Así mismo, utiliza la subrutina `RodriguesVector` (Anexo J) que convierte las matrices de rotación en vectores por medio del eje de rotación y su ángulo, siguiendo la fórmula de Rodrigues que se mostró al final del capítulo 2.

El segundo paso de la rutina LM es llamar a la función `lsqcurvefit`:

```
pref = lsqcurvefit(@MyfunCalib,p,Xw,Xi,[],[],opt)
```

Aquí p representa los parámetros iniciales de la función `MyfunCalib` (Anexo K). Entonces, la función `MyfunCalib` evaluada en p , toma los puntos X_w (puntos de calibración en el espacio) y retorna los valores calculados, minimizando con Levenberg-Marquardt el error entre los valores calculados por la función `MyfunCalib` y X_i (puntos de calibración de las imágenes) en el sentido de mínimos cuadrados. Entonces, la función `lsqcurvefit` retorna los parámetros `pref`, que son los parámetros p ya refinados que mejor se ajustan a la curva de la correspondencia entre X_w y X_i .

Una vez obtenidos los parámetros refinados en forma de vector, se reconstruyen las matrices, y se usa la subrutina `RodriguesMatrix` (Anexo L) para recuperar las matrices de rotación a partir de los vectores de rotación refinados.

Los resultados finales son los siguientes:

Tabla 5: Resultado final de la Matriz Intrínseca

A	486.346	-0.4334	328.4878
	0	559.55	238.9911
	0	0	1

Tabla 6: Resultados finales de las matrices de rotación y los vectores de traslación

R1	1	0.001	0.0061	t1	-4.964
	0.0007	-0.9994	0.0358		4.0582
	0.0061	-0.0358	-0.9993		12.6576
R2	0.9603	0.0171	0.2783	t2	-4.8606
	0.0113	-0.9997	0.0222		4.2921
	0.2786	-0.0182	-0.9602		12.0868
R3	0.9792	-0.017	-0.2022	t3	-5.4566
	-0.0057	-0.9984	0.0565		3.7544
	-0.2029	-0.0542	-0.9777		14.7954
R4	0.9996	-0.0043	-0.0271	t4	-5.1596
	0.0012	-0.9797	0.2004		4.2153
	-0.0274	-0.2003	-0.9793		11.9808

Tabla 7: Parámetros de distorsión finales

Parámetros de distorsión	
k_1	-0.0594
k_2	0.0928

Simulación con datos sintéticos

Para probar que el algoritmo funciona bien y da resultados satisfactorios, se realizó una prueba con los mismos datos sintéticos propuestos por Zhang en su reporte técnico. Se simula una cámara de matriz intrínseca:

$$A = \begin{bmatrix} 1250 & 1.09083 & 255 \\ 0 & 900 & 255 \\ 0 & 0 & 1 \end{bmatrix}$$

La resolución de la imagen es 512 x 512 y el plano es un patrón de cuadros de 10 x 14 = 140 esquinas que serán los puntos de calibración. El tamaño del plano es de 18cm x 25 cm.

Además, en esta simulación se usan 3 planos cada uno con un vector de rotación y otro de traslación específicos:

$$r_1 = [20^\circ, 0, 0]^T, t_1 = [-9, -12.5, 500]^T$$

$$r_2 = [0, 20^\circ, 0]^T, t_2 = [-9, -12.5, 510]^T$$

$$r_3 = \frac{1}{\sqrt{5}}[-30^\circ, -30^\circ, -15^\circ]^T, t_3 = [-10.5, -12.5, 525]^T$$

Con estos datos, se construyeron las coordenadas de los puntos de calibración en el espacio y se simularon los puntos de calibración de las 3 imágenes. Convirtiendo los ángulos a radianes, y usando la fórmula de Rodrigues implementada en MATLAB para recuperar las matrices, se aplicó todo el algoritmo sin tomar en cuenta la distorsión y los parámetros internos y externos de la cámara que se recuperaron resultaron ser exactamente los mismos. Esto demostró el buen funcionamiento del algoritmo de calibración.

Tabla 8: **Resultados de la simulación**

	Resultado inicial			Resultado final			
A0	1275.7	1.1132	254.91	A	1250	1.0908	255
	0	918.49	255		0	900	255
	0	0	1		0	0	1

Resultados finales

Se probó el algoritmo con 3 y 4 imágenes, obteniendo los resultados de los parámetros internos de la cámara de la solución en forma cerrada y después de aplicar la optimización con Levenberg-Marquardt. Se probó el error RMS, que mide el cuadrado de la diferencia entre los puntos de calibración de la imagen medidos y los puntos proyectados con los parámetros internos y externos obtenidos. Igualmente para este cálculo, se programó la rutina `RMS2` (Anexo M). A continuación se presentan los resultados finales del proyecto:

Tabla 9: Resultados de la solución en forma cerrada y de Levenberg-Marquardt para 3 y 4 conjuntos de imágenes

Parámetro	3 imágenes		4 imágenes	
	inicial	final	inicial	final
α	592.9	506.47	580.0973	486.346
β	681.31	581.71	667.4294	559.55
c	-0.26572	-0.50444	-0.223	-0.4334
U_0	327.19	331.37	326.1796	328.4878
V_0	237.48	238.92	241.1683	238.9911
k_1	-0.070466	-0.078631	-0.051941	-0.059449
k_2	0.2071	0.13309	0.23922	0.092789
RMS (pixels)	15.988	1.5416	18.637	1.5848

Tabla 10: Error de los parámetros conocidos de la cámara

Parámetro	Valores Teóricos	Error 3 imágenes	Error 4 imágenes
c	90°	0.07%	0.06%
U_0	320	3.44%	2.65%
V_0	240	0.45%	0.42%

En la tabla 9, se observa que el error inicial está entre 15 y 18 pixeles de diferencia promedio, y que este error se reduce significativamente después de aplicar la optimización. La ubicación de los puntos de la imagen medidos y los estimados difiere en menos de 2 pixeles. Se observa también que el error RMS aumenta en el conjunto de 4 imágenes, sin embargo esto no es conclusivo ya que se necesitarían más pruebas para notar una tendencia. Por otro lado, en la tabla 10 se observa que el error relativo es más bajo para el conjunto de 4 imágenes en comparación a los valores teóricos de la cámara.

CAPÍTULO 5: CONCLUSIONES Y RECOMENDACIONES

Conclusiones

En los capítulos anteriores se presentó el modelo de calibración de Zhang: la teoría, las ecuaciones y además el modelo de una cámara CCD para comprender el algoritmo de calibración. Después se expuso la implementación del algoritmo en MATLAB y los resultados de las pruebas con datos reales y con datos simulados. El error RMS final resulto ser menor a 2 pixeles, que es bastante satisfactorio pues es un error muy bajo respecto a la resolución de las imágenes de 640 x 480 pixeles y con un conjunto de 4 imágenes el error relativo respecto a los valores teóricos de los parámetros internos de la cámara es menor.

Ahora bien, se puede afirmar que la matriz intrínseca de la cámara usada para el proyecto es la siguiente:

$$\mathbf{A} = \begin{bmatrix} \alpha & c & u_0 \\ 0 & \beta & v_0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 486.346 & -0.4334 & 328.4878 \\ 0 & 559.55 & 238.9911 \\ 0 & 0 & 1 \end{bmatrix}$$

Y los factores de distorsión finales:

$$\mathbf{k} = [-0.0594, 0.0928]^T$$

De los resultados finales de los parámetros internos, se concluye que la cámara toma imágenes con pixeles rectangulares, pues los pixeles en el arreglo CCD no son cuadrados si la relación de aspecto β/α es diferente de 1 (Bouguet). De acuerdo a los resultados finales obtenidos, la tasa de aspecto de la cámara usada es $\beta/\alpha = 1.15$.

Adicionalmente, el parámetro interno de oblicuidad c viene dado por la fórmula $c = \alpha \cot \varphi$ donde φ es el ángulo entre los ejes x y y de los pixeles de la imagen (Bouguet). Despejando el ángulo de la formula, el resultado obtenido es $\varphi = -89.95^\circ$. Es decir que el ángulo es casi 90° , lo cual demuestra que en efecto los pixeles son

rectangulares y el signo negativo muestra la orientación de los mismos. Por último, la distancia focal viene dada por α y β en unidades de píxeles.

Respecto a los coeficientes de distorsión finales, se puede concluir que la cámara introduce un ligero efecto de barril debido al signo negativo del primer factor de distorsión radial. La imagen adquiere entonces el efecto de barril como se muestra en la Figura 3. Para aplicaciones de procesamiento de imágenes, estos factores son muy importantes ya que a partir de esto se puede hacer correcciones para tener imágenes de mejor calidad.

Finalmente, el algoritmo se puso a prueba con una cámara de vigilancia IP, modelo COMPRO IP570 / IP570P PTZ, que se usa actualmente para el proyecto UAV Galápagos, pues sería de mucha utilidad conocer los parámetros internos y externos de la cámara para el objetivo final de este proyecto: vigilar las islas Galápagos contra la pesca ilegal. Sin embargo, al ser una cámara con lente de gran apertura se introduce mucha distorsión para tener un ángulo más amplio de visión. Los valores obtenidos de distorsión fueron muy altos y los resultados finales no fueron satisfactorios. Se concluye entonces que este algoritmo funciona solamente para cámaras digitales regulares de uso personal. Para obtener los parámetros intrínsecos y extrínsecos de este tipo de cámaras de lente de gran apertura, es necesario ajustar el modelo de la distorsión.

En conclusión, el algoritmo funciona conforme a la expectativa pues el mismo es eficiente y confiable. Además, los recursos que utiliza el algoritmo de calibración son mínimos pues se requiere solamente de un plano de calibración y una cámara y la herramienta MATLAB está al alcance de los estudiantes y de la comunidad científica.

Con este algoritmo de calibración, se podrán desarrollar proyectos avanzados en el campo de Visión Artificial, Procesamiento de Imágenes, aplicaciones de estereoscopia y otras dado que el conocimiento de los parámetros internos de una cámara es la base de muchas aplicaciones que estos campos desarrollan hoy en día.

Recomendaciones

En este algoritmo, la detección de esquinas en las imágenes del plano no está automatizada. Se recomienda mejorar esta parte del algoritmo. Lo ideal sería tener un detector de esquinas que cuente automáticamente cuantos cuadrados hay en el patrón y retorne las esquinas solamente de los cuadrados del plano y en orden. Así, el usuario solo debería preocuparse de introducir las imágenes.

Es recomendable desarrollar la parte de interfaz gráfica. Esto se podría alcanzar usando la herramienta GUI de MATLAB, así los futuros usuarios no tendrían dificultad al usar el algoritmo. Por ahora el mismo se limita al uso de personas que conocen de programación y uso de funciones en MATLAB.

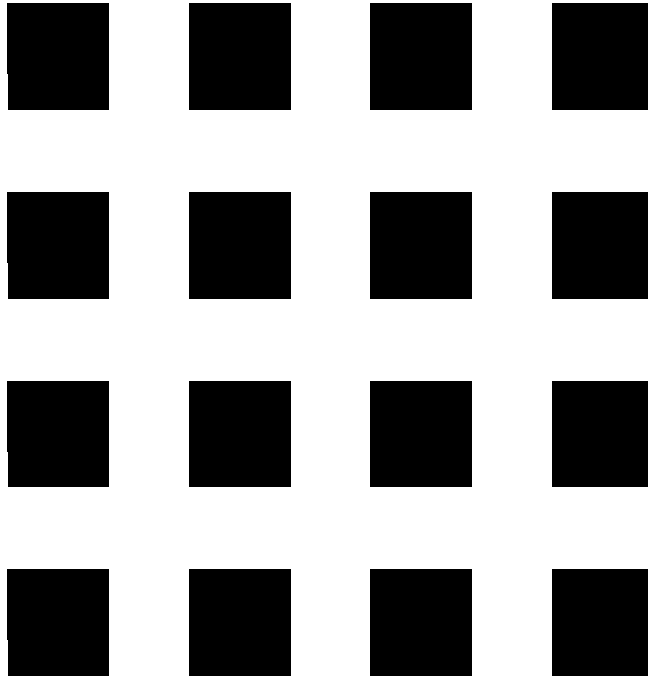
Finalmente, sería útil implementar un modelo de distorsión para realizar también la calibración de cámaras de gran apertura, como la cámara del proyecto UAV Galápagos y las cámaras IP de vigilancia.

REFERENCIAS

- Bacakoglu, H., & Kamel, M. S. (1997). A Three-Step Camera Calibration Method. *IEEE Transactions on Instrumentation and Measurement*, Vol.46, No.5, 1165-1172.
- Bigun, J. (2006). *Vision with Direction: A Systematic Introduction to Image Processing and Computer Vision*. Heidelberg: Springer.
- Bouguet, J.-Y. (n.d.). *Complete Camera Calibration Toolbox for MATLAB*. Retrieved Febrero 11, 2013, from Computational Vision at Caltech: http://www.vision.caltech.edu/bouguetj/calib_doc/index.html
- Forsyth, D., & Ponce, J. (2002). *Computer Vision: A Modern Approach*.
- Gavin, H. P. (2013, Octubre 9). *Duke University Department of Civil and Environmental Engineering*. Retrieved Febrero 18, 2013, from Sitio web de Duke University: <http://people.duke.edu/~hpgavin/ce281/lm.pdf>
- Grattoni, P., Pettiti, G., & Pollastri, F. (1991). *Geometric Camera Calibration: A Comparison of Methods*. Torino: IEEE.
- Hartley, R., & Zisserman, A. (2003). *Multiple View Geometry in Computer Vision*. Cambridge: Cambridge University Press.
- Jahne, B. (2002). *Digital Image Processing*. Springer.
- MATLAB. (2012). Documentacion de MATLAB.
- Tsai, R. (1987). A Versatile Camera Calibration Technique for High-Accuracy 3D Machine Vision Metrology Using Off-the-Shelf TV Cameras and Lenses. *IEEE Journal of Robotics and Automation*, 323-343.
- Wei, G.-Q., & Ma, S. D. (1993). A Complete Two-plane Camera Calibration Method and Experimental Comparisons. *IEEE*.
- Weng, J., Cohen, P., & Herniou, M. (1992). Camera Calibration with Distortion Models and Accuracy Evaluation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol.14, No. 10, 965-980.
- Zhang, Z. (1998). *A Flexible New Technique for Camera Calibration*. Microsoft Research.
- Zhang, Z. (1998). *Flexible Camera Calibration By Viewing a Plane From Unknown Orientations*. Redmond, WA: Microsoft Research.
- Zhang, Z. (2004). Camera Calibration with One-Dimensional Objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence* Vol.26, No.7.

ANEXO A

Plano de Calibración



ANEXO B

Coordenadas Homogéneas del Plano en el marco de coordenadas del espacio. X y Y son las coordenadas en el espacio de las esquinas del plano, medidas en unidades de centímetros. El origen del plano se ubica en la esquina inferior izquierda del plano.

X	Y	
0	0	1
1.4	0	1
2.55	0	1
3.95	0	1
5.1	0	1
6.5	0	1
7.65	0	1
9.05	0	1
0	1.3	1
1.4	1.3	1
2.55	1.3	1
3.95	1.3	1
5.1	1.3	1
6.5	1.3	1
7.65	1.3	1
9.05	1.3	1
0	2.35	1
1.4	2.35	1
2.55	2.35	1
3.95	2.35	1
5.1	2.35	1
6.5	2.35	1
7.65	2.35	1
9.05	2.35	1
0	3.65	1
1.4	3.65	1
2.55	3.65	1
3.95	3.65	1
5.1	3.65	1
6.5	3.65	1
7.65	3.65	1
9.05	3.65	1
0	4.7	1
1.4	4.7	1
2.55	4.7	1
3.95	4.7	1
5.1	4.7	1

6.5	4.7	1
7.65	4.7	1
9.05	4.7	1
0	6	1
1.4	6	1
2.55	6	1
3.95	6	1
5.1	6	1
6.5	6	1
7.65	6	1
9.05	6	1
0	7.05	1
1.4	7.05	1
2.55	7.05	1
3.95	7.05	1
5.1	7.05	1
6.5	7.05	1
7.65	7.05	1
9.05	7.05	1
0	8.35	1
1.4	8.35	1
2.55	8.35	1
3.95	8.35	1
5.1	8.35	1
6.5	8.35	1
7.65	8.35	1
9.05	8.35	1

ANEXO C.1

Coordenadas Homogéneas del plano a distintas orientaciones en el marco de referencia de la imagen. Las coordenadas u y v están medidas en pixeles.

Imagen 1

u	v	
124	434	1
180	433	1
229	433	1
286	433	1
335	433	1
392	433	1
441	432	1
498	432	1
123	375	1
180	374	1
229	374	1
286	374	1
335	374	1
392	374	1
441	374	1
498	374	1
122	322	1
179	322	1
228	322	1
285	322	1
335	322	1
392	322	1
441	322	1
499	322	1
122	261	1
178	261	1
228	261	1
285	261	1
334	261	1
392	261	1
442	262	1
499	262	1
122	209	1
178	209	1
228	209	1

285	209	1
335	209	1
393	209	1
442	209	1
500	209	1
121	148	1
178	148	1
227	148	1
285	148	1
335	148	1
393	148	1
443	148	1
500	149	1
120	94	1
177	94	1
228	95	1
286	94	1
335	94	1
393	95	1
444	95	1
501	96	1
119	31	1
177	32	1
227	32	1
285	32	1
336	32	1
394	33	1
445	33	1
502	34	1

ANEXO C.2

Imagen 2

u	v	
130	454	1
192	447	1
242	442	1
297	437	1
342	432	1
391	427	1
431	422	1
474	418	1
131	392	1
193	387	1
242	384	1
297	380	1
342	377	1
391	374	1
431	371	1
475	368	1
132	337	1
193	334	1
242	332	1
298	330	1
342	328	1
391	326	1
431	324	1
476	322	1
132	274	1
193	273	1
243	273	1
298	272	1
342	272	1
392	272	1
433	272	1
477	272	1
133	219	1
194	220	1
243	221	1
299	222	1
343	223	1
393	224	1
433	225	1
478	226	1

133	155	1
195	159	1
245	161	1
300	164	1
345	167	1
394	170	1
434	173	1
478	175	1
134	100	1
196	105	1
246	109	1
301	114	1
346	118	1
396	122	1
436	125	1
480	130	1
134	35	1
197	43	1
247	49	1
303	55	1
348	61	1
397	68	1
438	73	1
481	78	1

ANEXO C.3

Imagen 3

u	v	
131	392	1
173	394	1
214	397	1
260	400	1
304	403	1
355	406	1
403	409	1
457	413	1
130	342	1
172	344	1
213	346	1
259	347	1
304	349	1
354	351	1
402	353	1
457	356	1
128	298	1
170	299	1
212	299	1
258	301	1
303	301	1
353	303	1
402	304	1
457	305	1
128	247	1
169	247	1
211	246	1
257	246	1
302	245	1
353	245	1
401	245	1
457	246	1
126	202	1
168	201	1
210	200	1
256	198	1
301	197	1
352	196	1
401	195	1
457	194	1

125	150	1
166	148	1
208	145	1
254	143	1
300	141	1
351	138	1
400	136	1
456	133	1
123	104	1
165	102	1
207	99	1
254	95	1
300	92	1
351	89	1
400	85	1
456	82	1
120	51	1
162	47	1
206	43	1
252	38	1
298	34	1
349	29	1
400	24	1
456	19	1

ANEXO C.4

Imagen 4

u	v	
102	462	1
161	462	1
212	463	1
273	464	1
325	464	1
387	464	1
439	465	1
502	466	1
96	405	1
157	404	1
209	405	1
272	405	1
324	406	1
388	407	1
441	407	1
505	408	1
91	350	1
153	350	1
206	350	1
270	350	1
323	351	1
389	352	1
443	352	1
509	353	1
86	286	1
149	286	1
203	286	1
268	286	1
323	287	1
390	287	1
446	288	1
513	288	1
80	227	1
145	226	1
200	227	1
267	227	1
323	227	1
391	227	1
448	227	1
517	228	1

73	157	1
140	157	1
197	157	1
265	157	1
323	156	1
392	156	1
451	156	1
522	156	1
67	93	1
135	92	1
193	92	1
264	91	1
322	91	1
394	91	1
454	90	1
527	90	1
59	15	1
129	15	1
189	14	1
262	14	1
322	13	1
396	12	1
458	12	1
534	10	1

ANEXO D

SCRIPT EN MATLAB PARA DETECCION DE ESQUINAS

```
%Script para convertir imagenes a escala de grises
B=imread('foto4.jpg');
B=rgb2gray(B);
imwrite(B,'foto4gray.tif')
imshow(B)

%Script para detectar las esquinas en la imagen

I = im2single(imread('foto4gray.tif')); %normaliza la imagen

% Select FAST algorithm by Rosten & Drummond, se selecciona metodo de
% deteccion de esquinas
hcornerdet = vision.CornerDetector('Method', 'Local intensity comparison
(Rosten & Drummond)', 'MaximumCornerCount', 90);

pts = step(hcornerdet, I); %coordenadas de las esquinas en matriz [x y]

color = [1 0 0]; % [red, green, blue] marca con 1 el color con el que se
quiere el marcador de esquina
hdrawmarkers = vision.MarkerInserter('Shape', 'Circle', 'BorderColor',
'Custom', 'CustomBorderColor', color); %se escoge un marcador circular
rojo

imshow(I)
imwrite(uint8(I*255), 'out.tif');
% Convert the grayscale input image I to an RGB image J before inserting
the marker
J = repmat(I,[1 1 3]);
J = step(hdrawmarkers, J, pts); % draw directly in the image
imshow(J); title ('Esquinas detectadas en la Imagen'); %se muestra la
imagen con esquinas marcadas
imwrite(uint8(J*255), 'foto4corners.tif')
```

ANEXO E

IMPLEMENTACION EN MATLAB DE LA SOLUCION EN FORMA

CERRADA

```

%% Calibracion con el algoritmo propuesto por Zhang

% Input:           Xw: Matriz Npx3 de puntos homogeneos 3D en Z=0
%                 Xi: Matriz Npx3 de puntos homogeneos 2D
%                 n: numero de planos

% Output:          H0: Conjunto de homografias 3x3
%                 A: Matriz Intrinseca 3x3
%                 R: conjunto de matrices de rotacion 3x3
%                 t: conjunto vectores de traslacion 3xn

%[H0,A0,R0,t0]=calibZ2(Xw,Xi,5)
function [H0,A,R,t]=calibZ2(Xw,Xi,n)

%Numero de puntos
sizeW=size(Xw);
sizeI=size(Xi);

if sizeW(1)~=sizeI(1)
    error('Xw y Xi deben tener el mismo numero de filas');
end

Np=sizeW(1)/n;

%Calculo de homografias
for i=1:n
    H0(3*i-2:3*i,:)= dltnH(Xw(((i-1)*Np)+1:i*Np,:),Xi(((i-1)*Np)+1:i*Np,:));
end

% Calibracion de Zhang

[filas, colum]=size(H0);

V=[];

for j=1:3:filas
    V=[V;const(H0(j:j+2,:))];
end

[U,D,Vv]=svd(V);
b=Vv(:,6);

%b=[B11,B12,B22,B13,B23,B33] '

B=[b(1) b(2) b(4);b(2) b(3) b(5);b(4) b(5) b(6)];
vo=((b(2)*b(4)) - (b(1)*b(5)))/((b(1)*b(3))-(b(2)^2));
lamda=b(6)-((b(4)^2)+((vo*((b(2)*b(4))-(b(1)*b(5))))/b(1)));
alfa=sqrt(lamda/b(1));
beta=sqrt((lamda*b(1))/((b(1)*b(3))-(b(2)^2)));

```

```

c=-b(2)*(alfa^2)*beta/lamda;
uo=(c*vo/alfa)-(b(4)*(alfa^2)/lamda);
A=[alfa c uo;0 beta vo;0 0 1];

%Extraccion de R y t

p=1;
R=[];
for k=1:3:filas
    lamda=1/(norm(inv(A)*H0(k:k+2,1)));
    r1=lamda*inv(A)*H0(k:k+2,1);
    r2=lamda*inv(A)*H0(k:k+2,2);
    r3=cross(r1,r2);
    t(:,p)=lamda*inv(A)*H0(k:k+2,3);

    % Mejor matriz de rotacion
    Q=[r1 r2 r3];
    [U,S,V]=svd(Q);
    R=[R;U*eye(3,3)*V'];
    p=p+1;
end

end

function V=const(H)

%H=[h1 h2 h3] hi son vectores columnas de H
%hi=[hi1,hi2,hi3]'
%vij=[hi1*hj1, hi1*hj2 + hi2*hj1, hi2*hj2, hi3*hj1 + hi1*hj3, hi3*hj2 +

v12=[H(1,1)*H(1,2), (H(1,1)*H(2,2))+H(2,1)*H(1,2), H(2,1)*H(2,2),
(H(3,1)*H(1,2)) + (H(1,1)*H(3,2)), (H(3,1)*H(2,2)) + (H(2,1)*H(3,2)),
H(3,1)*H(3,2)];
v11=[H(1,1)*H(1,1), (H(1,1)*H(2,1)) + (H(2,1)*H(1,1)), H(2,1)*H(2,1),
(H(3,1)*H(1,1)) + (H(1,1)*H(3,1)), (H(3,1)*H(2,1)) + (H(2,1)*H(3,1)),
H(3,1)*H(3,1)];
v22=[H(1,2)*H(1,2), (H(1,2)*H(2,2)) + (H(2,2)*H(1,2)), H(2,2)*H(2,2),
(H(3,2)*H(1,2)) + (H(1,2)*H(3,2)), (H(3,2)*H(2,2)) + (H(2,2)*H(3,2)),
H(3,2)*H(3,2)];

V=[v12;(v11'-v22)'];
end

```

ANEXO F

IMPLEMENTACION EN MATLAB DEL ALGORITMO DLT NORMALIZADO

```

% Algoritmo DLT con normalizacion

% Input:    pw: matriz nx3 con coordenadas 3D homogeneas en Z=0
%          pi: matriz nx3 con coordenadas 2D homogeneas
%          n>=4 minimo 4 puntos

% Output:   H: Homografia 3x3

function H=dltnH(pw,pi)

%Transformacion de similitud
Tw=similitud(pw);
Ti=similitud(pi);

[m,n]=size(pw);
A=[];
pwn=zeros(m,n);
pin=zeros(m,n);
c=zeros(1,3);

%Normalizacion de puntos
for i=1:m
    pwn(i,:)=Tw*(pw(i,:))';
    pin(i,:)=Ti*(pi(i,:))';
end

for i=1:m
    wi=pin(i,3);
    yi=pin(i,2);
    xi=pin(i,1);
    Xi=pwn(i,:);
    Ai=[c -wi*Xi yi*Xi;wi*Xi c -xi*Xi];
    A=[A;Ai];
end

%Calculo de homografia
[U,D,V]=svd(A);
[a,b]=size(V);
V=V(:,b);
Hn=[V(1) V(2) V(3);V(4) V(5) V(6);V(7) V(8) V(9)];
Hn = Hn./Hn(3,3);
H=(inv(Ti))*Hn*Tw;
end

%Calculo de la transformada de similitud
function T=similitud(p)

[m,n]=size(p);
pn=zeros(m,n);
C=centroide(p);
S=1;
teta=0;

```

```
tx=-C(1);
ty=-C(2);
S=1;
Tt=[S*cos(teta) -S*sin(teta) tx;S*sin(teta) S*cos(teta) ty;0 0 1];
dp=0;

for i=1:m
    pn(i,:)=Tt*(p(i,:))';
    dp=dp+norm(pn(i,:));
end

dp=dp/m;
fs=sqrt(2)/dp;
Ts=[fs 0 0;0 fs 0;0 0 1];
T=Ts*Tt;
end

function Cc=centroide(p)

[m,n]=size(p);

X=mean(p(:,1));
Y=mean(p(:,2));
Cc=[X;Y];
end
```

ANEXO G

PROGRAMA PARA CALCULAR LA DISTORSION RADIAL

```

%% Funcion distorRad para calcular el vector k=[k1 k2]' con los
componentes
%%principales de distorsion radial introducida por el lente de la camara

% Input:      Xw: Matriz (Np*n)x3 de coordenadas homogeneas de puntos 3D
en Z=0
%            Xi: Matriz (Np*n)x3 de coordenadas homogeneos de puntos 2D
%            n: Numero de planos
%            A: Matriz Intrinseca 3x3
%            R: Matrices de rotacion 3nx3
%            t: Vectores de traslacion 3xn

% Output:     k: vector 2x1 de los coeficientes de distorsion radial

function k=distorRad(Xw,Xi,n,A,R,t)
sizeW=size(Xw);
sizeI=size(Xi);
if sizeW(1)~=sizeI(1)
    error('Xw y Xi deben tener el mismo numero de filas');
end
Np=sizeW(1)/n; % Numero de puntos
% Calculo de matrices D y d del sistema Dk=d
D=[];
d=[];
for i=1:n
    [Di di]=distR(Xi((i*Np)-Np+1:i*Np,:),Xw((i*Np)-
Np+1:i*Np,:),A,R((i*3)-2:3*i,:),t(:,i));
    D=[D;Di];
    d=[d;di];
end
% Solucion dada por minimo cuadrados lineal
k=(inv(D'*D))*(D')*d;
end
function [D,d]=distR(UVr,XYw,A,R,t)
[m,n]=size(XYw);
uo=A(1,3);
vo=A(2,3);
Rt=[R(:,1) R(:,2) t];
D=[];
d=[];
for i = 1:m
    xy=Rt*(XYw(i,:))';
    xy=xy./xy(3);
    uv=A*xy;
    u=uv(1);
    v=uv(2);
    r=(xy(1)^2)+(xy(2)^2);
    Di=[((u-uo)*r),((u-uo)*(r^2))];
    UVrr=UVr(i,:);
    di=[UVrr(1)-u;UVrr(2)-v];
    D=[D;Di];
    d=[d;di];
end
end

```

ANEXO H

OPTIMIZACION CON LEVENBERG-MARQUARDT

```

%Minimizacion de todos los parametros con LM

%Input:  Xi: Matriz (Np*n)x3 de coordenadas homogeneos de puntos 2D
%        Xw: Matriz (Np*n)x3 de coordenadas homogeneas de puntos 3D en
Z=0
%        A0: matriz intrinseca 3x3
%        R0: conjunto de matrices de rotacion, 3nx3 (n = numero de
planos)
%        t0: conjunto de vectores de traslacion, 3xn
%        k0: vector de distorsion radial
%        n: numero de planos

%Output: A: Matriz Intrinseca final 3x3
%        R: conjunto de matrices de rotacion finales, 3nx3 (n = numero de
planos)
%        t: conjunto de vectores de traslacion finales, 3xn
%        k: vector de distorsion radial final

function [A,R,t,k]=LM(Xi,Xw,A0,R0,t0,k0,n)

opt = optimset('Algorithm','levenberg-marquardt');
p=vectorP(A0,k0,R0,t0,n);
pref = lsqcurvefit(@MyfunCalib,p,Xw,Xi,[],[],opt); %parametros refinados
con LM
A = [pref(1) pref(2) pref(3); 0 pref(4) pref(5); 0 0 1];
R=[];
t=[];
j=8;
for i=1:n
    %Reconstruccion de R y t
    r = [pref(j);pref(j+1);pref(j+2)];
    ti = [pref(j+(n*3));pref(j+(n*3+1));pref(j+(n*3+2))];
    Ri = RodriguesMatrix(r);
    R=[R;Ri];
    t=[t,ti];
    j=j+3;
end
k=[pref(6),pref(7)];

```


ANEXO I

SUBROUTINA DE LA FUNCION LM PARA VECTORIZAR LOS PARAMETROS INICIALES

```

%%Esta funcion construye el vector p para la minimizacion con LM usando
la funcion de MATLAB: lsqcurvefit
%Input:  A: matriz intrinseca 3x3
%        k: vector de distorsion radial
%        R: conjunto de matrices de rotacion, 3nx3 (n = numero de planos)
%        t: conjunto de vectores de traslacion, 3xn
%        n: numero de planos

%Output: p: vector p que contiene todos los parametros de calibracion
%         iniciales para proceder a la optimizacion con LM

function p=vectorP(A,k,R,t,n)
r=[];
for i=1:n
    ri=RodriguesVector(R((i*3)-2:i*3,:));
    r=[r,ri'];
end
t = reshape(t,1,n*3);
p = [A(1,1), A(1,2),A(1,3),A(2,2),A(2,3),k(1),k(2),r,t];

```

ANEXO J**FORMULA DE RODRIGUES PARA VECTORIZAR LA MATRIZ DE
ROTACION**

```
% Formula de rotacion de Rodrigues
% R. Hartley y A. Zisserman, "Multiple View Geometry in Computer
Vision",pp.584

function r=RodriguesVector(R)
phi = acos((trace(R)-1)/2);
r = phi*[R(3,2)-R(2,3);R(1,3)-R(3,1);R(2,1)-R(1,2)]*(1/(2*sin(phi)));
end
```

ANEXO K

FUNCION MYFUNCALIB

```

function X = MyfunCalib(p,Xw)
%p=[A(1,1),
A(1,2),A(1,3),A(2,2),A(2,3),k(1),k(2),r(1),r(2),r(3).....,t(1),t(2),t(
3).....]
A = [p(1) p(2) p(3); 0 p(4) p(5); 0 0 1]; % Matriz Intrinseca
%Tamano de p
[~,m]=size(p);
k1=p(6); k2=p(7);
a=m-7;
Nplanos=(a/3)/2;
[f,~]=size(Xw);
Npuntos=f/Nplanos;
Xw=Xw(1:Npuntos,:);
X = [];
R=[];
t=[];
j=8;
for i=1:Nplanos
    %Reconstruccion de R y t
    r = [p(j);p(j+1);p(j+2)];
    ti = [p(j+(Nplanos*3));p(j+(Nplanos*3+1));p(j+(Nplanos*3+2))];
    Ri = RodriguesMatrix(r);
    R=[R;Ri];
    t=[t,ti];
    j=j+3;
    for k = 1:Npuntos
        xy=(Ri*Xw(k,:)') +ti;
        xy=xy./xy(3);
        d=(xy(1)^2)+(xy(2)^2);
        xy_dis(1)=xy(1)+(xy(1)*((k1*d)+(k2*(d^2))));
        xy_dis(2)=xy(2)+(xy(2)*((k1*d)+(k2*(d^2))));
        uv=A*[xy_dis(1);xy_dis(2);1];
        X=[X;uv'];
    end
end
end
end

```

ANEXO L

FORMULA DE RODRIGUES PARA RECUPERAR LA MATRIZ DE CALIBRACION A PARTIR DE SU VECTOR

```
%% Recuperar matriz R a partir de v
% http://mathworld.wolfram.com/RodriguesRotationFormula.html

function R = RodriguesMatrix(v)
w = [0, -v(3), v(2); v(3), 0, -v(1); -v(2), v(1), 0];
theta = norm(v);
R = eye(3)+(sin(theta)/(theta)*w) + ((1-cos(theta))/(theta^2))*w^2;
end
```

ANEXO M

FUNCION PARA CALCULAR EL ERROR RMS

```
%% Funcion para estimar el RMS de las distancias en pixeles entre las  
coordenadas detectadas y las proyectadas
```

```
function RMS=RMS2(XI,XW,A,k,R,t,n)
```

```
p=vectorP(A,k,R,t,n);  
Xp = MyfunCalib(p,XW);  
Ir=(XI-Xp);  
Xr=sqrt(mean(Ir(:,1).^2));  
Yr=sqrt(mean(Ir(:,2).^2));  
RMS=sqrt((Xr^2)+(Yr^2));
```